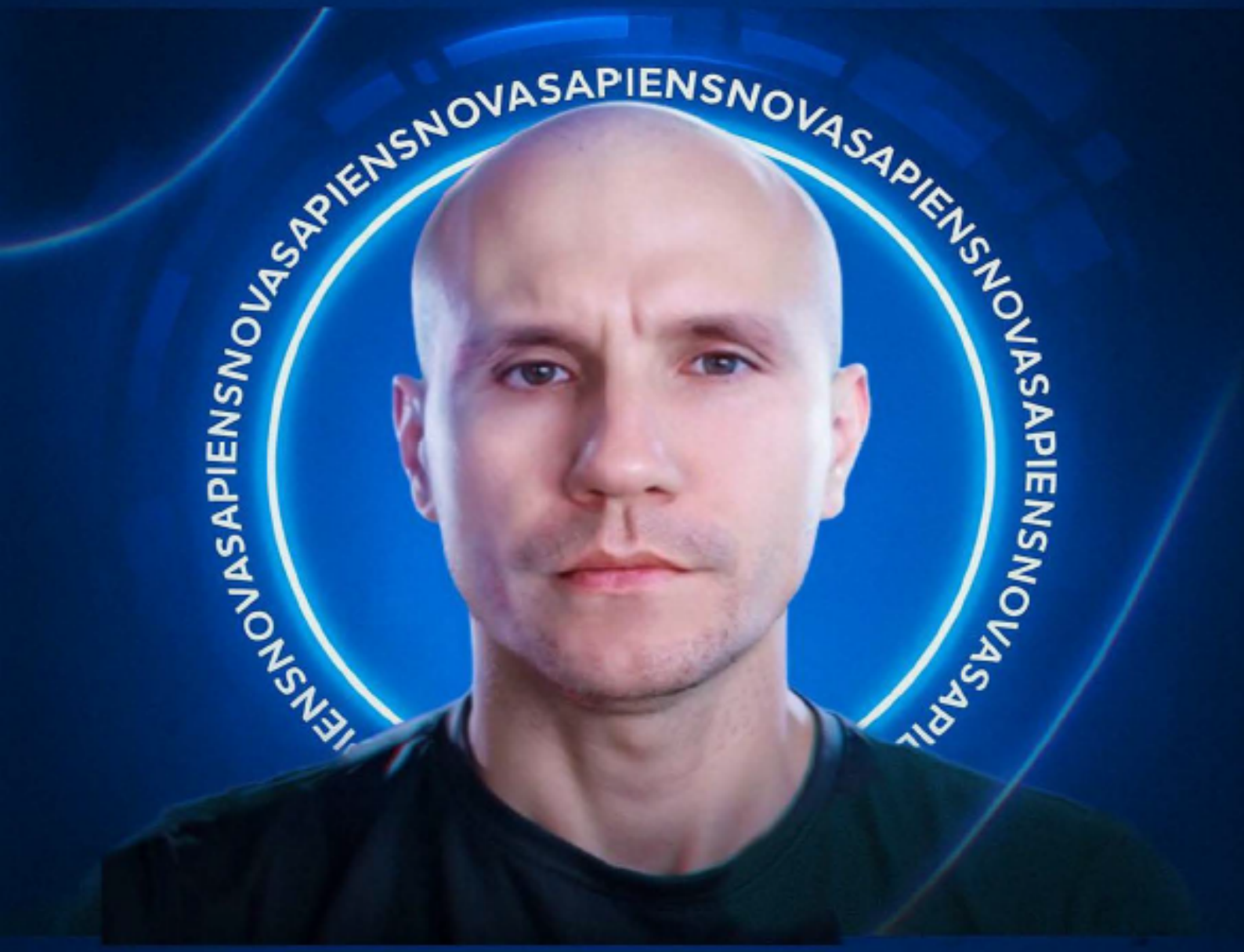
ДАЙДЖЕСТ ПО ПРОМТ-ИНЖИНИРИНГУ

Данные современных исследований
за январь, февраль 2025 год

NovaSapiens Research



Специально для сообщества ainovasapiens
https://t.me/ainovasapiens

Содержание

использоваться любым пользователем в повседневной работе с LLM для различных творческих задач.

24. Применение максима Грайса в цикле взаимодействия человек-ИИ: дизайнерские идеи из участнического подхода

Исследование предлагает 9 практических рекомендаций по дизайну взаимодействия с LLM, основанных на максимах Грайса. Эти рекомендации структурированы по циклу взаимодействия (формулирование цели, генерация ответа, оценка результата) и могут быть немедленно применены пользователями через стратегии составления промптов. Исследование объединяет теоретические основы коммуникации с практическими потребностями, учитывая разные уровни пользователей.

28. Диверсификация выборки улучшает инференс ScalingLLM

Исследование предлагает простые в применении методы диверсификации запросов (Role, Instruction, переформулирование), которые значительно улучшают качество ответов LLM. Пользователи любого уровня могут немедленно применить эти техники, не требующие API или специальных знаний. Методы универсальны для разных задач, показали эмпирически подтвержденную эффективность и имеют теоретическое обоснование.

32. Концептуально-ориентированное побуждение цепочки мыслей для парного сравнительного оценки текстов с использованием больших языковых моделей

Исследование предлагает практический метод анализа текстов с помощью LLM, который не требует больших размеченных данных. CGCoT-подход (поэтапные направленные вопросы) и попарные сравнения легко адаптируются для различных задач и доступны широкой аудитории. Метод показывает высокую эффективность при минимальных затратах на разработку, хотя полная реализация требует некоторых технических знаний.

36. За пределами цепочки размышлений: Обзор парадигм Chain-of-X для больших языковых моделей

Исследование предоставляет всеобъемлющую таксономию Chain-of-X методов, большинство из которых можно применить в повседневном взаимодействии с LLM. Особенно ценны концепции декомпозиции проблем, структурирования промежуточных шагов и механизмов самопроверки. Некоторые методы требуют технических знаний, что снижает доступность для неспециалистов, однако общие принципы легко адаптируются для стандартных чатов.

40. Понимание перед разумом: улучшение цепочки размышлений с помощью итеративного суммирования в преднастройке

Исследование предлагает метод "понимание перед рассуждением", который легко адаптировать для повседневного использования в чатах с LLM. Пользователи могут применять принцип поэтапной обработки информации, сначала структурируя данные, затем рассуждая. Метод показывает значительное улучшение точности на разных моделях и задачах, особенно когда ключевая информация неявна.

44. Большие языковые модели для локализации уязвимостей в файле могут оказаться «потерянными в конце»

Исследование выявляет "lost in the end" эффект в LLM и предлагает простую стратегию "chunking" для повышения эффективности обнаружения уязвимостей на 37%. Предоставляет конкретные рекомендации по оптимальным размерам фрагментов для анализа кода (500-6500 символов), которые любой пользователь может немедленно применить без специальных инструментов. Ограничения: исследованы только три типа уязвимостей и ограниченный набор моделей.

48. PReasoning о теории разума на основе гипотез для больших языковых моделей

Исследование представляет высокую ценность, предлагая метод улучшения взаимодействия с LLM в задачах понимания намерений. Алгоритм Thought Tracing дает практический подход к структурированию запросов, демонстрирует способы преодоления ограничений моделей и работы с неопределенностью. Основные концепции доступны для адаптации, хотя полная реализация требует технических знаний.

52. Модульное тестирование: прошлое и настоящее. Исследование влияния LLM на обнаружение дефектов и эффективность

Исследование демонстрирует высокую практическую ценность, предоставляя количественные доказательства преимуществ LLM в юнит-тестировании. Результаты показывают значительное повышение продуктивности (+119% тестов, больше обнаруженных дефектов) и применимы напрямую разработчиками. Выявление компромисса между количеством и качеством дает важное понимание ограничений.

56. VisPath: Автоматизированный синтез кода визуализации с помощью многопутевого рассуждения и оптимизации на основе обратной связи

VisPath предлагает ценную методологию мульти-путевого рассуждения и итеративного улучшения визуализаций, которая может быть адаптирована для широкого спектра взаимодействий с LLM. Пользователи могут применять принципы генерации нескольких вариантов решения, их оценки и синтеза оптимального результата для улучшения качества визуализаций и других задач. ## Ключевые аспекты исследования: 1.

60. О надежности генеративных базовых моделей: руководство, оценка и перспектива

Исследование предлагает комплексную основу для оценки надежности генеративных моделей с гибкими руководствами и динамической системой TrustGen. Высокая ценность для разработчиков и продвинутых пользователей, предоставляет как теоретическую базу, так и практические инструменты с открытым кодом. Требует определенной технической подготовки, но многие принципы могут быть адаптированы и упрощены для широкой аудитории.

64. Многоэтапное, цепочное редактирование пост-текстов для неверных резюме

Исследование предлагает практичную методологию многоэтапного редактирования текстов с использованием Chain-of-Thought промптов для выявления и исправления фактологических ошибок. Подход не требует технических знаний, применим в стандартных чатах с LLM и демонстрирует значительное улучшение точности текстов. Ценность для пользователей в готовых промптах и пошаговой методике, которые можно применять для улучшения автоматически генерируемого контента.

68. Парсинг логов с использованием LLM с самосгенерированным обучением в контексте и самокоррекцией

Исследование предлагает адаптивный фреймворк для парсинга логов с использованием LLM, демонстрируя инновационные подходы к самокоррекции и обучению в контексте. Методы могут быть адаптированы для различных LLM и применены в других задачах с эволюционирующими данными. Концепции самогенерируемого обучения и самокоррекции имеют широкий потенциал применения, хотя требуют некоторой адаптации для широкой аудитории.

72. Улучшение манипуляций на уровне символов с помощью метода разделяй и властвуй

Исследование предлагает практичный метод "разделяй и властвуй" для улучшения манипуляций с символами в LLM, который может быть применен без дополнительного обучения моделей. Подход решает реальную проблему обработки текста и значительно повышает точность базовых операций с символами. Требует некоторого технического понимания, но принципы доступны для адаптации широкой аудиторией.

76. Скамейка LCTG: Бенчмарк генерации текста с контролем LLM

Исследование предлагает универсальную методологию контроля генерации текста по четырем аспектам (формат, количество символов, ключевые/запрещенные слова), применимую в любых LLM. Представленные структуры промптов и подходы к оценке могут быть непосредственно использованы пользователями для повышения качества взаимодействия с чат-моделями. Выявленные особенности разных моделей помогают выбрать оптимальную для конкретных задач.

80. Запоминание вместо рассуждения? Обнаружение и снижение verbatim запоминания в оценке понимания персонажей большими языковыми моделями

Исследование предлагает практические методы промптинга, стимулирующие LLM к рассуждениям вместо воспроизведения запомненной информации. Концепции "gist memory" и "verbatim memory" имеют высокую образовательную ценность. Пользователи могут непосредственно применять предложенные промпты для получения более осмысленных ответов, особенно при анализе художественных произведений.

84. От Системы 1 к Системе 2: Обзор Рассуждений Больших Языковых Моделей

Исследование предоставляет ценное понимание принципов рассуждения в LLM, которые могут быть адаптированы в виде техник промптинга (структурированное рассуждение, верификация шагов, макро-действия). Понимание различий между System 1 и System 2 помогает пользователям эффективнее формулировать запросы для разных типов задач, хотя некоторые методы требуют технической подготовки и адаптации для широкого применения. ## Ключевые аспекты исследования: 1.

88. Изучение влияния больших языковых моделей на пользовательские истории, созданные студентами, и тестирование приемки в разработке программного обеспечения

Исследование дает конкретные данные о том, где LLM помогают (критерии приемки +88%, ценность формулировок +23%) и где мешают (определение объема -24%). Методика работы с LLM универсально применима. Пользователи получают важное концептуальное понимание: LLM эффективны для детализации, но требуют контроля объема задач.

92. Обнаружение неэффективностей в коде, сгенерированном LLM: к всеобъемлющей таксономии

Исследование предлагает практичную таксономию неэффективностей в коде, генерируемом LLM, которая может служить чеклистом при проверке кода. Выявленные категории проблем (логика, производительность, читаемость, сопровождаемость, ошибки) и их взаимосвязи помогают пользователям формировать более точные запросы и критически оценивать результаты. Опрос практиков подтверждает актуальность проблем для реальной разработки.

96. Сравнение кода, написанного человеком, и кода, сгенерированного ИИ: Вердикт всё ещё не вынесен!

Исследование предоставляет практически применимые выводы о сравнении кода, написанного людьми и сгенерированного LLM. Результаты показывают, что LLM лучше в

стандартных задачах, но отстают в сложных. Выводы о функциональных различиях, безопасности и сложности кода напрямую полезны для широкого круга пользователей.

100. Агентное извлечение информации

Исследование вводит концепцию агентного информационного поиска, переопределяя взаимодействие с LLM как достижение "информационного состояния", а не просто получение информации. Предлагает практичный подход к многошаговому взаимодействию с LLM для решения комплексных задач. Концепции и примеры применимы сразу, без дополнительных инструментов, хотя полная реализация некоторых возможностей может требовать API-доступа.

104. Научиться задавать вопросы: Когда LLM-агенты сталкиваются с неясными инструкциями

Исследование предлагает ценную концепцию проактивного запроса уточнений при неясных инструкциях и классификацию типичных проблем, что помогает пользователям формулировать более эффективные запросы. Основные принципы могут быть легко адаптированы обычными пользователями в их промптах. Техническая реализация бенчмарка и автооценщика имеет ограниченную применимость для широкой аудитории.

108. Могут ли большие языковые модели заменить человеческих оценщиков? Эмпирическое исследование LLM как судьи в программной инженерии

Исследование предоставляет практические рекомендации по использованию LLM для оценки кода, с акцентом на превосходство output-based методов с большими моделями. Выводы о различиях в эффективности методов для разных задач и предупреждение о ненадежности попарного сравнения имеют прямую практическую ценность. Ограничение исследования задачами программирования снижает его универсальность.

112. Выявление недостатков в том, как люди и большие языковые модели интерпретируют субъективный язык

Исследование выявляет критические несоответствия между ожиданиями людей и тем, как LLM интерпретируют субъективные инструкции. Конкретные примеры проблем (например, "энтузиастичный"=>"нечестный", "остроумный"=>"оскорбительный") имеют прямую практическую ценность для пользователей при формулировании запросов. Сам метод TED требует доступа к градиентам и вычислительным ресурсам, но концептуальное понимание проблемы применимо немедленно.

116. Постобучение LLM: Погружение в рассуждения больших языковых моделей

Исследование предоставляет всесторонний обзор методов пост-тренировки LLM с высокой концептуальной ценностью. Особую практическую пользу представляют методы масштабирования при тестировании (TTS), которые могут применяться через промпты. Однако многие методы RL требуют специальных знаний и ресурсов, что снижает прямую применимость для обычных пользователей.

120. HPSS: Эвристическая стратегия поиска подсказок для оценщиков LLME.

Исследование представляет высокую ценность, предлагая структурированный подход к оптимизации промптов через 8 ключевых факторов. Пользователи могут непосредственно применять выявленные принципы (шкала 1-10, структура промпта, критерии оценки) для улучшения взаимодействия с LLM. Несмотря на технический характер полной реализации, основные концепции доступны для адаптации широкой аудиторией.

124. Исследование и контроль разнообразия в беседе с LLM-агентом

Исследование предлагает практичный метод контроля разнообразия в диалогах с LLM через управление содержимым промпта. Хотя полная реализация APP требует доступа к весам внимания, основные принципы (удаление избыточной информации, порядок блоков) легко адаптируются к обычному использованию. Исследование дает глубокое понимание факторов, влияющих на разнообразие ответов, что ценно для любого пользователя LLM.

128. Визуальное описание на основе контекста снижает количество галлюцинаций и улучшает reasoning в LVLM

Исследование предоставляет ценное понимание причин галлюцинаций в LVLMs и предлагает метод VDGD для их снижения. Хотя полная реализация требует технических знаний, основной принцип (использование описания изображения перед основным запросом) может быть легко применен обычными пользователями через последовательные запросы, значительно улучшая точность ответов для задач, требующих рассуждения. ## Ключевые аспекты исследования: 1.

132. Галлюцинации LLM в практической генерации кода: феномены, механизмы и меры по их уменьшению

Исследование предоставляет ценную таксономию галлюцинаций в генерации кода, анализ их причин и практический метод смягчения на основе RAG. Эти знания помогают пользователям лучше формулировать запросы, оценивать ответы и понимать ограничения LLM в реальных сценариях разработки. Основные концепции могут быть адаптированы даже без сложной технической реализации.

136. Обратите внимание на разрыв уверенности: избыточная уверенность, калибровка и эффекты отвлекающих факторов в больших языковых моделях

Исследование выявляет критическую проблему избыточной уверенности LLM и предоставляет практические стратегии улучшения взаимодействия. Показывает, как формулировать запросы с вариантами ответов для повышения точности, особенно для меньших моделей. Объясняет различия в поведении моделей разного размера и влияние типов вопросов на калибровку.

140. Оценка управляемости подсказок больших языковых моделей

Исследование предоставляет ценную методологию для измерения и понимания стерируемости LLM через промпты. Основные выводы о количестве необходимых направляющих утверждений, асимметрии стерируемости и различиях между моделями напрямую применимы к разработке эффективных стратегий промптинга. Требует некоторых технических знаний, но концепции адаптируемы для обычных пользователей.

144. Объяснение сбоев GitHub Actions с помощью больших языковых моделей: вызовы, идеи и ограничения

Исследование демонстрирует эффективность LLM в объяснении ошибок GitHub Actions, выявляя пять ключевых атрибутов полезных объяснений: ясность, применимость, специфичность, контекстуальная релевантность и лаконичность. Результаты показывают, что LLM эффективны для простых ошибок, но требуют улучшения для сложных случаев. Концепции и методы могут быть адаптированы для других технических контекстов.

148. К способностям рассуждения малых языковых моделей

Исследование дает ценное понимание возможностей малых языковых моделей и методов их оптимизации. Выводы о формулировках запросов и выборе моделей практически применимы, а понимание ограничений помогает формировать

реалистичные ожидания. Однако многие технические аспекты недоступны для прямого применения обычными пользователями, а некоторые выводы имеют ограниченную практическую ценность для повседневного использования.

152. Самообучающееся агентное понимание длинного контекста

Исследование предлагает высокоэффективную методологию Chain of Clarifications для работы с длинными контекстами. Пользователи могут адаптировать ключевые концепции (поэтапное уточнение вопросов, указание на релевантные части текста) для повседневного использования LLM, значительно улучшая понимание длинных документов. Техническая сложность некоторых аспектов снижает непосредственную применимость, но концептуальная ценность остается высокой.

156. Формирование игры: как контекст влияет на принятие решений ИИ

Исследование демонстрирует, как контекст (тема, отношения между участниками, тип мира) существенно влияет на решения LLM даже при одинаковой базовой структуре задачи. Эти знания позволяют пользователям формировать более эффективные запросы, предвидеть реакции моделей и выбирать подходящие LLM для конкретных задач. Хотя методология требует адаптации, концепции применимы непосредственно.

160. Генерация входных данных для тестирования значений границ с использованием проектирования подсказок с большими языковыми моделями: обнаружение ошибок и анализ покрытия

Исследование предлагает практичную методологию использования LLM для генерации тестовых данных через простые промпты, которые любой пользователь может адаптировать. Демонстрирует эффективность LLM в обнаружении сложных ошибок и важность качества тестов над количеством. Однако полная ценность требует понимания концепций тестирования и доступа к исходному коду, что ограничивает применимость для некоторых пользователей.

164. LogiDynamics: Раскрывая динамику логического вывода в рассуждении больших языковых моделей

Исследование демонстрирует, когда использовать прямые запросы (для текстовых/простых задач) и когда структурированное рассуждение (для визуальных/сложных задач). Оно предлагает методы улучшения ответов через выбор гипотез, верификацию и уточнение. Выводы экспериментально подтверждены и применимы к широкому спектру задач, хотя требуют базового понимания логических концепций.

168. ПОПИШИ: Структурированное рассуждение Больших Языковых Моделей с экстраполяцией достоверности, вдохновленной графами знаний

GIVE предлагает мощный метод структурированного рассуждения с использованием ограниченной внешней информации. Хотя полная реализация технически сложна, ключевые концепции (разбиение запроса, экстраполяция на основе ограниченных фактов, контрфактуальное рассуждение) могут быть адаптированы обычными пользователями для улучшения взаимодействия с LLM и получения более достоверных ответов в сложных областях знаний. ## Ключевые аспекты исследования: 1.

172. Должны ли вы использовать вашу модель большого языка для исследования или эксплуатации?

Исследование демонстрирует высокую ценность в понимании возможностей LLM для исследования больших пространств действий (стратегии запросов легко применимы), но ограниченную полезность для задач оптимизации на основе числовых данных (требуются технические навыки). Предоставляет важные концептуальные знания о том,

когда и как использовать LLM для принятия решений. ## Ключевые аспекты исследования: 1.

176. Улучшение разговорных агентов с теорией разума: согласование убеждений, желаний и намерений для взаимодействия, похожего на человеческое

Исследование предлагает ценную BDI-модель (убеждения, желания, намерения) для улучшения диалога с LLM. Хотя технические методы требуют специальных навыков, принципы могут быть адаптированы для структурирования промптов. Наглядные примеры демонстрируют преимущества учета ToM.

180. SecureFalcon: Удалось ли нам достичь автоматического обнаружения уязвимостей в программном обеспечении с помощью LLM?

Исследование демонстрирует эффективное применение LLM для обнаружения уязвимостей в коде с высокой точностью (94%). Предлагаемая архитектура SecureFalcon и методология имеют значительную ценность для разработчиков и могут быть интегрированы в инструменты разработки. Однако узкая специализация (только C/C++ код) и необходимость значительных ресурсов для воспроизведения ограничивают непосредственную применимость для широкой аудитории.

184. Генерация ключевых фраз без обучения: исследование специализированных инструкций и агрегации многократных образцов на больших языковых моделях

Исследование предлагает высокоэффективные стратегии мульти-сэмплинга и агрегации результатов, которые значительно улучшают генерацию ключевых фраз. Особенно ценны методы Frequency order и динамический выбор количества результатов, которые легко адаптируются для широкого спектра задач. Однако некоторые исследованные подходы (специализированные промпты, дополнительные инструкции) оказались неэффективными, а специфика задачи генерации ключевых фраз ограничивает широкую применимость.

188. Пауза-Настройка для Понимания Долгого Контекста: Легкий Подход к Перенастройке Внимания LLM

Исследование предлагает методы улучшения работы с длинными контекстами через вставку пауз-токенов. Часть методов (вставка пауз без файнтюнинга) доступна для непосредственного применения обычными пользователями. Концепция структурирования длинных запросов с паузами проста для понимания и решает актуальную проблему "lost in the middle", значительно улучшая извлечение информации из длинных текстов.

192. CallNavi: Исследование и вызов маршрутизации и вызова функций в крупных языковых моделях

Исследование предлагает практичные методы оптимизации работы с API (асинхронная генерация, обратное мышление), применимые обычными пользователями. Понимание влияния сложности задач на производительность моделей и сравнительный анализ 17 LLM помогают формировать эффективные запросы и выбирать подходящие модели. Основные концепции могут быть адаптированы для различных задач.

196. Две головы лучше, чем одна: Двухмодельная вербальная рефлексия во время вывода

Исследование представляет ценную концепцию разделения ролей рассуждения и критики в LLM. Хотя техническая реализация сложна для обычных пользователей,

принципы могут быть адаптированы через структурированные запросы и многошаговый диалог. Высокая концептуальная ценность и методология структурированного дерева мышления дают практические инструменты для улучшения качества взаимодействия с LLM.

200. За пределами точного совпадения: семантическая переоценка извлечения событий с помощью крупных языковых моделей

Исследование предлагает ценную концепцию семантической оценки извлечения событий, демонстрируя, что LLM работают значительно лучше, чем показывают стандартные метрики. Пользователи могут применить принципы семантической оценки вместо точного совпадения, что улучшит интерпретацию ответов. Понимание типичных ошибок помогает формулировать более эффективные запросы.

204. «Эскалация бенчмаркинга перевода кода на основе LLM в эпоху класс-уровня»

Исследование предлагает три практические стратегии перевода кода на уровне классов, анализ их эффективности для разных LLM и языков программирования, а также детальную классификацию ошибок. Пользователи могут применять эти стратегии и знание о типичных ошибках для улучшения результатов перевода кода, хотя для полного использования результатов требуется определенная техническая подготовка. ## Ключевые аспекты исследования: 1.

208. Повторное исследование способности графов к рассуждению больших языковых моделей: случаи изучения в переводе, связанности и кратчайшем пути

Исследование предоставляет практические рекомендации по оптимальному представлению графов в запросах к LLM: использование списков соседей вместо списков рёбер, последовательное именование узлов, включение алгоритмических подсказок. Выявленные факторы влияния могут применяться для повышения точности ответов в графовых задачах. Ограничением является узкий фокус на графовых задачах и необходимость некоторых технических знаний.

212. RankCoT: Усовершенствование знаний для генерации с увеличением поиска через ранжирование цепочек мышления

RankCoT предлагает ценные методы для улучшения взаимодействия с LLM через структурированные рассуждения, ранжирование и самоанализ. Большинство концепций (множественные CoT, самоанализ, выбор лучших вариантов) могут быть адаптированы обычными пользователями для повышения точности ответов в стандартных чатах, несмотря на некоторые технические аспекты, требующие специальных знаний.

216. Осведомленное объединение с учетом неопределенности: ансамблевый каркас для снижения галлюцинаций в больших языковых моделях

Исследование предлагает ансамблевый метод UAF для снижения галлюцинаций LLM, комбинируя ответы нескольких моделей с учетом их точности и уверенности. Высокая концептуальная ценность основных принципов (использование нескольких моделей, учет уверенности, специализация моделей) позволяет пользователям адаптировать их для повседневного использования, особенно для критически важных запросов, требующих фактической точности.

220. Генерация онтологий с использованием больших языковых моделей

Исследование предлагает конкретные техники промптинга для генерации структурированных знаний и методы оценки качества. Хотя оно фокусируется на

узкоспециализированной области онтологий, принципы структурированного промптинга, формулирования требований через вопросы и многомерной оценки качества применимы к широкому спектру задач взаимодействия с LLM. Требует некоторой адаптации для широкой аудитории.

224. Генеративный искусственный интеллект: развивающаяся технология, растущее социальное воздействие и возможности для исследований в области информационных систем

Исследование предлагает ценную концептуальную основу для понимания GenAI как социотехнической системы с уникальными свойствами. Особенно полезны анализ "темной стороны" GenAI и системный взгляд на его возможности и ограничения. Однако высокий уровень абстракции и отсутствие конкретных практических рекомендаций снижают непосредственную применимость для широкой аудитории.

228. Раскрытие и причинное объяснение CoT: Причинная перспектива

Исследование предлагает ценную концепцию о причинно-следственных связях в рассуждениях LLM. Практическую ценность имеют техника ролевых запросов для улучшения логики рассуждений, классификация типичных ошибок и понимание важности первого шага. Однако многие технические аспекты (SCM, CACE, FSCE) недоступны широкой аудитории без специальных знаний.

232. Эффективность больших языковых моделей в написании формул сплавов

Исследование демонстрирует способность LLM переводить естественный язык в формальные спецификации Alloy, генерировать эквивалентные формулы и заполнять шаблоны. Несмотря на специализированный характер Alloy, методы имеют более широкое применение и могут быть адаптированы для других языков, упрощая работу с формальными методами для неспециалистов. ## Ключевые аспекты исследования: 1.

236. DeepRAG: Поэтапное мышление при извлечении для крупных языковых моделей

DeepRAG предлагает ценную методологию декомпозиции сложных вопросов на подзапросы и определения необходимости внешнего поиска. Хотя полная техническая реализация недоступна обычным пользователям, концептуальные принципы могут быть адаптированы для более эффективного взаимодействия с LLM через структурированные запросы и пошаговое рассуждение. ## Ключевые аспекты исследования: 1.

240. Изучение понимания кода в научном программировании: предварительные выводы от исследователей

Исследование выявляет конкретные проблемы с читаемостью кода (недостаточное комментирование, плохое именование, неудачная структура), которые пользователи могут учитывать при формулировании запросов к LLM и оценке результатов. Тенденция использования LLM для улучшения кода подтверждает ценность этого подхода для широкой аудитории. ## Ключевые аспекты исследования: 1.

244. От диагностики суб-способностей к генерации, согласованной с человеком: преодоление разрыва для контроля длины текста с помощью MARKERGEN

Исследование представляет трехэтапный подход к контролю длины текста (планирование, генерация, корректировка), который может быть адаптирован пользователями через промпты. Оно дает понимание причин ошибок в контроле длины и предлагает концептуальные решения. Однако полная реализация MARKERGEN требует технических знаний, что ограничивает прямую применимость для обычных

пользователей.

248. Соединение исследований HCI и ИИ для оценки разговорных помощников в области программной инженерии

Исследование предлагает ценные концепции, которые могут быть адаптированы для повседневного использования LLM: "LLM как судья" для оценки ответов, учет разнообразия пользователей, многоходовые взаимодействия и критическое отношение к "эталонным" ответам. Хотя полная реализация методологии требует технических навыков, общие принципы доступны широкой аудитории. ## Ключевые аспекты исследования: 1.

252. Обучение ИИ обработке исключений: Управляемая тонкая настройка с учетом человеческого суждения

Исследование выявляет критическое ограничение LLM (чрезмерную приверженность правилам) и предлагает практические решения. Особенно ценны выводы о важности объяснений и цепочек рассуждений. Пользователи могут применять эти принципы для получения более гибких ответов, формулируя запросы, учитывающие потребность в исключениях. Часть методов требует технических навыков, но концептуальное понимание доступно всем.

256. FINEREASON: Оценка и улучшение преднамеренного мышления больших языковых моделей через решение рефлексивных головоломок

Исследование предлагает ценные концепции для улучшения рассуждений LLM через проверку состояний и планирование шагов. Пользователи могут адаптировать принципы "State Checking" и "State Transition" для получения более надежных ответов. Однако полная реализация методологии требует технических знаний, что ограничивает прямую применимость для обычных пользователей.

260. Вознаграждение процесса графового рассуждения делает LLM более обобщенными рассуждателями

Исследование предлагает ценные концепции пошагового рассуждения и проверки для улучшения взаимодействия с LLM. Хотя технические аспекты требуют значительной адаптации, пользователи могут применять принципы генерации нескольких решений, структурированного рассуждения и перекрестного использования навыков между доменами задач в повседневной работе с LLM. ## Ключевые аспекты исследования: 1.

264. Шахерезада: Оценка математического рассуждения с помощью цепочки цепочек проблем в языковых моделях

Исследование предлагает ценные концепции для понимания возможностей LLM в логических рассуждениях. Методы forward и backward chaining могут быть адаптированы для проверки последовательности рассуждений моделей. Знание типичных ошибок помогает формулировать эффективные запросы.

268. Оценка предпочтений языковой модели с помощью нескольких слабых оценщиков

Исследование демонстрирует, как комбинирование оценок нескольких "слабых" моделей может превзойти одну "сильную" модель. Эта концепция адаптируема для обычных пользователей через запросы к разным моделям или использование разных формулировок. Метод устранения противоречий в оценках имеет высокую концептуальную ценность, помогая понять ограничения LLM и улучшить критическую оценку полученных ответов.

272. SAGE: Framework точного извлечения для RAG

SAGE предлагает ценные концепции для работы с LLM: семантическая целостность контекста, динамический отбор информации и самооценка качества ответов. Хотя техническая реализация недоступна обычным пользователям, принципы можно адаптировать для улучшения запросов к LLM и структурирования информации.

276. Ненастоящие языки - это не ошибки, а особенности для больших языковых моделей

Исследование демонстрирует, что LLM могут понимать даже сильно искаженный текст, что имеет высокую концептуальную ценность для понимания работы моделей. Однако методы требуют специальных алгоритмов, недоступных обычным пользователям. Ценность в основном в понимании устойчивости LLM к шуму и способов эффективной формулировки запросов.

280. Поспешность приводит к расточительности: оценка планировочных способностей LLM для эффективного и осуществимого многозадачности с временными ограничениями между действиями

Исследование имеет высокую концептуальную ценность в понимании ограничений LLM при планировании с временными ограничениями. Выявленные принципы (приоритет выполнимости над эффективностью, источники ошибок) полезны для формирования реалистичных ожиданий. Однако большинство выводов требуют значительной адаптации для практического применения, а технические детали ориентированы больше на исследователей, чем на широкую аудиторию.

284. Обобщение против запоминания: прослеживание возможностей языковых моделей до данных предварительной тренировки

Исследование имеет высокую концептуальную ценность, объясняя разницу между меморизацией и генерализацией в LLM для разных типов задач. Практическая ценность включает методы оптимизации промптов и понимание, что фактические вопросы требуют меморизации, а рассуждения — генерализации. Однако многие технические аспекты недоступны широкой аудитории без специальных знаний.

288. Думай внутри JSON: Стратегия укрепления соблюдения строгой схемы LLMSchema

Исследование предлагает ценный подход "think-then-answer" для структурированных ответов в JSON-формате. Основные концепции поэтапного заполнения структуры и разделения рассуждения и ответа могут быть адаптированы в промптах, однако техническая реализация (RL, функции вознаграждения) недоступна обычным пользователям. Ценность в понимании принципов структурированного взаимодействия с LLM.

292. Самоорганизованная цепочка размышлений

Исследование предлагает ценный подход к улучшению промптов через унификацию примеров. Концептуально полезно для понимания важности согласованности при создании примеров рассуждений, но полная реализация требует технических навыков и доступа к API. Обычные пользователи могут адаптировать принципы согласованности и итеративного улучшения.

№ 4. SR-FoT: Систематическая рамка силлогистического мышления для крупных языковых моделей, решающих задачи, основанные на знаниях

Ссылка: https://arxiv.org/pdf/2501.11599

Рейтинг: 85
Адаптивность: 90

Ключевые выводы:

Исследование предлагает новый фреймворк SR-FoT (Syllogistic reasoning Framework of Thought) для улучшения дедуктивного рассуждения в больших языковых моделях (LLM). Основная цель - повысить точность и строгость рассуждений LLM при решении задач, требующих знаний, путем применения силлогистического подхода. Результаты показывают, что SR-FoT превосходит существующие методы, такие как Chain-of-Thought (CoT), на нескольких наборах данных.

Объяснение метода:

SR-FoT предлагает практичный фреймворк силлогистического рассуждения, который может быть непосредственно применен пользователями для улучшения качества ответов LLM. Метод предоставляет готовые шаблоны промптов, универсален для разных задач и значительно повышает строгость рассуждений. Основное ограничение - необходимость структурирования многоэтапных промптов, что может быть сложно для начинающих пользователей.

Ключевые аспекты исследования: 1. **Фреймворк SR-FoT (Syllogistic reasoning Framework of Thought)** - многоступенчатая структура, направляющая LLM через процесс силлогистического рассуждения для решения сложных задач на основе знаний.

Пятиэтапный процесс рассуждения, включающий: объяснение вопроса, формулировку большой посылки, постановку вопроса для малой посылки, формулировку малой посылки и итоговое силлогистическое рассуждение.

Контролируемый доступ к информации на каждом этапе рассуждения для минимизации ошибок и повышения строгости логических выводов.

Автономное формулирование посылок моделью на основе встроенных знаний и контекста задачи без необходимости предварительной формализации библиотеки посылок.

Повышение строгости рассуждений по сравнению с методом Chain-of-Thought (CoT), что подтверждается экспериментально на нескольких наборах данных.

Дополнение: Исследование SR-FoT не требует дообучения модели или специального API. Все методы и подходы могут быть применены в стандартном чате с LLM. Авторы использовали как закрытые (GPT-3.5-turbo), так и открытые (DeepSeek-V2, Qwen1.5-32B) модели через стандартные API-вызовы, но сама методология полностью применима в обычном чате.

Основные концепции и подходы, которые можно применить в стандартном чате:

Пятиэтапная структура рассуждения: Объяснение вопроса (понимание задачи) Формулирование большой посылки (общее правило) Формулирование вопроса для малой посылки (что нужно знать о конкретном случае) Получение малой посылки (ответ на этот вопрос) Проведение силлогистического рассуждения (применение правила к конкретному случаю)

Шаблоны промптов для каждого этапа (приведены в статье) могут быть напрямую использованы пользователями.

Принцип ограничения видимой информации можно реализовать, разбивая взаимодействие на отдельные сообщения, где в каждом новом сообщении предоставляется только необходимая информация.

Ожидаемые результаты от применения этих концепций: - Повышение строгости рассуждений (до 96% строгости по сравнению с 80% у CoT) - Снижение количества ошибок в сложных рассуждениях - Более надежные и обоснованные ответы на вопросы, требующие логического мышления - Возможность решения сложных задач с использованием встроенных знаний модели

Пользователи могут адаптировать этот подход для собственных задач, не требуя никаких дополнительных инструментов или специального доступа к моделям.

Prompt:

Использование силлогистического мышления (SR-FoT) в промптах ## Основные принципы SR-FoT

Исследование SR-FoT предлагает пятиэтапный подход к решению задач с помощью больших языковых моделей:

Интерпретация вопроса Формулировка большей посылки (общий принцип) Постановка вопроса для меньшей посылки Формирование меньшей посылки (конкретный факт) Проведение силлогистического рассуждения для получения ответа ## Пример промпта с использованием SR-FoT

[======] Я хочу, чтобы ты решил следующую задачу, используя структурированный подход к рассуждению:

ЗАДАЧА: [Вставьте вашу задачу здесь, например: "Может ли дельфин выжить в пресной воде?"]

Пожалуйста, следуй этому пошаговому процессу:

ИНТЕРПРЕТАЦИЯ: Сначала объясни, как ты понимаешь вопрос и что именно нужно выяснить.

БОЛЬШАЯ ПОСЫЛКА: Сформулируй общий принцип или знание, которое относится к данному вопросу. Это должно быть утверждение, которое всегда верно и применимо к данной ситуации.

ВОПРОС ДЛЯ МЕНЬШЕЙ ПОСЫЛКИ: Определи, какую конкретную информацию нужно установить, чтобы применить общий принцип к данной задаче.

МЕНЬШАЯ ПОСЫЛКА: Предоставь конкретные факты о ситуации, описанной в задаче, которые соответствуют вопросу из предыдущего шага.

СИЛЛОГИСТИЧЕСКОЕ РАССУЖДЕНИЕ: Используя большую и меньшую посылки, проведи логическое рассуждение и сделай обоснованный вывод.

ИТОГОВЫЙ ОТВЕТ: Сформулируй четкий и однозначный ответ на исходный вопрос. [======]

Почему это работает

Данный подход эффективен по следующим причинам:

Структурированность - разбивает сложную задачу на понятные этапы **Изоляция информации** - на каждом этапе модель фокусируется только на релевантной информации **Строгость рассуждений** - силлогистический формат обеспечивает логическую связность **Снижение ошибок** - пошаговый подход минимизирует "галлюцинации" и логические ошибки **Прозрачность** - позволяет отследить, на каком этапе могла произойти ошибка Исследование показало, что этот метод превосходит стандартный Chain-of-Thought (CoT) подход, повышая точность ответов на различных наборах данных и обеспечивая более строгие рассуждения.

Для еще большей надежности можно использовать самосогласованность (SC-SR-FoT), генерируя несколько вариантов рассуждений и выбирая наиболее согласованный результат.

№ 8. Программирование, ориентированное на планирование: рабочий процесс программирования на большом языковой модели

Ссылка: https://arxiv.org/pdf/2411.14503

Рейтинг: 85
Адаптивность: 90

Ключевые выводы:

Исследование представляет новый рабочий процесс программирования с использованием больших языковых моделей (LPW), состоящий из двух фаз: генерации решения и реализации кода. Основная цель - улучшить как начальную генерацию кода, так и последующие уточнения. Результаты показывают значительное улучшение точности Pass@1 до 16.4% на различных бенчмарках по сравнению с существующими методами.

Объяснение метода:

Исследование предлагает двухфазный подход к генерации кода - планирование решения с верификацией и последующую реализацию с отладкой. Метод значительно повышает точность кода, легко адаптируется к стандартным чатам без API, помогает пользователям структурировать запросы и понимать ошибки. Подход применим не только к программированию, но и к другим задачам, требующим пошагового планирования и проверки.

Ключевые аспекты исследования: 1. **Двухфазный рабочий процесс для генерации кода (LPW)**: Исследование представляет структурированный подход к генерации кода с помощью больших языковых моделей (LLM), разделенный на фазу генерации решения и фазу реализации кода.

Верификация плана решения: Ключевая инновация заключается в проверке плана решения на тестовых примерах перед написанием кода. Это позволяет LLM понять логику решения и проверить ее корректность.

Пошаговая отладка на основе плана: При возникновении ошибок в коде, система сравнивает фактическое выполнение с ожидаемым поведением из верифицированного плана, что позволяет точно локализовать и исправить ошибки.

Автономная генерация информации для обратной связи: Вся дополнительная информация (план решения, верификация, объяснение кода) генерируется самой моделью LLM без необходимости в дополнительном обучении или аннотированных

корпусах.

Значительное улучшение точности генерации кода: На различных бенчмарках метод демонстрирует существенное повышение точности (Pass@1) по сравнению с существующими подходами, особенно для сложных задач.

Дополнение:

Применимость методов исследования в стандартном чате

Исследование LPW (Large Language Model Programming Workflow) представляет методы, которые **не требуют дообучения или специального API** и могут быть полностью реализованы в стандартном чате с LLM.

Ключевые концепции, которые можно применить:

Двухфазный подход: Разделение работы на планирование решения и реализацию кода. Пользователь может явно запросить: "Сначала составь пошаговый план решения задачи" "Теперь проверь этот план на следующем тестовом примере..." "Теперь напиши код, основываясь на проверенном плане"

Верификация плана перед написанием кода:

Попросить LLM проверить план на конкретных примерах с пошаговым выполнением Запросить анализ промежуточных значений, чтобы убедиться в правильности логики

Структурированная отладка:

При ошибках в коде, попросить LLM сравнить фактическое выполнение с ожидаемым поведением из верифицированного плана Запросить анализ расхождений и предложения по исправлению

Объяснение кода:

Запрашивать подробные объяснения каждой строки кода для лучшего понимания Ожидаемые результаты от применения этих концепций: - Значительное повышение качества генерируемого кода - Меньшее количество итераций отладки - Лучшее понимание логики решения - Более точная локализация ошибок

Эти подходы особенно полезны для сложных задач программирования, но концепция "план => верификация => реализация => отладка на основе плана" может быть адаптирована практически для любой сложной задачи, где важна точность выполнения.

Prompt:

Использование исследования LPW в промптах для GPT ## Ключевые идеи исследования для промптов

Исследование "Программирование, ориентированное на планирование" предлагает двухфазный подход к генерации кода с использованием LLM: 1. **Фаза планирования** - создание и верификация плана решения 2. **Фаза реализации** - написание кода на основе плана и его итеративное улучшение

Пример промпта на основе методологии LPW

[======] # Задача программирования: [описание задачи]

Инструкции: Я хочу, чтобы ты решил эту задачу программирования, используя двухфазный подход:

ФАЗА 1: ПЛАНИРОВАНИЕ РЕШЕНИЯ 1. Проанализируй задачу и создай детальный план решения 2. Определи ключевые алгоритмы и структуры данных 3. Перечисли шаги с ожидаемыми промежуточными результатами 4. Верифицируй план на примерах из условия задачи, "пройдя" через него вручную

ФАЗА 2: РЕАЛИЗАЦИЯ КОДА 1. Напиши код на [язык программирования] в соответствии с планом 2. Добавь комментарии, объясняющие ключевые части кода 3. Проверь код на тестовых примерах 4. Если найдены ошибки, локализуй их точно и предложи исправления

Примеры для проверки: [Входные данные 1] -> [Ожидаемый результат 1] [Входные данные 2] -> [Ожидаемый результат 2] [======]

Как работает этот подход

Улучшение понимания задачи: Заставляя модель сначала создать и верифицировать план, мы помогаем ей лучше понять суть проблемы до начала кодирования.

Локализация ошибок: Сравнивая ожидаемые промежуточные результаты из плана с фактическими результатами кода, модель может точнее определить источник ошибок.

Структурированное мышление: Двухфазный подход предотвращает "прыжки к решению" и заставляет модель мыслить более методично.

Эффективное использование токенов: Такой подход демонстрирует лучшее соотношение точности к затратам токенов, особенно для сложных задач.

Этот промпт можно адаптировать для различных сценариев программирования, от простых алгоритмических задач до сложных проектов разработки.

№ 12. Цепь описаний: То, что я могу понять, я могу выразить словами

Ссылка: https://arxiv.org/pdf/2502.16137

Рейтинг: 85
Адаптивность: 90

Ключевые выводы:

Основная цель исследования - разработка и оценка новой стратегии промптинга для мультимодальных больших языковых моделей (MLLMs), названной Chain of Description (CoD). Главный результат: CoD-промптинг значительно улучшает производительность моделей по сравнению со стандартными методами, показывая улучшение почти на 4% в категории речи в аудио-бенчмарке AIR-Bench-Chat и на 5,3% в сложных задачах визуального бенчмарка MMMU_Pro.

Объяснение метода:

Chain-of-Description — простой, но эффективный метод промптинга, требующий от модели сначала описать мультимодальные входные данные перед ответом. Показывает значительные улучшения для сложных задач (до 5.3%), легко применим любым пользователем без технических знаний, и работает в стандартном интерфейсе чата без API или дообучения.

Ключевые аспекты исследования: 1. **Chain-of-Description (CoD) Prompting** — новый метод для работы с мультимодальными LLM (MLLM), который предполагает сначала генерацию детального описания входных данных (аудио или изображения), а затем ответа на вопрос.

Эффективность метода — исследование демонстрирует значительное улучшение производительности моделей при использовании CoD по сравнению со стандартным промптингом: улучшение на 4% для аудиомоделей в категории речи (AIR-Bench-Chat) и на 5.3% для моделей обработки изображений в сложных задачах (MMMU_Pro).

Зависимость от сложности задачи — CoD показывает наибольшую эффективность для сложных задач в визуальной модальности и для задач с высокой информационной плотностью (например, распознавание речи в аудио) по сравнению с более простыми задачами.

Качество описаний — ключевым фактором эффективности CoD является качество генерируемых описаний. Эксперименты показали, что более мощные модели генерируют лучшие описания, что приводит к улучшению результатов.

Теоретическое обоснование — метод основан на идее "что я могу понять, то могу

выразить словами", предполагая, что способность модели генерировать подробное описание входных данных указывает на более глубокое понимание.

Дополнение:

Применимость в стандартном чате без дообучения или API

Метод Chain-of-Description (CoD) **не требует дообучения или API** и может быть применен в стандартном чате с мультимодальными LLM. Исследователи использовали дообучение и API лишь для проведения систематической оценки, но сама техника полностью реализуема в обычном диалоговом режиме.

Концепции и подходы для стандартного чата

Двухэтапный промптинг: Пользователи могут напрямую запрашивать модель сначала описать входные данные, а затем ответить на вопрос. Например: Сначала детально опиши, что ты видишь на этом изображении/слышишь в этом аудио, а затем ответь на мой вопрос: [вопрос]

Адаптация для разных типов входных данных: Для изображений: "Опиши объекты, сцены, цвета, пространственные отношения" Для аудио: "Опиши речь, фоновые звуки, музыку, эмоциональный контекст"

Фокус на сложных задачах: Наибольшую пользу CoD приносит при сложных задачах и высокой информационной плотности, поэтому пользователям стоит применять этот метод именно в таких случаях.

Ожидаемые результаты

Улучшение точности ответов на сложные вопросы (до 5.3% для визуальных задач) **Повышение качества распознавания речи** в аудио (до 4% улучшения) **Более полное понимание контекста** мультимодальных данных **Снижение вероятности "галлюцинаций"** за счет более детального анализа входных данных Метод особенно эффективен, когда пользователь запрашивает информацию, которая не очевидна на первый взгляд или требует тщательного анализа деталей в изображении или аудио.

Prompt:

Применение Chain of Description (CoD) в промптах для GPT ## Суть метода CoD Chain of Description (CoD) - это стратегия промптинга, при которой мультимодальная модель сначала создает подробное описание входных данных (аудио/изображения), а затем использует это описание как основу для ответа на вопрос.

Пример промпта с использованием CoD для изображения

[======] Я применяю метод Chain of Description (CoD) для анализа изображения.

Пожалуйста:

Сначала создай подробное описание изображения, включая: Все видимые объекты Их пространственное расположение Цвета и текстуры Любые текстовые элементы Контекст сцены

Затем, используя это описание как основу, ответь на следующий вопрос: [Ваш вопрос об изображении]

Важно: создай максимально детальное описание перед ответом на вопрос. [======]

Как это работает

Разделение задачи на этапы: Модель сначала фокусируется только на описании входных данных, что позволяет ей лучше обработать и структурировать информацию.

Повышение информационной плотности: Согласно исследованию, CoD помогает модели извлечь больше релевантной информации из входных данных (например, ~4 токена описания в секунду для речи).

Улучшение понимания: Создавая явное описание, модель лучше "осознает" содержание входных данных, что особенно важно для сложных запросов.

Эффективность для сложных задач: Исследование показало улучшение на 5.3% для сложных визуальных задач и на 4% для обработки речи.

Когда использовать CoD-промптинг

• При работе со сложными визуальными сценами

• Для задач, требующих детального понимания контента

• Когда стандартный подход дает неудовлетворительные результаты

• В комбинации с другими техниками (например, Chain-of-Thought) для еще большего улучшения результатов

CoD особенно эффективен, потому что следует принципу "то, что я могу понять, я могу выразить словами" - заставляя модель сформулировать свое понимание, мы помогаем ей лучше обработать информацию.

№ 16. Профиль пользователя с большими языковыми моделями: создание, обновление и оценка

Ссылка: https://arxiv.org/pdf/2502.10660

Рейтинг: 85
Адаптивность: 90

Ключевые выводы:

Исследование направлено на разработку методологии построения и обновления пользовательских профилей с использованием больших языковых моделей (LLM). Основные результаты показывают, что модели Mistral-7b и Llama2-7b достигают высокой эффективности в обеих задачах, значительно улучшая точность и полноту генерируемых профилей.

Объяснение метода:

Исследование предлагает готовые методы создания и обновления пользовательских профилей с помощью LLM, с открытыми датасетами и четкой методологией, применимой для широкого спектра задач персонализации. Основные концепции доступны для реализации даже без специализированных технических знаний.Ключевые аспекты исследования 1. Создание и обновление пользовательских профилей с использованием LLM, представляя профиль как набор пар ключ-значение на основе текстовых данных о пользователе. 2. Разработка двух новых открытых наборов данных: один для построения профилей, другой для их обновления, что заполняет пробел в исследованиях профилирования пользователей. 3. Методология использования вероятностного подхода в LLM для прогнозирования атрибутов пользователей из текстовых данных с высокой точностью. 4. Экспериментальное сравнение различных моделей (Mistral-7b, Llama2-7b, и др.) для задач профилирования, оценивая их эффективность через метрики точности, полноты и F1-score. 5. Механизм динамического обновления профилей при появлении новой информации о пользователе, сохраняя актуальность и релевантность профиля.

Анализ практической применимости **1. Создание и обновление профилей с LLM:** - Прямая применимость: Пользователи могут использовать готовую методологию для автоматического извлечения структурированных профилей из неструктурированных текстов, что полезно в широком спектре задач от CRM до контент-рекомендаций. - Концептуальная ценность: Показывает, как современные LLM могут эффективно трансформировать текстовые описания в структурированные данные. - Потенциал для адаптации: Подход может быть адаптирован для различных типов текстового контента и для создания профилей разной сложности.

2. Открытые наборы данных: - Прямая применимость: Пользователи могут сразу использовать эти датасеты для тестирования своих подходов к профилированию. - Концептуальная ценность: Стандартизированные датасеты позволяют лучше понять, какие типы данных важны для профилирования. - Потенциал для адаптации: Датасеты могут служить основой для создания собственных, более специализированных наборов данных для конкретных областей.

3. Вероятностный подход в LLM: - Прямая применимость: Пользователи могут применить предложенную математическую модель для работы с неопределенностью в предсказании атрибутов пользователей. - Концептуальная ценность: Демонстрирует, как формализовать неопределенность в процессе профилирования. - Потенциал для адаптации: Модель может быть расширена для учета дополнительных факторов и источников данных.

4. Сравнительный анализ моделей: - Прямая применимость: Пользователи получают готовую информацию о том, какие модели лучше подходят для задач профилирования. - Концептуальная ценность: Понимание сильных и слабых сторон различных LLM для задач структурирования информации. - Потенциал для адаптации: Методология оценки может быть применена к другим моделям или задачам.

5. Механизм динамического обновления: - Прямая применимость: Предоставляет готовую методологию для поддержания актуальности профилей пользователей. - Концептуальная ценность: Демонстрирует важность включения временного аспекта в профилирование. - Потенциал для адаптации: Подход может быть адаптирован для различных сценариев обновления данных и разных скоростей изменения пользовательских предпочтений.

Сводная оценка полезности На основе проведенного анализа, исследование заслуживает оценку **85 из 100**. Исследование предоставляет готовые методы и концепции, которые могут быть немедленно применены широкой аудиторией пользователей LLM.

Контраргументы к оценке: 1. **Почему оценка могла бы быть выше:** Исследование предлагает открытые наборы данных и полностью описывает методологию, что делает ее исключительно доступной для практического применения. Также оно решает реальную проблему динамического обновления профилей. 2. **Почему оценка могла бы быть ниже:** Для полноценного использования предложенных методов требуется доступ к LLM и определенные навыки в ML/NLP, что может ограничить их применимость пользователями без технического бэкграунда. Также, методы могут быть избыточными для простых сценариев профилирования.

После рассмотрения этих аргументов, я подтверждаю оценку **85** как обоснованную, поскольку положительные аспекты (готовые методы, открытые датасеты) перевешивают ограничения.

Основные причины данной оценки: - Предоставление полноценных открытых наборов данных для тестирования и развития методов профилирования - Четкая практическая методология для создания и обновления профилей - Готовые показатели эффективности различных LLM для этих задач - Решение актуальной проблемы актуализации профилей с течением времени - Потенциал для адаптации методов в различных сценариях использования

Уверенность в оценке Очень сильная. Исследование предоставляет исчерпывающую информацию о методологии, результатах экспериментов и имеет явные практические применения.

Оценка адаптивности Оценка адаптивности: **90 из 100**

Основные принципы исследования (вероятностное моделирование профилей, обновление профилей на основе новой информации) могут быть легко адаптированы для использования в стандартных чат-интерфейсах LLM. Пользователи могут извлечь идею структурирования информации о себе или других в формате ключ-значение и применить ее даже без использования специализированных инструментов. Концепция динамического обновления профилей имеет высокий потенциал для улучшения персонализации взаимодействий с LLM в будущем. Специализированные методы статистического моделирования могут быть упрощены до простых промптов для получения структурированной информации из неструктурированного текста. || <Оценка: 85> || <Объяснение: Исследование предлагает готовые методы создания и обновления пользовательских профилей с помощью LLM, с открытыми датасетами и четкой методологией, применимой для широкого спектра задач персонализации. Основные концепции доступны для реализации даже без специализированных технических знаний.> || <Адаптивность: 90> ||

Дополнение

Применимость методов исследования в стандартном чате

Для работы методов из исследования **не требуется** дообучение или API. Хотя авторы использовали fine-tuning для получения лучших результатов, основные концепции могут быть применены в стандартном чате с LLM:

Структурированное профилирование: Можно использовать промпты, которые инструктируют LLM извлекать из текста структурированную информацию в формате ключ-значение. Например:

Prompt:

Применение исследования о профилях пользователей в промптах для GPT Исследование о создании и обновлении пользовательских профилей с

использованием LLM предоставляет ценный фреймворк, который можно адаптировать для создания эффективных промптов в GPT. Ключевые идеи исследования, применимые к промптам

Структурированное представление информации в виде пар ключ-значение Вероятностная модель для извлечения профилей из текста Механизм обновления профилей с интеграцией новой информации Форматирование вывода для точного представления данных

Пример промпта для GPT markdownCopy# Запрос на создание профиля пользователя

Контекст Ты - ассистент, который создает структурированный профиль пользователя из текстовой информации. Используй вероятностную модель извлечения данных - выделяй только ту информацию, которая явно указана в тексте, не выдумывай дополнительные детали.

Инструкция Внимательно проанализируй биографический текст ниже и создай профиль пользователя в формате ключ-значение. Включи следующие категории (если информация доступна): - Name: [имя] - Profession: [профессия/занятия] - BirthDate/BirthPlace: [дата/место рождения] - Education: [образование] - Likes: [интересы и предпочтения] - Dislikes: [что не нравится] - Hobbies: [хобби и увлечения] - Achievements: [достижения] - Location: [текущее место проживания]

Биографический текст [Ваш текст здесь]

Формат вывода Представь профиль в четко структурированном формате ключ-значение. Не добавляй информацию, которой нет в тексте. Если информация по какой-либо категории отсутствует, не включай эту категорию в профиль. Как это работает

Вероятностный подход: Промпт инструктирует модель использовать только информацию, которая явно присутствует в тексте, что соответствует вероятностному фреймворку исследования. Структурирование данных: Формат ключ-значение из исследования применяется для организации извлеченной информации. Гибкость категорий: Мы указываем модели включать только те категории, по которым есть данные, что соответствует подходу в исследовании. Точность вывода: Инструкция не добавлять отсутствующую информацию соответствует принципу P(y|x), где модель предсказывает профиль только на основе имеющихся данных.

Для обновления профиля Для обновления существующего профиля можно адаптировать исследование, предоставив модели и текущий профиль, и новую информацию: markdownCopy## Инструкция для обновления профиля Обнови существующий профиль пользователя на основе новой информации. Сохрани существующие данные, если они не противоречат новой информации, и интегрируй новые данные для создания обновленного профиля.

Существующий профиль [Профиль в формате ключ-значение]

Новая информация [Текст с новыми данными] Этот подход основан на механизме P(y^u|x^u,y;ζ) из исследования, где модель учится переходить от существующего профиля к обновленному на основе новой информации.

№ 20. Время имеет значение: Как использование больших языковых моделей в разное время влияет на восприятие писателей и результаты идейной деятельности в условиях поддержки ИИ

Ссылка: https://arxiv.org/pdf/2502.06197

Рейтинг: 85
Адаптивность: 90

Ключевые выводы:

Исследование изучает, как разное время использования LLM (до или после самостоятельной генерации идей) влияет на восприятие и результаты идеации пользователей. Основной вывод: использование LLM после самостоятельной идеации приводит к большей автономии, чувству владения результатом и самоэффективности, а также к более оригинальным идеям, чем использование LLM с самого начала процесса.

Объяснение метода:

Исследование предлагает непосредственно применимый метод повышения эффективности работы с LLM - сначала самостоятельная генерация идей, затем использование LLM. Это повышает оригинальность мышления, чувство автономии и собственности над идеями. Метод не требует специальных инструментов и может использоваться любым пользователем в повседневной работе с LLM для различных творческих задач.

Ключевые аспекты исследования: 1. **Влияние времени использования LLM на результаты генерации идей**: Исследование сравнивает использование LLM для генерации идей в двух временных точках - до самостоятельной генерации идей пользователем (Cbefore) и после самостоятельной работы (Cafter).

Влияние на автономию и чувство собственности: Использование LLM после самостоятельной генерации идей приводит к более высокому уровню воспринимаемой автономии, чувству собственности над идеями и творческой самоэффективности.

Фиксация на идеях LLM: Раннее использование LLM приводит к большему сходству между идеями пользователя и идеями, предложенными LLM, что указывает на "фиксацию идей" и снижение оригинальности мышления.

Механизм медиации: Исследование выявило, что автономия является ключевым

медиатором между временем использования LLM и результатами генерации идей, влияя на чувство собственности и самоэффективность.

Распределение заслуг: Участники, использовавшие LLM с самого начала, приписывали больше заслуг ИИ и меньше себе, в то время как участники, использовавшие LLM после собственной работы, приписывали больше заслуг себе.

Дополнение:

Применимость в стандартном чате без дообучения или API

Методы и подходы исследования полностью применимы в стандартном чате с LLM без необходимости в дообучении или API. Исследователи использовали API только для удобства проведения эксперимента и сбора данных, но сами концепции и подходы не зависят от этого.

Ключевые концепции для применения в стандартном чате:

Отложенное использование LLM: Пользователь может самостоятельно внедрить практику сначала генерировать собственные идеи, записывать их, и только потом обращаться к LLM для расширения или улучшения этих идей.

Сохранение автономии: Пользователь может сознательно формулировать запросы к LLM таким образом, чтобы модель дополняла его идеи, а не заменяла их (например, "Помоги мне расширить следующие идеи, которые я уже сформулировал...").

Избегание фиксации идей: Пользователь может сначала записать свои мысли без обращения к LLM, чтобы избежать преждевременной фиксации на идеях, предложенных моделью.

Стратегическое распределение задач: Пользователь может использовать LLM на этапе конвергентного мышления (структурирование и организация идей), а не на этапе дивергентного мышления (генерация разнообразных идей).

Ожидаемые результаты при применении этих концепций:

• Более оригинальные и разнообразные идеи

• Повышенное чувство собственности над результатами работы

• Более высокая творческая самоэффективность

• Более сбалансированное распределение заслуг между собой и LLM

• Снижение риска чрезмерной зависимости от LLM в творческих процессах

Эти подходы не требуют никаких специальных инструментов или технических

знаний и могут быть немедленно внедрены любым пользователем в обычном чате с LLM.

Prompt:

Использование исследования о времени взаимодействия с LLM в промптах ## Ключевой вывод исследования

Исследование показывает, что **время использования LLM** имеет значительное влияние на качество идей и восприятие пользователем собственной работы:

• Использование LLM после самостоятельной идеации => более оригинальные идеи, выше автономия, чувство владения результатом и самоэффективность

• Использование LLM с самого начала => снижение оригинальности, творческой самоэффективности и автономии

Пример промпта на основе исследования

[======] Я работаю над [описание задачи/проекта]. Чтобы максимизировать оригинальность идей и сохранить чувство владения результатом, я буду использовать двухэтапный подход:

ЭТАП 1: Мои собственные идеи (которые я уже сгенерировал): [перечислите ваши идеи, которые вы придумали самостоятельно]

ЭТАП 2: Теперь, основываясь на моих исходных идеях, помоги мне: 1. Расширить и улучшить мои существующие идеи 2. Предложить 2-3 дополнительных направления, которые я мог упустить 3. Помочь структурировать и организовать все идеи в логичную систему

Важно: пожалуйста, сохрани основу моих оригинальных идей, предлагая улучшения, а не полностью новые решения. [======]

Почему это работает

Этот промпт применяет ключевые выводы исследования:

Сначала самостоятельная идеация - вы генерируете собственные идеи до обращения к LLM **LLM как помощник, а не основной генератор** - модель расширяет ваши идеи, а не создает их с нуля **Структурированный двухэтапный подход** - разделение на дивергентное (ваше) и конвергентное (с помощью LLM) мышление **Сохранение автономии** - явное указание модели уважать и развивать ваши идеи, а не заменять их Такой подход позволяет получить преимущества LLM, минимизируя негативное влияние на вашу творческую самоэффективность и оригинальность мышления.

№ 24. Применение максима Грайса в цикле взаимодействия человек-ИИ: дизайнерские идеи из участнического подхода

Ссылка: https://arxiv.org/pdf/2503.00858

Рейтинг: 85
Адаптивность: 90

Ключевые выводы:

Исследование направлено на применение принципов коммуникации Грайса (Gricean Maxims) к взаимодействию человека с большими языковыми моделями (LLM). Основная цель - разработать дизайн-рекомендации для улучшения взаимодействия человек-LLM на основе этих принципов. Главные результаты включают 9 дизайн-рекомендаций, сгруппированных по трем стадиям цикла взаимодействия, и переосмысление максим Грайса в контексте взаимодействия с LLM.

Объяснение метода:

Исследование предлагает 9 практических рекомендаций по дизайну взаимодействия с LLM, основанных на максимах Грайса. Эти рекомендации структурированы по циклу взаимодействия (формулирование цели, генерация ответа, оценка результата) и могут быть немедленно применены пользователями через стратегии составления промптов. Исследование объединяет теоретические основы коммуникации с практическими потребностями, учитывая разные уровни пользователей.

Ключевые аспекты исследования: 1. **Применение максим Грайса к взаимодействию человека и LLM**: Исследование адаптирует классические принципы эффективной коммуникации (максимы Грайса: количество, качество, отношение, способ) к контексту взаимодействия человека с языковыми моделями.

Реинтерпретация максим для контекста LLM: Авторы переосмыслили каждую максиму с учетом особенностей взаимодействия с LLM, например, максима количества расширена до оптимизации когнитивной нагрузки пользователя.

Девять конкретных рекомендаций по дизайну: На основе партисипативных воркшопов с экспертами по коммуникации, дизайнерами интерфейсов и опытными пользователями LLM были сформулированы 9 практических рекомендаций.

Структурирование рекомендаций по циклу взаимодействия: Авторы распределили рекомендации по трем стадиям взаимодействия человек-LLM: формулирование цели, интерпретация и выполнение, оценка результата.

Конкретные функции дизайна: Для каждой стадии взаимодействия разработаны конкретные элементы дизайна, которые могут быть реализованы в интерфейсах взаимодействия с LLM.

Дополнение: Исследование не требует дообучения или специального API для применения его методов. Большинство принципов и подходов могут быть успешно адаптированы для использования в стандартном чате с LLM.

Концепции и подходы, применимые в стандартном чате:

Структурирование запросов по максиме Количества: Запрашивать иерархические ответы: "Предоставь ответ в иерархической форме, сначала основные пункты, затем детали по каждому из них" Указывать желаемый уровень детализации: "Дай краткий обзор в 3 пункта, а затем подробно раскрой пункт X"

Улучшение качества ответов (максима Качества):

Просить LLM объяснять свои рассуждения: "Объясни, каким образом ты пришел к этому выводу" Запрашивать план выполнения задачи: "Перед ответом, опиши как ты планируешь подойти к решению этой задачи"

Повышение релевантности (максима Отношения):

Явно указывать контекст и цель: "Учитывая наш предыдущий разговор о X, помоги мне с Y" Проверять понимание контекста: "Перечисли ключевые моменты нашего разговора, которые ты учитываешь при ответе"

Улучшение способа представления информации (максима Способа):

Указывать предпочтительный формат ответа: "Представь ответ в виде таблицы/списка/схемы" Запрашивать выделение ключевых моментов: "Выдели наиболее важные части ответа"

Ожидаемые результаты применения:

Повышение эффективности взаимодействия - более четкие, структурированные и релевантные ответы LLM **Снижение когнитивной нагрузки** - лучшая организация информации, облегчающая её восприятие и использование **Повышение доверия к ответам LLM** - благодаря объяснению рассуждений и прозрачности процесса **Улучшение контекстуальной релевантности** - более точное соответствие ответов намерениям пользователя **Большая гибкость в форматировании ответов** - адаптация представления информации к конкретным задачам Эти подходы не требуют технических модификаций модели и могут быть реализованы через обычные текстовые запросы в любом стандартном интерфейсе чата с LLM.

Анализ практической применимости: 1. **Применение максим Грайса к взаимодействию человека и LLM** - **Прямая применимость**: Высокая.

Пользователи могут немедленно использовать эти принципы для формулирования более эффективных запросов и оценки ответов LLM. - **Концептуальная ценность**: Очень высокая. Предоставляет теоретическую основу для понимания, почему некоторые взаимодействия с LLM успешны, а другие нет. - **Потенциал для адаптации**: Высокий. Эти принципы универсальны и могут быть применены к любому типу взаимодействия с LLM.

Реинтерпретация максим для контекста LLM Прямая применимость: Средняя. Требует некоторого понимания теории коммуникации, но предлагает конкретные рекомендации. **Концептуальная ценность**: Высокая. Помогает пользователям понять, какие аспекты коммуникации с LLM отличаются от человеческой коммуникации. **Потенциал для адаптации**: Высокий. Эти реинтерпретации могут быть использованы для разработки персональных стратегий взаимодействия с LLM.

Девять конкретных рекомендаций по дизайну

Прямая применимость: Очень высокая. Рекомендации конкретны и могут быть немедленно применены пользователями при составлении запросов. **Концептуальная ценность**: Высокая. Помогает пользователям систематизировать подход к взаимодействию с LLM. **Потенциал для адаптации**: Высокий. Рекомендации могут быть адаптированы к различным задачам и контекстам использования.

Структурирование рекомендаций по циклу взаимодействия

Прямая применимость: Высокая. Пользователи могут легко определить, какие рекомендации применять на каждой стадии взаимодействия. **Концептуальная ценность**: Очень высокая. Предоставляет системный подход к взаимодействию с LLM. **Потенциал для адаптации**: Высокий. Структура применима к любому типу взаимодействия с LLM.

Конкретные функции дизайна

Прямая применимость: Средняя. Некоторые функции могут быть реализованы пользователями через промпты, но другие требуют изменений в интерфейсе. **Концептуальная ценность**: Высокая. Демонстрирует, как теоретические принципы могут быть воплощены в конкретные решения. **Потенциал для адаптации**: Средний. Пользователи могут адаптировать некоторые идеи дизайна через стратегии составления промптов. Сводная оценка полезности: На основе анализа я оцениваю полезность этого исследования для широкой аудитории в **85 баллов из 100**.

Исследование предлагает исключительно полезную теоретическую основу и практические рекомендации, которые могут быть немедленно применены пользователями LLM разного уровня подготовки. Девять конкретных рекомендаций по дизайну и их структурирование по циклу взаимодействия предоставляют готовую систему для более эффективной коммуникации с LLM.

Контраргументы к высокой оценке: 1. Исследование опирается на теорию коммуникации (максимы Грайса), которая может быть не знакома обычным пользователям, что затрудняет полное понимание некоторых рекомендаций. 2. Некоторые предложенные функции дизайна требуют изменений в интерфейсе LLM, которые пользователи не могут реализовать самостоятельно.

Контраргументы к низкой оценке: 1. Даже без глубокого понимания теории коммуникации, пользователи могут непосредственно применять конкретные рекомендации и видеть улучшение результатов. 2. Многие рекомендации могут быть адаптированы и реализованы через стратегии составления промптов, без необходимости изменения интерфейса.

После рассмотрения этих аргументов, я сохраняю оценку в **85 баллов**, так как практическая ценность и универсальность рекомендаций перевешивают необходимость некоторой адаптации и базовых знаний.

Эта оценка обоснована следующими факторами: 1. Исследование предоставляет конкретные, практически применимые рекомендации для улучшения взаимодействия с LLM. 2. Рекомендации структурированы по стадиям взаимодействия, что облегчает их применение. 3. Исследование объединяет теоретические основы коммуникации с практическими потребностями пользователей. 4. Многие рекомендации могут быть немедленно применены через стратегии составления промптов. 5. Исследование учитывает разные уровни пользователей и разнообразие задач.

Уверенность в оценке: Моя уверенность в оценке **очень сильная**.

Причины высокой уверенности: 1. Исследование предоставляет четкие, структурированные рекомендации, основанные на хорошо изученной теории коммуникации. 2. Методология исследования включает участие трех типов экспертов: специалистов по коммуникации, дизайнеров интерфейсов и опытных пользователей LLM, что обеспечивает комплексный взгляд на проблему. 3. Рекомендации конкретны, практичны и структурированы по стадиям взаимодействия, что облегчает их оценку и применение. 4. Исследование напрямую адресует проблемы, с которыми сталкиваются пользователи при взаимодействии с LLM. 5. Результаты исследования согласуются с передовыми практиками в области HCI и дизайна взаимодействия.

Оценка адаптивности: Я оцениваю адаптивность исследования в **90 баллов из 100**.

Универсальность принципов: Максимы Грайса и их реинтерпретация для LLM представляют универсальные принципы коммуникации, которые могут быть применены в любом контексте взаимодействия с LLM, включая обычные чаты.

Применимость рекомендаций через промпты: Большинство рекомендаций могут быть реализованы через стратегии составления промптов. Например, пользователи могут:

Просить LLM предоставить план выполнения задачи перед генерацией ответа (DC2) Запрашивать иерархическую структуру ответа (DC6) Указывать желаемый формат ответа (DC7) Просить выделить ключевые моменты или изменения (DC4)

Концептуальная адаптация: Исследование предлагает концептуальную основу для понимания взаимодействия с LLM, которая может быть использована для разработки персональных стратегий, независимо от конкретного интерфейса.

Потенциал для будущих взаимодействий: Рекомендации предвосхищают направления развития интерфейсов LLM и могут быть использованы для формирования ожиданий и запросов к будущим системам.

Высокая оценка адаптивности обоснована тем, что исследование фокусируется на фундаментальных принципах коммуникации, которые универсальны и могут быть применены в различных контекстах, независимо от конкретного интерфейса или технической реализации LLM.

|| <Оценка: 85> || <Объяснение: Исследование предлагает 9 практических рекомендаций по дизайну взаимодействия с LLM, основанных на максимах Грайса. Эти рекомендации структурированы по циклу взаимодействия (формулирование цели, генерация ответа, оценка результата) и могут быть немедленно применены пользователями через стратегии составления промптов. Исследование объединяет теоретические основы коммуникации с практическими потребностями, учитывая разные уровни пользователей.> || <Адаптивность: 90>

Prompt:

Применение принципов Грайса в промптах для GPT
Краткое объяснение

Исследование показывает, как принципы коммуникации Грайса можно применить для улучшения взаимодействия с языковыми моделями. Эти принципы помогают структурировать промпты таким образом, чтобы получать более точные, релевантные и полезные ответы.

Пример промпта с применением принципов Грайса

[======]

Запрос по финансовому анализу
Контекст и цель

Я финансовый аналитик, готовлю квартальный отчет для руководства компании среднего размера в сфере розничной торговли. Мне нужен анализ влияния

сезонности на продажи.

Желаемый формат ответа

• Начни с краткого резюме (3-4 предложения)

• Затем представь основные пункты в виде маркированного списка

• После этого дай более подробный анализ каждого пункта

• Заверши рекомендациями для бизнеса

Параметры

• Уровень детализации: средний (для руководителей, не экспертов)

• Тон: профессиональный, но доступный

• Объем: не более 500 слов

• Роль: выступи в качестве опытного финансового консультанта

План выполнения задачи

Пожалуйста, перед ответом: 1. Определи ключевые факторы сезонности в розничной торговле 2. Структурируй анализ по кварталам 3. Учти как внешние, так и внутренние факторы

Если тебе не хватает каких-то данных, укажи, какую информацию стоило бы добавить для более точного анализа. [======]

Как этот промпт использует принципы из исследования

Максима Количества: Четко задает объем и уровень детализации ответа, использует иерархическую структуру (от резюме к деталям).

Максима Качества: Определяет роль модели и задает параметры профессионального тона, что способствует более достоверным ответам.

Максима Отношения: Предоставляет четкий контекст и цель, что помогает модели генерировать релевантный контент.

Максима Способа: Задает конкретную структуру ответа и просит предварительный план выполнения задачи, что делает процесс более прозрачным.

Такой подход к составлению промптов позволяет получать более структурированные, релевантные и полезные ответы от GPT, минимизируя недопонимание и повышая эффективность взаимодействия.

№ 28. Диверсификация выборки улучшает инференс ScalingLLM

Ссылка: https://arxiv.org/pdf/2502.11027

Рейтинг: 82
Адаптивность: 90

Ключевые выводы:

Исследование направлено на улучшение эффективности вывода LLM путем повышения разнообразия генерируемых ответов. Основная идея заключается в том, что однообразие выходных данных LLM приводит к неэффективному сэмплированию, поскольку модели повторно генерируют похожие, но неточные ответы. Авторы предлагают метод DivSampling, который вносит разнообразие в промпты, что значительно улучшает точность решений при масштабировании вывода.

Объяснение метода:

Исследование предлагает простые в применении методы диверсификации запросов (Role, Instruction, переформулирование), которые значительно улучшают качество ответов LLM. Пользователи любого уровня могут немедленно применить эти техники, не требующие API или специальных знаний. Методы универсальны для разных задач, показали эмпирически подтвержденную эффективность и имеют теоретическое обоснование.

Ключевые аспекты исследования: 1. **Диверсифицированная выборка (DivSampling)** - метод улучшения качества ответов LLM путем внесения разнообразия в запросы для получения более вариативных ответов. Исследование выявило связь между разнообразием ответов и их точностью.

Подходы к диверсификации запросов - предложены два типа стратегий: не зависящие от задачи (task-agnostic) и специфичные для задачи (task-specific) методы внесения разнообразия в промпты.

Task-agnostic подходы включают три техники: Jabberwocky (вставка фрагментов поэмы), Role (добавление ролевых описаний) и Instruction (добавление конкретных инструкций).

Task-specific подходы включают Random Idea Injection (генерация идей для решения задачи) и Random Query Rephrase (переформулирование запроса).

Теоретическое обоснование - доказано, что диверсификация запросов существенно снижает долю ошибок в ответах LLM при масштабировании вывода.

Дополнение: Методы исследования **не требуют дообучения или специального API** для применения в стандартном чате. Авторы использовали API для экспериментального подтверждения эффективности, но сами концепции полностью применимы в любом стандартном интерфейсе LLM.

Основные концепции и подходы, которые можно внедрить в стандартном чате:

Добавление ролевых описаний (Role Injection) - пользователь может добавлять к запросам различные роли для модели, например: "Ты аналитик, который фокусируется на деталях" или "Ты исследователь, который рассматривает проблему с разных сторон".

Добавление инструкций (Instruction Injection) - пользователь может включать в запрос конкретные инструкции по решению задачи, например: "Раздели задачу на логические шаги" или "Используй наглядные примеры в объяснении".

Переформулирование вопросов (Query Rephrase) - пользователь может задать один и тот же вопрос несколькими способами и сравнить ответы.

Генерация идей (Idea Injection) - пользователь может сначала попросить модель предложить несколько подходов к решению, а затем использовать эти идеи в последующих запросах.

Ожидаемые результаты: - Более разнообразные и качественные ответы - Снижение вероятности "застревания" модели в неоптимальных решениях - Повышение точности ответов на сложные вопросы, особенно в задачах рассуждения, математики и программирования - Возможность выбора лучшего ответа из нескольких альтернатив

Эти методы особенно эффективны при решении сложных задач, где первое предложенное решение может быть неоптимальным.

Prompt:

Использование методов диверсификации выборки в промптах для GPT ## Ключевые знания из исследования

Исследование показало, что разнообразие в промптах значительно улучшает точность ответов LLM. Метод DivSampling предлагает несколько подходов:

Задаче-агностические подходы: Role, Instruction, Jabberwocky
Задаче-специфические подходы: Random Idea Injection, Random Query Rephrase
Комбинированные методы: сочетание различных подходов для максимального эффекта ## Пример промпта с применением методов диверсификации

[======] # Промпт с использованием Role Injection + Random Idea Injection

Роль Ты опытный инженер-оптимизатор, специализирующийся на эффективных алгоритмах и нестандартных решениях сложных задач. Твой подход характеризуется систематическим анализом и поиском оптимальных решений.

Случайная идея для вдохновения Рассмотри концепцию динамического программирования и кэширования промежуточных результатов как потенциальный подход к решению.

Задача Разработай алгоритм для нахождения наибольшей общей подпоследовательности двух строк с оптимальной временной и пространственной сложностью.

Инструкции 1. Проанализируй проблему 2. Предложи несколько различных подходов к решению 3. Выбери наиболее эффективный подход и объясни его преимущества 4. Предоставь псевдокод или реализацию на Python 5. Проанализируй временную и пространственную сложность твоего решения [======]

Как это работает

Role Injection задает конкретную роль (инженер-оптимизатор), что направляет модель на генерацию ответов с определенной перспективы и стилем мышления.

Random Idea Injection предоставляет дополнительный контекст и идею (динамическое программирование), которая может направить мышление модели в продуктивном направлении.

Структурированные инструкции обеспечивают четкий формат ответа, что также способствует разнообразию и полноте генерируемого контента.

Такой подход, согласно исследованию, может привести к значительному улучшению качества ответов (до 15-75% в зависимости от задачи) по сравнению с обычными промптами без диверсификации.

Для получения максимального эффекта можно комбинировать несколько методов диверсификации и создавать несколько вариантов промптов для одной задачи.

№ 32. Концептуально-ориентированное побуждение цепочки мыслей для парного сравнительного оценки текстов с использованием больших языковых моделей

Ссылка: https://arxiv.org/pdf/2310.12049

Рейтинг: 82
Адаптивность: 90

Ключевые выводы:

Исследование представляет новый фреймворк для оценки текстов с использованием больших языковых моделей (LLM), который позволяет эффективно анализировать латентные концепции в текстах. Основная цель - создание метода, который преобразует попарные сравнения текстов из задачи рассуждения в задачу распознавания паттернов с помощью подхода Concept Guided Chain of Thought (CGCoT). Результаты показывают, что этот метод превосходит существующие неконтролируемые методы оценки текста и сопоставим с контролируемыми подходами, требующими значительно больше размеченных данных.

Объяснение метода:

Исследование предлагает практический метод анализа текстов с помощью LLM, который не требует больших размеченных данных. CGCoT-подход (поэтапные направленные вопросы) и попарные сравнения легко адаптируются для различных задач и доступны широкой аудитории. Метод показывает высокую эффективность при минимальных затратах на разработку, хотя полная реализация требует некоторых технических знаний.

Ключевые аспекты исследования: 1. **Концепция CGCoT (Concept-Guided Chain-of-Thought)** - авторы предлагают новый подход к оценке текстов с использованием LLM, где модель анализирует тексты через серию последовательных вопросов, разработанных исследователем, для выделения конкретных аспектов интересующего концепта.

Попарное сравнение текстов - вместо прямой оценки текстов по шкале, авторы используют попарные сравнения между "концептуальными разбивками" текстов, превращая сложную задачу рассуждения в задачу распознавания паттернов.

Модель Брэдли-Терри - для преобразования результатов попарных сравнений в числовые оценки используется вероятностная модель, которая позволяет ранжировать тексты по степени выраженности целевого концепта.

Применение к оценке политической неприязни - методология была применена для измерения степени неприязни к политическим партиям в твитах, показав высокую корреляцию с оценками людей и превзойдя другие неконтролируемые методы анализа текста.

Минимальная потребность в размеченных данных - метод требует лишь небольшой пилотный набор размеченных примеров для разработки CGCoT-промптов, в отличие от традиционных методов, требующих тысячи размеченных примеров.

Дополнение:

Применимость в стандартном чате без дообучения или API

Не требуется дообучение или API: Методы исследования полностью применимы в стандартном чате с LLM. Хотя авторы использовали GPT-3.5 через API для автоматизации процесса, сама методология CGCoT и попарных сравнений может быть реализована через обычный интерфейс чата.

Ключевые концепции для применения в стандартном чате

Поэтапное структурирование запросов (CGCoT): Разбивка сложной задачи на последовательность простых вопросов Использование ответов на предыдущие вопросы как контекст для последующих Пример: При анализе текста сначала попросить LLM резюмировать его, затем выделить ключевые объекты, затем определить отношение к этим объектам

Попарное сравнение вместо прямой оценки:

Просить LLM сравнивать два текста по определенному критерию вместо прямой оценки по шкале Это соответствует сильным сторонам LLM (распознавание паттернов) и минимизирует слабости (прямая количественная оценка)

Использование "концептуальных разбивок":

Создание подробного анализа текста с помощью серии вопросов перед сравнением Это переводит задачу из области рассуждения в область распознавания паттернов
Ожидаемые результаты при применении в стандартном чате

• Повышение точности анализа текстов по сложным концептам

• Возможность работы с короткими текстами (твиты, комментарии)

• Минимизация потребности в размеченных данных

• Более последовательные и обоснованные оценки текстов

Пример использования в стандартном чате

Для анализа эмоциональной окраски отзыва о продукте: 1. "Резюмируй этот отзыв." 2. "Какие аспекты продукта упоминаются в отзыве?" 3. "Для каждого аспекта определи, выражено ли положительное или отрицательное отношение." 4. "Исходя из предыдущих ответов, насколько сильно в отзыве выражено [целевое отношение]?"

Затем можно сравнивать результаты анализа разных отзывов попарно.

Prompt:

Применение исследования CGCoT в промптах для GPT ## Ключевая концепция Исследование представляет метод **Concept Guided Chain of Thought (CGCoT)**, который превращает сложную оценку текста в структурированный анализ через концептуальные разбивки и попарные сравнения.

Пример промпта на основе CGCoT

[======] # Промпт для анализа эмоциональной окраски текстов

Шаг 1: Концептуальная разбивка Проанализируй следующий текст с точки зрения следующих концепций: 1. Использование эмоционально окрашенных слов 2. Наличие негативных стереотипов 3. Степень выраженности агрессии 4. Использование сарказма/иронии 5. Наличие призывов к действию

Текст для анализа: "[ВСТАВИТЬ ТЕКСТ]"

Для каждой концепции: - Определи ее наличие (да/нет) - Оцени интенсивность (низкая/средняя/высокая) - Приведи конкретные примеры из текста

Шаг 2: Сравнительная оценка Теперь сравни этот анализ с предыдущим текстом, который мы анализировали. Какой из текстов содержит более выраженную негативную окраску? Объясни свое решение, опираясь на концептуальную разбивку, а не на общее впечатление. [======]

Как работает CGCoT в этом промпте

Структурированная декомпозиция — вместо прямой оценки текста мы разбиваем анализ на конкретные концепции **Цепочка рассуждений** — модель вынуждена последовательно анализировать каждый аспект **Попарное сравнение** — сравнение по концепциям, а не по целым текстам, делает оценку более точной **Объяснимость** — получаем не только оценку, но и обоснование, опирающееся на конкретные элементы текста ## Преимущества такого подхода

• Точность — снижает влияние предвзятости модели, фокусируясь на конкретных

аспектах

• Прозрачность — обоснования решений понятны и проверяемы

• Гибкость — можно адаптировать концептуальную разбивку под конкретную задачу

• Минимальная потребность в обучении — не требует размеченных данных

Этот подход особенно полезен для сложных субъективных оценок, где простой промпт может давать непоследовательные результаты.

№ 36. За пределами цепочки размышлений: Обзор парадигм Chain-of-X для больших языковых моделей

Ссылка: https://arxiv.org/pdf/2404.15676

Рейтинг: 82
Адаптивность: 90

Ключевые выводы:

Исследование представляет собой комплексный обзор методов Chain-of-X (CoX), которые являются расширением концепции Chain-of-Thought (CoT) для больших языковых моделей (LLM). Основная цель - систематизировать и категоризировать различные методы CoX по типам узлов цепочки и областям применения, а также выявить их потенциал для решения разнообразных задач.

Объяснение метода:

Исследование предоставляет всеобъемлющую таксономию Chain-of-X методов, большинство из которых можно применить в повседневном взаимодействии с LLM. Особенно ценны концепции декомпозиции проблем, структурирования промежуточных шагов и механизмов самопроверки. Некоторые методы требуют технических знаний, что снижает доступность для неспециалистов, однако общие принципы легко адаптируются для стандартных чатов.

Ключевые аспекты исследования: 1. **Таксономия Chain-of-X (CoX)**: Исследование представляет систематическую классификацию методов Chain-of-X, расширяющих концепцию Chain-of-Thought. Выделяются четыре основных типа узлов в цепочке: промежуточные шаги, аугментация, обратная связь и модели.

Разнообразие применений: Авторы анализируют применение CoX в различных областях: мультимодальное взаимодействие (текст-изображение, текст-таблица, текст-код, текст-речь), уменьшение галлюцинаций, многошаговые рассуждения, выполнение инструкций, агенты на основе LLM и инструменты оценки.

Варианты промежуточных шагов: Детальное описание различных типов промежуточных элементов в цепочке рассуждений: декомпозиция проблемы, композиция знаний, инструкции и история взаимодействий.

Способы аугментации: Исследование описывает различные методы расширения возможностей LLM через цепочки инструкций, извлечение информации, использование внешних инструментов и исторических данных.

Механизмы обратной связи: Анализируются подходы к использованию внешней и

самогенерируемой обратной связи для улучшения качества ответов LLM.

Дополнение:

Большинство методов Chain-of-X, описанных в исследовании, не требуют дообучения или специального API и могут быть применены в стандартном чате с LLM. Хотя некоторые исследователи использовали специальные техники для своего удобства или для количественной оценки, основные концепции можно реализовать через грамотное структурирование промптов.

Концепции для применения в стандартном чате:

Chain-of-Thought (CoT) - базовый подход "давай думать пошагово", который можно применять без дополнительных инструментов.

Декомпозиция проблемы - разбиение сложной задачи на подзадачи, реализуемое через структурированный промпт.

Механизмы самопроверки - например, Chain-of-Verification, где модель сначала генерирует ответ, затем формулирует вопросы для проверки и исправляет ошибки.

Chain-of-Instructions - последовательное создание и выполнение инструкций для сложных задач.

Self-Refine - итеративное улучшение собственных ответов через критический анализ.

Ожидаемые результаты:

• Повышение точности решения сложных задач

• Уменьшение галлюцинаций через механизмы проверки

• Более структурированные и обоснованные ответы

• Улучшенное выполнение многошаговых инструкций

• Более эффективное решение задач, требующих логических рассуждений

Даже без специальных API или дообучения, правильное применение этих концепций может значительно повысить эффективность взаимодействия с LLM в стандартном чате.

Prompt:

Использование методов Chain-of-X в промптах для GPT ## Ключевые принципы из исследования

Исследование "За пределами цепочки размышлений" предоставляет систематизацию различных методов Chain-of-X (CoX), которые можно эффективно применять при составлении промптов для GPT. Эти методы позволяют значительно улучшить качество генерируемых ответов через структурированные подходы к решению сложных задач.

Пример промпта с использованием Chain-of-Verification

[======] Я хочу, чтобы ты решил следующую задачу по финансовому планированию, используя метод Chain-of-Verification:

Задача: Семья Ивановых хочет накопить 2 миллиона рублей за 5 лет для первоначального взноса по ипотеке. Их ежемесячный доход составляет 150,000 рублей, а расходы - 120,000 рублей. Какую сумму им нужно ежемесячно откладывать, и какой годовой процент доходности инвестиций необходим для достижения цели?

Пожалуйста, выполни следующие шаги: 1. Сначала предложи первоначальное решение задачи. 2. Составь список из 3-5 проверочных вопросов для верификации своего решения. 3. Ответь на каждый из этих вопросов, проверяя свои вычисления и предположения. 4. На основе проведенной самопроверки предоставь улучшенное, окончательное решение. 5. Укажи, какие коррекции были внесены и почему. [======]

Объяснение применения метода

В этом примере я использовал **Chain-of-Verification** (цепочка верификации) - один из методов CoX из категории "цепочки обратной связи". Этот метод работает следующим образом:

Декомпозиция процесса решения - промпт разбивает сложную финансовую задачу на последовательные шаги **Самопроверка** - модель генерирует проверочные вопросы для своего первоначального решения **Итеративное улучшение** - на основе ответов на проверочные вопросы модель корректирует свое решение **Прозрачность рассуждений** - весь процесс верификации доступен пользователю, что повышает доверие к результату Преимущество такого подхода в том, что он значительно снижает вероятность ошибок в вычислениях и логических рассуждениях, позволяя модели самостоятельно выявлять и исправлять недостатки в своем первоначальном ответе.

Другие возможные применения CoX в промптах

• Chain-of-Thought: для задач, требующих пошагового логического рассуждения

• Chain-of-Knowledge: для получения фактологически точных ответов с опорой на конкретные источники

• Chain-of-Experts: для задач, требующих знаний из разных областей, имитируя диалог специалистов

• Chain-of-Code: для решения программистских задач с пошаговой разработкой и отладкой

Каждый из этих методов можно адаптировать под конкретные задачи, создавая более эффективные промпты для GPT и получая более качественные и надежные результаты.

№ 40. Понимание перед разумом: улучшение цепочки размышлений с помощью итеративного суммирования в преднастройке

Ссылка: https://arxiv.org/pdf/2501.04341

Рейтинг: 78
Адаптивность: 85

Ключевые выводы:

Исследование предлагает метод Iterative Summarization Pre-prompting (ISP2) для улучшения способностей больших языковых моделей (LLM) к рассуждению. Основная цель - повысить эффективность Chain of Thought (CoT) путем предварительной обработки информации перед рассуждением. Результаты показывают улучшение производительности на 7.1% по сравнению с существующими методами.

Объяснение метода:

Исследование предлагает метод "понимание перед рассуждением", который легко адаптировать для повседневного использования в чатах с LLM. Пользователи могут применять принцип поэтапной обработки информации, сначала структурируя данные, затем рассуждая. Метод показывает значительное улучшение точности на разных моделях и задачах, особенно когда ключевая информация неявна.

Ключевые аспекты исследования: 1. **Метод Iterative Summarization Pre-prompting (ISP2)** - авторы предлагают метод предварительного промптинга, который улучшает способность LLM к рассуждению, особенно когда ключевая информация не представлена явно. ISP2 действует до применения Chain-of-Thought (CoT), помогая модели сначала понять и структурировать информацию, прежде чем начать рассуждение.

Трехэтапный процесс обработки информации - метод включает: (а) адаптивное извлечение кандидатной информации, (б) оценку надежности информационных пар, (в) итеративное обобщение для понимания знаний. Это позволяет постепенно уточнять информацию и формировать более полное понимание задачи.

Фокус на понимании проблемы перед рассуждением - в отличие от стандартных методов CoT, которые сразу переходят к цепочке рассуждений, ISP2 сначала фокусируется на извлечении и структурировании информации, что помогает модели лучше понять суть проблемы.

Плагин для существующих методов рассуждения - ISP2 разработан как дополнение к существующим методам CoT, которое можно легко интегрировать в

различные подходы к рассуждению, улучшая их эффективность без изменения базовой архитектуры.

Значительное улучшение производительности - на тестовых наборах данных ISP2 показал улучшение точности на 7.1% для GPT-3.5, 8.1% для LLaMA2-13B и 12.4% для LLaMA2-7B, особенно в задачах, требующих сложных рассуждений.

Дополнение:

Применимость метода в стандартном чате без дообучения

Методы исследования ISP2 **не требуют дообучения или специального API** для их применения. Хотя авторы использовали различные модели для тестирования (GPT-3.5, LLaMA2), сам подход основан исключительно на структурировании промптов и может быть применен в любом стандартном чате с LLM.

Ключевые концепции для адаптации в стандартном чате:

Двухэтапный промптинг - Пользователи могут разбить взаимодействие на два шага: Шаг 1: "Пожалуйста, проанализируй этот вопрос и выдели ключевую информацию, организовав её в информационные пары сущность-описание" Шаг 2: "Теперь, используя эту структурированную информацию, ответь на исходный вопрос, рассуждая шаг за шагом"

Итеративное обобщение - Можно попросить модель объединять и обобщать информацию:

"Пожалуйста, объедини эти две информационные пары в более полное описание проблемы" "Определи, какая информация кажется неполной или противоречивой, и уточни её"

Оценка надежности информации - Пользователи могут запросить оценку извлеченной информации:

"Оцени надежность каждой части информации по шкале от 1 до 10" "Какие аспекты задачи требуют дополнительного уточнения?" ### Ожидаемые результаты:

При применении этих концепций в стандартном чате пользователи могут ожидать: - Более точные ответы на сложные вопросы с неявной информацией - Лучшую структуризацию мышления модели - Снижение ошибок, вызванных пропуском ключевой информации - Более прозрачное рассуждение, позволяющее отследить ход мыслей модели

Хотя полная реализация трехэтапного процесса ISP2 может быть громоздкой для повседневного использования, даже частичное применение основных принципов может значительно улучшить результаты, особенно в задачах, требующих сложного рассуждения.

Prompt:

Использование ISP2 в промптах для GPT ## Ключевые аспекты ISP2 Метод Iterative Summarization Pre-prompting (ISP2) улучшает способность языковых моделей к рассуждению через: 1. **Адаптивное извлечение** информационных пар из вопроса 2. **Оценку надежности** этих пар 3. **Итеративное объединение** пар с низким рейтингом

Пример промпта с применением ISP2

[======] Я хочу, чтобы ты решил следующую задачу, используя метод итеративного суммирования перед рассуждением:

Задача: В городе N проживает 150,000 человек. Ежегодно население увеличивается на 3%, а также прибывает около 2,000 новых жителей из других регионов. Сколько человек будет проживать в городе через 5 лет?

Сначала выдели ключевые информационные пары из задачи (сущности и их описания): [Начальное население]: 150,000 человек [Ежегодная миграция]: +2,000 человек [Временной период]: 5 лет [Искомая величина]: население через 5 лет

Оцени надежность каждой пары и определи, достаточно ли информации для решения.

Объедини информационные пары в краткое обобщение задачи: "Задача на расчет будущего населения города, начиная со 150,000 человек, с учетом ежегодного прироста 3% и дополнительной миграции 2,000 человек в год на протяжении 5 лет."

Теперь, используя это обобщение, построй цепочку рассуждений для решения задачи. [======]

Как это работает

Данный промпт реализует принципы ISP2:

Структурированное извлечение информации: Мы явно просим модель выделить ключевые пары "сущность-описание", что помогает ей не упустить важные детали.

Проверка достаточности данных: Этап оценки надежности помогает выявить возможные пробелы в информации до начала решения.

Итеративное обобщение: Объединение информационных пар в краткое резюме задачи позволяет модели лучше понять общую структуру проблемы.

Разделение понимания и рассуждения: Сначала модель фокусируется на понимании задачи, и только затем переходит к построению цепочки рассуждений.

Такой подход особенно эффективен для математических задач и задач, требующих

здравого смысла, так как помогает модели сначала полностью понять контекст, а затем применить логическое рассуждение.

№ 44. Большие языковые модели для локализации уязвимостей в файле могут оказаться «потерянными в конце»

Ссылка: https://arxiv.org/pdf/2502.06898

Рейтинг: 78
Адаптивность: 85

Ключевые выводы:

Исследование оценивает эффективность современных LLM (GPT-3.5, GPT-4, Mixtral, Llama) в обнаружении уязвимостей в файлах кода. Основной вывод: LLM значительно хуже обнаруживают уязвимости, расположенные ближе к концу больших файлов (эффект «lost in the end»), что противоречит ранее известному эффекту «lost in the middle».

Объяснение метода:

Исследование выявляет "lost in the end" эффект в LLM и предлагает простую стратегию "chunking" для повышения эффективности обнаружения уязвимостей на 37%. Предоставляет конкретные рекомендации по оптимальным размерам фрагментов для анализа кода (500-6500 символов), которые любой пользователь может немедленно применить без специальных инструментов. Ограничения: исследованы только три типа уязвимостей и ограниченный набор моделей.

Ключевые аспекты исследования: 1. **"Lost in the End" эффект** - исследование выявило, что современные LLM (GPT-4, Llama 3, Mixtral) имеют тенденцию пропускать уязвимости, расположенные в конце длинных файлов, что авторы назвали эффектом "lost in the end".

Влияние размера файла и позиции уязвимости - обнаружена отрицательная корреляция между размером файла/позицией уязвимости и вероятностью её обнаружения LLM. Чем больше файл или чем дальше к концу файла расположена уязвимость, тем ниже вероятность её обнаружения.

Стратегия "chunking" - разделение больших файлов на меньшие фрагменты повышает эффективность обнаружения уязвимостей LLM-моделями (в среднем на 37% по всем моделям).

Оптимальные размеры входных данных - исследование определило оптимальные размеры фрагментов для разных типов уязвимостей (CWE-22: 3000-6500 символов, CWE-89 и CWE-79: 500-1500 символов).

Сравнение коммерческих и открытых моделей - при уменьшении размера

входных данных разница в производительности между коммерческими и открытыми моделями сокращается, что указывает на то, что преимущество коммерческих моделей может заключаться в лучшей обработке контекстных окон.

Дополнение: Действительно ли для работы методов этого исследование требуется дообучение или API? Или методы и подходы можно применить в стандартном чате, а ученые лишь для своего удобства использовали расширенные техники?

Методы и подходы из этого исследования **не требуют дообучения или специального API** и могут быть применены в стандартном чате с LLM. Исследователи использовали модели "off-the-shelf" (без дополнительной настройки), как указано в разделе 5: "мы использовали эти модели в их конфигурациях по умолчанию без какой-либо тонкой настройки".

Основные концепции, которые можно применить в стандартном чате:

Стратегия "chunking" - разделение больших файлов кода на меньшие фрагменты перед отправкой в чат. Это самый важный и простой в реализации метод, который значительно повышает эффективность обнаружения проблем в коде.

Оптимальные размеры фрагментов - использование рекомендованных размеров: 500-1500 символов для большинства проблем безопасности и до 6500 символов для более сложных случаев.

Приоритизация начала файла - понимание того, что LLM хуже обрабатывают конец длинных файлов, и соответственно структурирование запросов.

Структурированный промпт - использование формата промпта с четкой структурой ожидаемого ответа, как в исследовании: "Se: [объяснение], Bl: [проблемная строка], BUG FOUND: YES/NO".

Результаты от применения этих концепций: - Значительное повышение эффективности обнаружения проблем в коде (до 95% для некоторых типов проблем) - Более точное указание проблемных мест в коде - Снижение влияния размера файла и позиции проблемы на эффективность анализа - Более структурированные и полезные ответы от LLM

Эти подходы применимы не только к поиску уязвимостей, но и к другим задачам анализа кода: поиску багов, code review, оптимизации и рефакторингу.

Prompt:

Применение исследования для оптимизации промптов при поиске уязвимостей ## Ключевые инсайты из исследования

Исследование показывает, что большие языковые модели (LLM) значительно хуже

обнаруживают уязвимости в конце больших файлов кода ("эффект lost in the end"). Разделение файлов на меньшие фрагменты существенно улучшает результаты.

Пример оптимизированного промпта

[======] # Задача: Проверка кода на уязвимости XSS (CWE-79)

Контекст Я разрабатываю стратегию анализа безопасности кода. Исследования показывают, что LLM часто пропускают уязвимости в конце файлов, поэтому я разделил код на фрагменты размером ~500 символов.

Инструкции 1. Проанализируй следующий фрагмент кода на наличие XSS-уязвимостей (CWE-79) 2. Обрати ОСОБОЕ внимание на код в конце фрагмента 3. Рассмотри, как пользовательский ввод обрабатывается и выводится 4. Проверь все переменные в контексте данного фрагмента 5. Если найдешь уязвимость, опиши её, почему она возникает и как её исправить

Фрагмент кода для анализа [======]javascript // Код фрагмента здесь [======]

Дополнительный контекст Этот фрагмент является частью [описание функциональности]. Переменные [X, Y, Z] получают данные от пользователя. [======]

Почему этот промпт работает эффективнее

Оптимальный размер фрагмента: Промпт учитывает рекомендации исследования по оптимальному размеру фрагментов (500 символов для XSS-уязвимостей)

Акцент на конец фрагмента: Явно обращает внимание модели на конец фрагмента, где уязвимости чаще пропускаются

Сохранение контекста: Включает информацию о переменных и их происхождении, что важно для определения уязвимостей, связанных с пользовательским вводом

Специфика типа уязвимости: Промпт сфокусирован на конкретном типе уязвимости (XSS/CWE-79), что улучшает точность анализа

Для других типов уязвимостей размер фрагментов нужно корректировать: 500-1500 символов для SQL-инъекций (CWE-89) и 3000-6500 символов для уязвимостей обхода пути (CWE-22).

№ 48. PReasoning о теории разума на основе гипотез для больших языковых моделей

Ссылка: https://arxiv.org/pdf/2502.11881

Рейтинг: 78
Адаптивность: 85

Ключевые выводы:

Исследование представляет алгоритм Thought Tracing для улучшения способности больших языковых моделей (LLM) отслеживать и выводить ментальные состояния агентов в тексте. Основная цель - разработать метод, который может отслеживать мысли и убеждения персонажей в тексте без опоры на заранее известные ответы. Результаты показывают, что этот алгоритм значительно улучшает производительность LLM на задачах теории сознания (Theory of Mind), превосходя базовые модели и специализированные модели рассуждения.

Объяснение метода:

Исследование представляет высокую ценность, предлагая метод улучшения взаимодействия с LLM в задачах понимания намерений. Алгоритм Thought Tracing дает практический подход к структурированию запросов, демонстрирует способы преодоления ограничений моделей и работы с неопределенностью. Основные концепции доступны для адаптации, хотя полная реализация требует технических знаний.

Ключевые аспекты исследования: 1. **Алгоритм Thought Tracing** - новый метод для отслеживания и вывода ментальных состояний агентов в тексте, основанный на принципах байесовской теории разума (BToM) и алгоритме Sequential Monte Carlo. Алгоритм генерирует множество гипотез о мыслях агента и взвешивает их на основе наблюдений.

Естественно-языковое представление гипотез - в отличие от традиционных вероятностных моделей, гипотезы о ментальных состояниях представлены в виде естественного языка и генерируются языковыми моделями.

Улучшение производительности моделей в задачах Theory of Mind - алгоритм значительно улучшает способность языковых моделей отвечать на вопросы, связанные с пониманием намерений и ментальных состояний агентов, без специального дообучения.

Сравнение с моделями рассуждения - исследование выявило, что модели, специализирующиеся на рассуждениях (O1, R1), не демонстрируют такого же превосходства в задачах Theory of Mind, как в математических задачах.

Эффективность в условиях неопределенности - алгоритм специально разработан для работы в социальной сфере, где отсутствуют объективно проверяемые ответы, в отличие от математических или программистских задач.

Дополнение: Исследование представляет алгоритм "Thought Tracing", который действительно можно адаптировать для использования в стандартном чате с LLM без необходимости дообучения или специальных API.

Хотя авторы использовали API для подсчета весов гипотез, они отмечают, что вместо этого можно использовать простой подход с инструкциями для модели выбрать из шести вариантов вероятности (от "очень вероятно" до "очень маловероятно"), что работает даже лучше.

Основные концепции, которые можно применить в стандартном чате:

Разделение текста на состояния и действия - можно попросить модель проанализировать текст, выделив состояния и действия агента.

Генерация гипотез о ментальных состояниях - можно попросить модель сгенерировать несколько (3-4) гипотез о том, что агент мог думать в определенный момент.

Оценка вероятности гипотез - можно попросить модель оценить, насколько вероятно каждое действие агента, учитывая каждую гипотезу.

Обновление гипотез - на основе новых действий можно попросить модель обновить гипотезы.

Суммирование гипотез - в конце можно попросить модель обобщить наиболее вероятные мысли агента.

Пример применения в стандартном чате:

Пользователь: Проанализируй этот текст: "Джон искал по всему дому ключи. Он проверил кухню, гостиную и спальню, но не заглянул в ванную. Затем он вышел из дома."

Выдели состояния и действия Джона. Предложи 3 гипотезы о том, что Джон мог думать после проверки спальни. Оцени, насколько вероятно его действие "выйти из дома" при каждой гипотезе. Какое наиболее вероятное ментальное состояние Джона когда он выходил из дома? Такой подход может значительно улучшить понимание намерений персонажей в текстах, анализ литературных произведений, и даже помочь в интерпретации реальных ситуаций, новостей или поведения людей. Это особенно ценно для писателей, психологов, аналитиков и всех, кто работает с анализом поведения и намерений.

Prompt:

Использование Thought Tracing в промтах для GPT ## Суть метода Thought Tracing

Метод Thought Tracing позволяет языковым моделям лучше отслеживать и выводить ментальные состояния персонажей в тексте. Он работает путем: - Генерации множественных гипотез о мыслях персонажей - Взвешивания этих гипотез на основе наблюдаемых действий - Последовательного обновления представлений о ментальных состояниях

Пример промта для анализа литературного произведения

[======] Проанализируй следующий отрывок из романа, используя метод Thought Tracing:

[ТЕКСТ ОТРЫВКА]

Инструкции: 1. Разбей текст на последовательность состояний и действий каждого ключевого персонажа. 2. Для каждого персонажа сгенерируй 3-4 возможные гипотезы о его текущих мыслях, убеждениях и намерениях в каждой ключевой точке повествования. 3. Оцени вероятность каждой гипотезы, основываясь на наблюдаемых действиях персонажа. 4. Для наиболее вероятных гипотез опиши, как они объясняют последующие действия персонажа. 5. В заключении, представь наиболее правдоподобную траекторию мыслей каждого персонажа на протяжении всего отрывка.

Важно: Фокусируйся не только на том, что персонажи знают, но и на том, во что они верят, чего не знают, и как их неполное или ошибочное понимание ситуации влияет на их действия. [======]

Почему это работает

Данный промт использует ключевые аспекты Thought Tracing:

Генерация множественных гипотез - просим модель создать несколько возможных объяснений ментальных состояний **Оценка вероятности** - заставляем модель взвешивать гипотезы на основе действий персонажей **Последовательное обновление** - требуем отслеживать изменения в ментальных состояниях с течением повествования Такой подход позволяет GPT выйти за рамки поверхностного анализа и глубже проникнуть в теорию сознания персонажей, что, согласно исследованию, значительно улучшает качество анализа социальных взаимодействий и понимание мотиваций персонажей.

№ 52. Модульное тестирование: прошлое и настоящее. Исследование влияния LLM на обнаружение дефектов и эффективность

Ссылка: https://arxiv.org/pdf/2502.09801

Рейтинг: 78
Адаптивность: 85

Ключевые выводы:

Основная цель исследования - изучить влияние больших языковых моделей (LLM) на эффективность обнаружения дефектов при модульном тестировании. Исследование показало, что использование LLM значительно повышает продуктивность тестирования: участники создали больше тестов, достигли более высокого покрытия кода и обнаружили больше дефектов по сравнению с ручным тестированием.

Объяснение метода:

Исследование демонстрирует высокую практическую ценность, предоставляя количественные доказательства преимуществ LLM в юнит-тестировании. Результаты показывают значительное повышение продуктивности (+119% тестов, больше обнаруженных дефектов) и применимы напрямую разработчиками. Выявление компромисса между количеством и качеством дает важное понимание ограничений.

Ключевые аспекты исследования: 1. **Сравнение эффективности юнит-тестирования с LLM и без них**: Исследование сравнивает результаты тестирования, выполненного с поддержкой LLM и вручную, на основе одинакового набора задач и временных ограничений.

Количественные метрики эффективности: Авторы измеряли количество созданных тестов, покрытие кода, количество обнаруженных дефектов и число ложноположительных срабатываний.

Значительное повышение продуктивности: Участники с поддержкой LLM создали в среднем на 119% больше тестов (59.3 против 27.1) и обнаружили больше дефектов (6.5 против 3.7) по сравнению с ручным тестированием.

Соотношение качества и количества: Исследование выявило корреляцию между количеством созданных тестов и числом ложноположительных результатов, что указывает на компромисс между объемом и точностью тестирования.

Практическое применение LLM в процессе тестирования: Участники

использовали различные LLM-инструменты (ChatGPT, GitHub Copilot) в интерактивном режиме для поддержки юнит-тестирования.

Дополнение: Исследование не требует дообучения или специального API для применения его методов и подходов. Авторы сравнивали стандартное использование общедоступных LLM (ChatGPT, GitHub Copilot) с ручным написанием юнит-тестов. Все участники работали в стандартном интерфейсе этих инструментов без дополнительной настройки.

Концепции и подходы, применимые в стандартном чате:

Интерактивное использование LLM в процессе тестирования: Пользователи могут запрашивать у LLM помощь в создании тестов, анализе покрытия и поиске потенциальных дефектов, не полагаясь полностью на автоматическую генерацию.

Стратегия балансирования количества и качества: Зная о корреляции между объемом тестов и ложноположительными результатами, пользователи могут формулировать запросы, акцентирующие внимание на качестве, а не только количестве.

Повышение эффективности тестирования: Применение LLM для быстрого создания базовых тестов, которые затем могут быть доработаны вручную, что значительно ускоряет процесс разработки.

Ожидаемые результаты от применения этих концепций: - Увеличение количества созданных тестов в 2-2.5 раза - Повышение покрытия кода на 10% и более - Обнаружение большего количества дефектов (примерно на 75% больше) - Более эффективное использование ограниченного времени разработки - Возможное увеличение числа ложноположительных результатов, требующих дополнительной проверки

Важно отметить, что эти подходы не требуют специальных инструментов или API, а могут быть применены при обычном взаимодействии с LLM через стандартный интерфейс чата.

Prompt:

Использование результатов исследования по модульному тестированию в промптах для GPT ## Ключевые знания из исследования для применения в промптах

Исследование демонстрирует, что LLM значительно повышают эффективность модульного тестирования: - Увеличение количества создаваемых тестов на 119% - Повышение обнаружения дефектов на 76% - Улучшение покрытия кода на 10 процентных пунктов - Возможное увеличение ложноположительных результатов

Пример промпта для GPT

[======] Действуй как опытный инженер по тестированию, специализирующийся на модульном тестировании Java-кода с использованием JUnit.

Мне нужно создать модульные тесты для следующего класса:

[======]java [ВСТАВИТЬ КОД КЛАССА] [======]

Основываясь на результатах исследования о влиянии LLM на эффективность модульного тестирования, я прошу:

Создать максимально полный набор тестов для покрытия не менее 75% ветвей кода Особое внимание уделить краевым случаям и потенциальным дефектам Для каждого теста: Написать ясный комментарий, объясняющий цель теста Указать, какую часть кода он покрывает Описать потенциальные дефекты, которые он может обнаружить

Предложить стратегию по минимизации ложноположительных результатов

Пожалуйста, структурируй тесты по функциональным блокам и отметь приоритетные тесты, которые с наибольшей вероятностью выявят дефекты. [======]

Как работают знания из исследования в этом промпте

Ориентация на высокое покрытие кода (75% и выше) основана на данных исследования о том, что с LLM можно достичь покрытия в 74% против 67% при ручном тестировании.

Акцент на обнаружение дефектов соответствует выводам о том, что с LLM можно обнаружить значительно больше дефектов (в среднем 6,5 против 3,7).

Запрос большого количества тестов опирается на факт, что с LLM можно создать в 2 раза больше тестов за то же время.

Внимание к ложноположительным результатам учитывает обнаруженную проблему увеличения ложноположительных тестов при использовании LLM (5,1 против 2,7).

Структурирование и приоритизация тестов помогает эффективнее использовать повышенную производительность, которую дает LLM.

Такой промпт позволяет максимально использовать преимущества LLM в модульном тестировании, выявленные в исследовании, одновременно учитывая потенциальные недостатки.

№ 56. VisPath: Автоматизированный синтез кода визуализации с помощью многопутевого рассуждения и оптимизации на основе обратной связи

Ссылка: https://arxiv.org/pdf/2502.11140

Рейтинг: 78
Адаптивность: 85

Ключевые выводы:

Исследование представляет VisPath - новую систему для автоматического создания кода визуализации данных с использованием LLM. Основная цель - преодолеть ограничения существующих методов путем применения мультипутевого рассуждения и оптимизации на основе обратной связи. Результаты показывают, что VisPath значительно превосходит современные методы, повышая точность и надежность генерации кода визуализации в среднем на 17%.

Объяснение метода:

VisPath предлагает ценную методологию мульти-путевого рассуждения и итеративного улучшения визуализаций, которая может быть адаптирована для широкого спектра взаимодействий с LLM. Пользователи могут применять принципы генерации нескольких вариантов решения, их оценки и синтеза оптимального результата для улучшения качества визуализаций и других задач.

Ключевые аспекты исследования: 1. **Мульти-путевое рассуждение (Multi-Path Reasoning)** - VisPath генерирует несколько вариантов интерпретации запроса пользователя, создавая различные пути рассуждения для более полного понимания намерений пользователя, особенно при неоднозначных запросах.

Генерация кода из путей рассуждения - На основе каждого пути рассуждения система генерирует отдельный код визуализации с использованием цепочки мышления (Chain of Thought), что позволяет создать несколько вариантов визуализаций.

Оптимизация кода на основе обратной связи - Система использует модели зрительно-языкового интеллекта (Vision-Language Models) для оценки качества каждой визуализации и предоставления структурированной обратной связи.

Синтез оптимального результата - На заключительном этапе VisPath объединяет лучшие элементы из всех сгенерированных кодов и обратной связи, создавая оптимальный финальный код визуализации.

Итеративное улучшение визуализации - Фреймворк способен адаптироваться к неоднозначным запросам, улучшая выполняемость кода и визуальное качество результата.

Дополнение: Для работы методов этого исследования в полном объёме действительно требуется API для доступа к нескольким моделям (LLM и VLM), однако основные концепции и подходы могут быть успешно адаптированы для использования в стандартном чате с LLM. Вот ключевые концепции, которые можно применить:

Мульти-путевое рассуждение: Пользователь может попросить LLM сгенерировать несколько разных интерпретаций своего запроса и разработать отдельные подходы для каждой интерпретации. Например: "Предложи 3 разных способа визуализации данных о продажах, фокусируясь на разных аспектах: временные тренды, сравнение категорий, географическое распределение".

Структурированное рассуждение через Chain of Thought: Пользователь может попросить LLM объяснить свой ход мыслей при создании кода визуализации: "Объясни пошагово, как ты решаешь задачу визуализации этих данных, и какие решения принимаешь на каждом этапе".

Самооценка и обратная связь: Пользователь может попросить LLM оценить собственный сгенерированный код: "Проанализируй этот код визуализации и укажи его сильные и слабые стороны, а также предложи улучшения".

Синтез оптимального решения: После получения нескольких вариантов кода, пользователь может попросить LLM объединить лучшие элементы: "На основе этих трёх вариантов кода визуализации, создай оптимальный вариант, который объединяет лучшие практики из каждого".

Применение этих концепций в стандартном чате позволит получить следующие результаты: - Более точное понимание LLM намерений пользователя - Разнообразные варианты решения одной задачи - Более качественный и надёжный код визуализации - Лучшее понимание пользователем возможностей и ограничений визуализации данных

Хотя полная автоматизация процесса (как в исследовании) требует API, базовые принципы VisPath могут значительно повысить качество взаимодействия с LLM даже в стандартном чате.

Prompt:

Использование знаний из исследования VisPath в промптах для GPT ## Ключевые концепции для применения в промптах

Исследование VisPath предлагает несколько ценных подходов, которые можно адаптировать для создания более эффективных промптов:

Многопутевое рассуждение - генерация нескольких интерпретаций запроса **Chain of Thought (CoT)** - пошаговое рассуждение для генерации кода **Оптимизация на основе обратной связи** - итеративное улучшение результатов ## Пример промпта с применением принципов VisPath

[======] # Запрос на визуализацию данных с использованием многопутевого подхода

Описание данных [Описание датасета: структура, переменные, типы данных]

Желаемая визуализация [Описание того, что нужно визуализировать]

Инструкции: 1. Сгенерируй 3 различные интерпретации моего запроса на визуализацию, учитывая возможную неоднозначность. 2. Для каждой интерпретации: - Объясни свой ход мыслей (Chain of Thought) - Предложи подходящий тип визуализации - Опиши, какие инсайты можно получить из этой визуализации 3. Создай Python-код для каждой из трех интерпретаций, используя библиотеку matplotlib/seaborn. 4. Проанализируй потенциальные недостатки каждой визуализации и предложи улучшения. 5. Выбери наиболее информативную визуализацию из трех и объясни свой выбор.

Важно: Убедись, что код включает правильно оформленные легенды, метки осей и подходящие стили линий/цветов. [======]

Как это работает

Данный промпт адаптирует ключевые принципы VisPath:

Многопутевое рассуждение - запрашивая 3 разные интерпретации, мы получаем оптимальное количество путей рассуждения (согласно исследованию VisPath), что обеспечивает разнообразие без избыточного шума.

Chain of Thought (CoT) - требуя объяснения хода мыслей, мы побуждаем модель к более глубокому анализу, что повышает точность генерации кода.

Оптимизация на основе обратной связи - хотя мы не можем напрямую использовать VLM для оценки, мы имитируем этот процесс, прося модель проанализировать недостатки и предложить улучшения для каждой визуализации.

Такой подход позволяет получить более качественные и надежные визуализации, особенно при работе с неоднозначными запросами, что соответствует основным преимуществам VisPath, отмеченным в исследовании.

№ 60. О надежности генеративных базовых моделей: руководство, оценка и перспектива

Ссылка: https://arxiv.org/pdf/2502.14296

Рейтинг: 78
Адаптивность: 85

Ключевые выводы:

Исследование направлено на создание комплексной системы оценки надежности генеративных моделей искусственного интеллекта (GenFMs) через разработку стандартизированных руководящих принципов и динамической системы оценки TrustGen. Основные результаты показывают, что современные GenFMs демонстрируют высокий уровень надежности, но сохраняют уязвимости в различных аспектах, таких как безопасность, конфиденциальность и этика. Открытые модели значительно сократили разрыв в надежности с проприетарными моделями.

Объяснение метода:

Исследование предлагает комплексную основу для оценки надежности генеративных моделей с гибкими руководствами и динамической системой TrustGen. Высокая ценность для разработчиков и продвинутых пользователей, предоставляет как теоретическую базу, так и практические инструменты с открытым кодом. Требует определенной технической подготовки, но многие принципы могут быть адаптированы и упрощены для широкой аудитории.

Ключевые аспекты исследования: 1. **Комплексная концептуальная основа для оценки доверия к генеративным моделям**: Исследование представляет структурированные руководства по обеспечению надежности генеративных моделей (GenFMs), включающие ключевые аспекты: правдивость, безопасность, справедливость, устойчивость, конфиденциальность и этику.

Динамическая система оценки TrustGen: Разработана первая динамическая платформа для оценки надежности различных типов генеративных моделей (текст-в-изображение, языковые модели, мультимодальные модели). В отличие от статических тестов, TrustGen постоянно адаптируется к новым моделям и угрозам.

Модульная архитектура оценки: TrustGen включает три основных компонента: куратор метаданных, конструктор тестовых примеров и контекстуальный вариатор, что обеспечивает гибкость и постоянное обновление тестов.

Всесторонняя оценка существующих моделей: Исследование предоставляет детальный анализ надежности ведущих генеративных моделей по различным параметрам, выявляя их сильные и слабые стороны.

Стратегическое видение будущих направлений: Работа обсуждает ключевые проблемы и перспективы в области надежности генеративных моделей, предоставляя стратегическую дорожную карту для будущих исследований.

Дополнение: Исследование не требует дообучения или API для применения его основных методов и подходов. Хотя авторы используют продвинутые технические средства для своей работы, концепции и методология могут быть адаптированы для использования в стандартном чате.

Ключевые концепции, которые можно применить в стандартном чате:

Структурированная оценка доверия к моделям: Пользователи могут применять предложенные измерения (правдивость, безопасность, справедливость и т.д.) для систематической оценки ответов моделей.

Контекстуальная вариация запросов: Можно задавать один и тот же вопрос в различных формулировках для проверки устойчивости ответов модели.

Тестирование на предвзятость и справедливость: Пользователи могут проверять, насколько ответы модели варьируются при изменении демографических атрибутов в запросе.

Проверка на склонность к "сикофантству": Можно сформулировать запрос таким образом, чтобы проверить, будет ли модель необоснованно соглашаться с утверждениями пользователя.

Оценка честности модели: Можно проверять, признает ли модель границы своего знания или склонна генерировать правдоподобную, но неверную информацию.

Многоуровневая проверка безопасности: Можно тестировать отказоустойчивость модели к запросам о потенциально вредной информации.

Сравнительный анализ различных моделей: Пользователи могут сравнивать ответы разных доступных моделей на одинаковые запросы.

Результатом применения этих концепций будет более осознанное и критическое использование LLM, лучшее понимание их ограничений и возможностей, а также способность формулировать запросы, которые с большей вероятностью приведут к надежным и полезным ответам.

Prompt:

Применение исследования надежности GenFMs в промптах для GPT ## Ключевые аспекты исследования для использования в промптах

Исследование "О надежности генеративных базовых моделей" предоставляет

ценные знания о сильных и слабых сторонах современных генеративных моделей. Эти знания можно использовать для создания более эффективных промптов, учитывающих:

Семь измерений надежности моделей Уязвимые места в правдивости, безопасности и конфиденциальности Динамический подход к тестированию вместо статического Необходимость контекстной адаптации надежности ## Пример промпта с применением знаний из исследования

[======] Действуй как эксперт по медицинской информации. Мне нужна информация о методах лечения диабета 2 типа.

При ответе: 1. Четко разделяй научно доказанные методы и экспериментальные подходы (учитывая измерение правдивости) 2. Укажи степень уверенности в каждом утверждении (применяя калибровку доверия) 3. Предоставь информацию в контексте разных профилей пациентов (используя контекстный вариатор) 4. Не рекомендуй конкретные дозировки лекарств без медицинской консультации (соблюдая безопасность) 5. Учитывай этические аспекты доступности лечения (измерение справедливости)

В конце ответа предложи 3 вопроса для уточнения, которые помогут персонализировать информацию под мои конкретные потребности. [======]

Объяснение эффективности промпта

Данный промпт применяет знания из исследования следующим образом:

Учитывает многомерность надежности - явно запрашивает соблюдение нескольких измерений надежности (правдивость, безопасность, справедливость) **Внедряет механизмы калибровки доверия** - требует указания степени уверенности в утверждениях **Использует контекстную вариативность** - запрашивает адаптацию информации для разных профилей пользователей **Устанавливает этические границы** - предотвращает потенциально опасные рекомендации **Создает динамическую обратную связь** - через запрос дополнительных вопросов, что имитирует динамическую систему оценки из исследования Такой подход позволяет получить более надежный, контекстуально-релевантный и безопасный ответ от модели, используя принципы, выявленные в исследовании.

№ 64. Многоэтапное, цепочное редактирование пост-текстов для неверных резюме

Ссылка: https://arxiv.org/pdf/2501.11273

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование направлено на улучшение фактической точности автоматически сгенерированных текстовых резюме с помощью LLM. Основная цель - разработка метода многоэтапного редактирования резюме с использованием цепочки рассуждений (Chain of Thought) для выявления и исправления фактических несоответствий. Результаты показывают, что предложенный подход значительно превосходит существующие методы редактирования, достигая более высокого процента успешных исправлений.

Объяснение метода:

Исследование предлагает практичную методологию многоэтапного редактирования текстов с использованием Chain-of-Thought промптов для выявления и исправления фактологических ошибок. Подход не требует технических знаний, применим в стандартных чатах с LLM и демонстрирует значительное улучшение точности текстов. Ценность для пользователей в готовых промптах и пошаговой методике, которые можно применять для улучшения автоматически генерируемого контента.

Ключевые аспекты исследования: 1. **Многоэтапное редактирование саммари**: Исследование предлагает фреймворк, где LLM выступает одновременно в роли критика (оценивает фактологическую точность) и редактора (исправляет неточности) саммари. Ключевая инновация - многораундовое редактирование до достижения фактической точности.

Chain-of-Thought (CoT) промпты: Авторы разработали специальные промпты для LLM, где модель сначала определяет проблемные места и типы ошибок в саммари, а затем на основе этого анализа редактирует текст.

Типология фактологических ошибок: Исследование использует детальную классификацию типов ошибок (ошибки в предикатах, сущностях, обстоятельствах и т.д.), что позволяет более точно выявлять и исправлять неточности.

Сравнение различных CoT-стратегий: Авторы сравнивают эффективность разных подходов к редактированию - с определением проблемного фрагмента, типа ошибки или обоих параметров одновременно.

Метрики оценки фактологической точности: В работе представлена методология

оценки точности саммари с использованием LLM в качестве оценщика, показывающая высокую корреляцию с человеческими оценками.

Дополнение: Исследование "Multi-round, Chain-of-thought Post-editing for Unfaithful Summaries" особенно ценно тем, что его методы **не требуют дополнительного дообучения или специальных API** для применения в обычном чате с LLM.

Ключевые концепции, применимые в стандартном чате:

Итеративный подход к проверке и редактированию - пользователь может запрашивать LLM сначала оценить текст, затем отредактировать его, а после повторно оценить. Эти шаги можно повторять до достижения удовлетворительного результата.

Chain-of-Thought промпты для редактирования - авторы обнаружили, что просьба к модели сначала выявить конкретные проблемные фрагменты и типы ошибок значительно улучшает качество редактирования. Эти промпты можно напрямую использовать в обычном чате.

Типология ошибок - классификация различных типов фактологических ошибок (в предикатах, сущностях, обстоятельствах и т.д.) помогает пользователям более точно формулировать запросы на редактирование.

Ожидаемые результаты от применения этих методов: - Значительное улучшение фактологической точности текстов (авторы показали улучшение на ~50% между первым и последним раундами редактирования) - Более высокое качество редактирования по сравнению с одноразовой правкой - Лучшая сохранность смысла исходного текста при исправлении ошибок

Хотя исследователи использовали различные модели для экспериментов, сами методы не зависят от конкретной модели и могут быть применены с любой современной LLM в обычном чате без технических модификаций.

Prompt:

Использование исследования о многоэтапном редактировании в промптах для GPT Исследование о многоэтапном редактировании текстов содержит ценные находки, которые можно эффективно применить при составлении промптов для GPT. Вот как это можно сделать:

Ключевые принципы для использования в промптах:

Многоэтапное редактирование вместо одноразовой правки **Цепочка рассуждений (Chain of Thought)** для выявления и исправления ошибок **Разделение ролей** на "критика" и "редактора" **Фокус на типичных ошибках** (предикаты, сущности, отсутствующая информация) ## Пример промпта для улучшения фактической точности текста:

[======] Я хочу, чтобы ты помог мне проверить и улучшить фактическую точность следующего резюме статьи. Действуй поэтапно:

РОЛЬ КРИТИКА: Оцени фактическую точность резюме по шкале от 1 до 5 Используй цепочку рассуждений (CoT) для выявления всех фактических ошибок Классифицируй каждую ошибку по типу: предикаты (действия), сущности (объекты), отсутствующая в оригинале информация

РОЛЬ РЕДАКТОРА:

Исправь выявленные ошибки, сохраняя стиль и структуру текста Объясни внесенные изменения

ИТЕРАТИВНАЯ ПРОВЕРКА:

Проведи повторную оценку исправленного текста При необходимости выполни дополнительные циклы редактирования (до 3 раз) После каждого цикла указывай оставшиеся проблемы Исходный документ: [ВСТАВИТЬ ОРИГИНАЛЬНЫЙ ДОКУМЕНТ]

Резюме для проверки и редактирования: [ВСТАВИТЬ РЕЗЮМЕ] [======]

Почему это работает:

Данный промпт применяет ключевые находки исследования:

Разделение на роли критика и редактора - исследование показало, что такое разделение повышает эффективность обнаружения и исправления ошибок

Многоэтапный подход - согласно исследованию, около 50% улучшений происходит между первым и последним раундами редактирования

Цепочка рассуждений (CoT) - исследование доказало, что промпты с CoT значительно улучшают результаты редактирования

Классификация типов ошибок - особое внимание к ошибкам в предикатах, сущностях и добавленной информации, которые исследование выявило как критические

Такой структурированный подход позволяет GPT более методично выявлять и исправлять фактические ошибки, что приводит к значительно более точным резюме, чем при использовании простых одноэтапных промптов.

№ 68. Парсинг логов с использованием LLM с самосгенерированным обучением в контексте и самокоррекцией

Ссылка: https://arxiv.org/pdf/2406.03376

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование представляет AdaParser - эффективный и адаптивный фреймворк для парсинга логов с использованием больших языковых моделей (LLM). Основная цель - преодолеть ограничения существующих парсеров логов, которые плохо справляются с эволюционирующими логами. AdaParser превосходит современные методы по всем метрикам точности, даже в сценариях без предварительных данных (zero-shot), благодаря использованию самогенерируемого обучения в контексте (SG-ICL) и механизма самокоррекции.

Объяснение метода:

Исследование предлагает адаптивный фреймворк для парсинга логов с использованием LLM, демонстрируя инновационные подходы к самокоррекции и обучению в контексте. Методы могут быть адаптированы для различных LLM и применены в других задачах с эволюционирующими данными. Концепции самогенерируемого обучения и самокоррекции имеют широкий потенциал применения, хотя требуют некоторой адаптации для широкой аудитории.

Ключевые аспекты исследования: 1. **AdaParser** - адаптивный фреймворк для парсинга логов, использующий большие языковые модели (LLM) с самогенерируемым обучением в контексте (SG-ICL) и самокоррекцией для эффективной обработки эволюционирующих логов.

Самогенерируемое обучение в контексте (SG-ICL) - компонент, который поддерживает динамический набор кандидатов из ранее сгенерированных шаблонов и выбирает демонстрации из этого набора для создания более точных запросов к LLM.

Корректор шаблонов - новый компонент, который использует LLM для исправления потенциальных ошибок парсинга в шаблонах, которые она генерирует, улучшая точность парсинга.

Древовидный парсер - компонент, использующий дерево парсинга для хранения шаблонов, сгенерированных LLM, и быстрого сопоставления шаблонов для новых сообщений логов, повышая эффективность.

Адаптивность к эволюционирующим логам - способность фреймворка работать с меняющимися логами без необходимости в обширных исторических данных, что делает его применимым в реальных сценариях.

Дополнение: Исследование AdaParser не требует дообучения или специального API для реализации его основных концепций. Хотя авторы использовали API для ChatGPT в своих экспериментах, они также продемонстрировали, что фреймворк работает с различными LLM, включая локальные открытые модели (DeepSeek-v2-chat и Qwen-1.5-72B-chat), которые могут быть развернуты локально.

Основные концепции и подходы, которые можно применить в стандартном чате:

Самогенерируемое обучение в контексте (SG-ICL) - пользователи могут сохранять успешные примеры взаимодействия с LLM и использовать их как демонстрации в будущих запросах. Например, если LLM успешно структурировала какой-то текст, этот пример можно включить в следующий запрос на структурирование похожего текста.

Стратегии самокоррекции - пользователи могут реализовать двухэтапный подход к запросам: сначала получить ответ, а затем задать уточняющий вопрос, указывающий на возможные ошибки. Например: "Проверь, правильно ли ты структурировал этот текст. Обрати внимание на X и Y."

Верификация результатов - пользователи могут запрашивать LLM проверить собственные ответы, задавая конкретные критерии проверки.

Динамический набор примеров - пользователи могут поддерживать и обновлять библиотеку успешных примеров взаимодействия с LLM для разных типов задач.

Результаты от применения этих концепций: - Повышение точности ответов LLM - Лучшая адаптация к специфическим задачам пользователя - Снижение необходимости в сложном промптинге для каждого запроса - Более эффективное использование контекстного окна LLM - Возможность работы с эволюционирующими данными и требованиями без необходимости в переобучении

Эти подходы особенно полезны в сценариях, когда требуется высокая точность или структурированный вывод, например при анализе документов, извлечении структурированной информации из текста или формализации знаний.

Prompt:

Использование знаний из исследования AdaParser в промптах для GPT ## Ключевые знания из исследования

Исследование AdaParser демонстрирует эффективный подход к парсингу логов с

использованием: - Самогенерируемого обучения в контексте (SG-ICL) - Механизма самокоррекции - Древовидного парсера для эффективной обработки

Пример промпта для парсинга логов с использованием принципов AdaParser

[======] Действуй как продвинутый парсер логов с функциями самообучения и самокоррекции, основанный на методологии AdaParser.

Вот набор логов, которые нужно проанализировать: [ВСТАВИТЬ ЛОГИ ЗДЕСЬ]

Выполни следующие шаги: 1. Сгенерируй шаблоны для каждого уникального типа лог-сообщения, абстрагируя переменные части (IP-адреса, временные метки, ID и т.д.) 2. Сгруппируй логи по этим шаблонам 3. Проведи самокоррекцию шаблонов, проверяя: - Нет ли слишком широких шаблонов, объединяющих разные типы сообщений - Нет ли слишком специфичных шаблонов, разделяющих однотипные сообщения 4. Для каждого шаблона извлеки ключевые переменные и их значения

Представь результаты в структурированном формате: - Список шаблонов с количеством соответствующих им сообщений - Для каждого шаблона: пример исходного лога и извлеченные переменные - Статистика распределения сообщений по шаблонам [======]

Как работают знания из исследования в этом промпте

Применение SG-ICL: Промпт инструктирует модель генерировать шаблоны на основе примеров логов и использовать их для дальнейшего анализа, что имитирует самогенерируемое обучение в контексте.

Механизм самокоррекции: Включен явный шаг проверки и исправления ошибок парсинга, фокусируясь на двух типах ошибок, выявленных в исследовании: слишком широкие шаблоны и неточные шаблоны.

Древовидный подход: Хотя GPT не может создать настоящую древовидную структуру, промпт направляет модель на группировку подобных логов, что концептуально соответствует древовидному парсеру AdaParser.

Адаптивность к новым форматам: Промпт не предполагает предварительных знаний о форматах логов, что позволяет модели адаптироваться к новым и эволюционирующим логам, как это делает AdaParser в режиме zero-shot.

Такой промпт позволяет максимально использовать способности GPT для анализа логов, применяя научно обоснованные подходы из исследования AdaParser.

№ 72. Улучшение манипуляций на уровне символов с помощью метода разделяй и властвуй

Ссылка: https://arxiv.org/pdf/2502.08180

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование направлено на улучшение способности больших языковых моделей (LLM) выполнять манипуляции на уровне символов. Основной вывод: LLM испытывают значительные трудности с базовыми операциями на уровне символов (удаление, вставка, замена), несмотря на их сильные способности в других задачах NLP. Предложенный метод 'Character Level Manipulation via Divide and Conquer' значительно улучшает точность этих операций без дополнительного обучения.

Объяснение метода:

Исследование предлагает практичный метод "разделяй и властвуй" для улучшения манипуляций с символами в LLM, который может быть применен без дополнительного обучения моделей. Подход решает реальную проблему обработки текста и значительно повышает точность базовых операций с символами. Требует некоторого технического понимания, но принципы доступны для адаптации широкой аудиторией.

Ключевые аспекты исследования: 1. **Выявление проблемы манипуляции символами в LLM**: Исследование обнаруживает, что современные LLM испытывают значительные трудности при выполнении базовых операций с символами (удаление, вставка, замена), несмотря на высокую точность в задачах правописания.

Анализ причин ограничений: Авторы определили, что проблема коренится в особенностях токенизации текста. LLM обрабатывают текст на уровне токенов, а не отдельных символов, что затрудняет манипуляции на уровне символов.

Метод "разделяй и властвуй": Предложен трехэтапный подход для улучшения манипуляций с символами: (1) атомизация токена на отдельные символы, (2) манипуляция на уровне символов, (3) реконструкция токена из модифицированной последовательности.

Экспериментальное подтверждение: Метод значительно повышает точность выполнения задач по удалению, вставке и замене символов без дополнительного обучения моделей.

Атомизированная структура слов: Исследование показывает, что представление слов в виде последовательности отдельных символов активирует скрытые способности LLM к рассуждениям на уровне символов.

Дополнение:

Исследование не требует дообучения или API для применения методов. Предложенный подход "разделяй и властвуй" полностью реализуем в стандартном чате с LLM через правильное форматирование запросов.

Основные концепции, применимые в стандартном чате:

Атомизация токенов: Разбивать слова на отдельные символы с пробелами между ними. Например, вместо "hello" использовать "h e l l o". Это помогает LLM работать с каждым символом отдельно.

Поэтапная манипуляция: Разбивать сложные операции на простые шаги:

Явно указывать декомпозицию слова на символы Четко описывать требуемую манипуляцию с каждым символом Запрашивать реконструкцию результата

Контроль автокоррекции: Направлять модель через процесс синтеза, чтобы она не возвращалась к общим формам слов.

Ожидаемые результаты: - Значительное повышение точности операций удаления, вставки и замены символов - Уменьшение количества ошибок автокоррекции - Более надежное выполнение нестандартных манипуляций с текстом

Эти подходы особенно полезны при работе с кодом, созданием нестандартных слов, обработкой специфических форматов данных и задачами редактирования текста.

Prompt:

Использование метода "Divide and Conquer" для символьных манипуляций в промптах ## Ключевая идея исследования

Исследование показывает, что LLM испытывают трудности с базовыми операциями на уровне символов (удаление, вставка, замена), но метод "Divide and Conquer" существенно повышает точность таких операций через три этапа: 1. **Атомизация токена** - разделение слова на отдельные символы 2. **Манипуляция на уровне символов** - выполнение нужной операции 3. **Реконструкция токена** - сборка результата обратно в слово

Пример промпта для символьных манипуляций

[======] Я хочу, чтобы ты выполнил точную манипуляцию с символами в слове "programming" - удали третью букву и замени пятую букву на символ '@'.

Пожалуйста, используй метод "Divide and Conquer":

Сначала раздели слово на отдельные символы (атомизируй его): p r o g r a m m i n g

Затем выполни манипуляции с символами:

Удали третью букву (o) Замени пятую букву (r) на символ '@'

Наконец, реконструируй слово, объединив символы обратно в единое слово.

Покажи каждый шаг процесса и финальный результат. [======]

Почему это работает

Этот подход работает эффективнее, потому что:

Активирует скрытые знания модели - атомизированная форма слов лучше активирует внутренние представления модели о символах **Упрощает сложную задачу** - разбивая операцию на явные подзадачи, мы снижаем когнитивную нагрузку на модель **Обеспечивает контроль** - явное разделение этапов позволяет модели сосредоточиться на каждой подзадаче отдельно Такой структурированный подход особенно полезен для задач, требующих точных символьных манипуляций, работы с редкими словами или специальными терминами.

№ 76. Скамейка LCTG: Бенчмарк генерации текста с контролем LLM

Ссылка: https://arxiv.org/pdf/2501.15875

Рейтинг: 75
Адаптивность: 80

Ключевые выводы:

Исследование представляет LCTG Bench - первый японский бенчмарк для оценки контролируемости (управляемости) больших языковых моделей (LLM) при генерации текста. Основная цель - создать унифицированную систему оценки способности LLM следовать конкретным инструкциям при генерации текста на японском языке. Результаты показали значительный разрыв в производительности между многоязычными моделями (GPT-4, GPT-3.5, Gemini-Pro) и японскими моделями, а также выявили общие проблемы с контролем количества символов во всех моделях.

Объяснение метода:

Исследование предлагает универсальную методологию контроля генерации текста по четырем аспектам (формат, количество символов, ключевые/запрещенные слова), применимую в любых LLM. Представленные структуры промптов и подходы к оценке могут быть непосредственно использованы пользователями для повышения качества взаимодействия с чат-моделями. Выявленные особенности разных моделей помогают выбрать оптимальную для конкретных задач.

Ключевые аспекты исследования: 1. **LCTG Bench** - первый японский бенчмарк для оценки управляемости (контролируемости) LLM при генерации текста, позволяющий выбрать наиболее подходящую модель для различных сценариев использования.

Четыре аспекта контролируемости генерации текста: Format (формат), Character Count (количество символов), Keyword (ключевые слова), Prohibited Word (запрещенные слова), которые оцениваются единообразно в трех задачах.

Три генеративные задачи: Summarization (суммаризация), Ad Text Generation (генерация рекламного текста) и Pros & Cons Generation (генерация плюсов и минусов), каждая с различными характеристиками для всесторонней оценки.

Методология оценки: использование правило-ориентированной проверки для измерения контролируемости и GPT-4 для оценки качества генерируемого содержимого.

Выявление разрыва в производительности между многоязычными моделями

(GPT-4, GPT-3.5, Gemini-Pro) и японскими моделями в контексте контролируемости генерации текста.

Дополнение: Для работы с методами этого исследования не требуется дообучение или API. Хотя авторы использовали GPT-4 для оценки качества и постобработки результатов, основные концепции и подходы полностью применимы в стандартном чате с любой LLM.

Концепции и подходы, которые можно применить в стандартном чате:

Четыре аспекта контролируемости: FORMAT: указание в промпте "выведи только результат, без пояснений" C-COUNT: указание точного количества символов/слов в выводе KEYWORD: требование использовать определенные ключевые слова P-WORD: запрет на использование определенных слов

Структура промптов:

Трехчастная структура: инструкция задачи + условие + базовый текст Четкое разделение условий от основной инструкции

Понимание ограничений моделей:

Учет того, что контроль количества символов может быть проблематичным Подготовка к тому, что модель может добавлять объяснения, даже если их не просили Применяя эти концепции, пользователи могут получить: - Более точное соответствие выводов заданным требованиям - Лучшее понимание ограничений моделей - Более эффективные стратегии формулирования запросов - Возможность контролировать включение/исключение определенного содержимого

Примечательно, что даже GPT-4 показывает низкие результаты при контроле точного количества символов, что подсказывает пользователям необходимость проверки и возможной постобработки результатов при работе с такими ограничениями.

Prompt:

Использование результатов исследования LCTG Bench в промптах для GPT ## Ключевые инсайты для создания эффективных промптов

Исследование LCTG Bench предоставляет ценную информацию о контролируемости языковых моделей, которую можно использовать для оптимизации промптов:

Многоязычные модели превосходят специализированные (особенно для японского языка) **Контроль количества символов** — слабое место всех моделей **Четкие инструкции по формату** значительно улучшают результат **Явное указание ключевых и запрещенных слов** требует особого внимания ## Пример оптимизированного промпта

[======] # Задание: Создание рекламного текста для японского ресторана

Требования к формату: - Структура: заголовок, 3 абзаца основного текста, призыв к действию - Длина: ровно 400 символов (не токенов) - Заголовок выделить жирным шрифтом - Не добавлять пояснения до и после текста

Обязательные элементы: - Ключевые слова для включения: "аутентичный", "свежие ингредиенты", "традиции" - Запрещенные слова: "дешевый", "быстрый", "фастфуд"

Дополнительные инструкции: - После создания текста проверь количество символов и скорректируй до точного соответствия требованию в 400 символов - Убедись, что все ключевые слова включены естественным образом - Подтверди отсутствие всех запрещенных слов

Пожалуйста, создай рекламный текст, строго соблюдая все указанные требования. [======]

Почему это работает

Данный промпт учитывает выводы исследования LCTG Bench следующим образом:

Четкая структура формата — исследование показало, что модели лучше справляются с задачами, когда формат четко определен

Явное указание на проверку количества символов — компенсирует слабое место всех моделей (C-COUNT), заставляя модель дополнительно проверить этот параметр

Выделение ключевых и запрещенных слов в отдельные списки — улучшает понимание модели о том, что должно и не должно быть включено

Запрет на пояснительные тексты — решает проблему, когда модели добавляют ненужные пояснения в начале или конце ответа

Дополнительные инструкции по самопроверке — заставляют модель провести внутреннюю валидацию результата перед выдачей ответа

Такой подход к составлению промптов значительно повышает вероятность получения текста, соответствующего всем заданным параметрам контролируемости.

№ 80. Запоминание вместо рассуждения? Обнаружение и снижение verbatim запоминания в оценке понимания персонажей большими языковыми моделями

Ссылка: https://arxiv.org/pdf/2412.14368

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование направлено на выявление и смягчение проблемы дословного запоминания (verbatim memorization) в задачах понимания персонажей большими языковыми моделями (LLM). Основные результаты показывают, что LLM часто полагаются на дословное запоминание популярных художественных произведений вместо настоящего понимания и рассуждения о персонажах. Предложенный подход, основанный на концепции «gist memory» (запоминание сути), позволяет снизить зависимость моделей от дословного запоминания и стимулировать более глубокое понимание персонажей.

Объяснение метода:

Исследование предлагает практические методы промптинга, стимулирующие LLM к рассуждениям вместо воспроизведения запомненной информации. Концепции "gist memory" и "verbatim memory" имеют высокую образовательную ценность. Пользователи могут непосредственно применять предложенные промпты для получения более осмысленных ответов, особенно при анализе художественных произведений. Однако некоторые методы требуют адаптации для широкого использования.

Ключевые аспекты исследования: 1. **Выявление проблемы дословного запоминания**: Исследование показывает, что языковые модели (LLM) часто демонстрируют хорошие результаты в задачах понимания персонажей не благодаря реальному пониманию, а из-за дословного запоминания популярных художественных произведений из обучающих данных.

Концепции "gist memory" и "verbatim memory": Авторы используют когнитивные концепции "обобщенной памяти" (gist memory), которая фокусируется на общем смысле, и "дословной памяти" (verbatim memory), запоминающей точные детали. Это позволяет разработать методы, стимулирующие использование моделями рассуждений, а не механического воспроизведения.

Методы снижения зависимости от запоминания: Предложены два основных

метода: "hard setting" (замена имен персонажей) и "soft setting" (специальные промпты, направляющие модель к использованию рассуждений вместо запоминания).

Экспериментальные результаты: Применение предложенных методов приводит к значительному снижению производительности моделей (до 45.8%), что подтверждает их зависимость от запоминания, а не реального понимания персонажей.

Промпты, основанные на обобщенной памяти: Авторы разработали специальные промпты для различных задач понимания персонажей, которые стимулируют модели использовать рассуждения вместо воспроизведения запомненного контента.

Дополнение:

Применимость методов исследования в стандартном чате

Исследование не требует дообучения моделей или специального API для применения основных методов. Хотя авторы использовали некоторые расширенные техники (например, для создания датасетов), основные концепции и подходы могут быть применены в стандартном чате.

Ключевые концепции для применения в стандартном чате:

Промпты, основанные на "gist memory": Пользователи могут формулировать запросы, стимулирующие модель к анализу отношений, характеров и ключевых событий, а не к прямому воспроизведению информации. Например: "Проанализируй отношения между персонажами в этом диалоге, основываясь на их речевых паттернах и поведении" "На основе действий и высказываний, какие черты личности демонстрирует этот персонаж?"

Минимизация прямых ссылок на популярные произведения: При анализе художественных произведений можно избегать прямого упоминания названий и имен персонажей, заменяя их обобщенными обозначениями (например, "главный герой", "второстепенный персонаж").

Явное указание на использование рассуждений: Включение в промпт инструкций типа "не полагайся на запоминание диалогов, а используй логические рассуждения" или "выведи ответ из анализа текста, а не из знания о произведении".

Ожидаемые результаты:

Более оригинальные и глубокие анализы художественных произведений, не ограниченные запомненными шаблонами. Развитие навыков формулирования запросов, стимулирующих реальные рассуждения LLM. Более критичное отношение к ответам LLM о популярных произведениях культуры. Повышение качества творческих и аналитических задач, выполняемых с помощью LLM.

Prompt:

Как использовать знания из исследования о verbatim запоминании в промптах для GPT ## Ключевые идеи исследования для применения

Исследование показало, что большие языковые модели (LLM) часто полагаются на дословное запоминание текстов вместо настоящего понимания. Особенно это проявляется при работе с популярными произведениями. Используя принципы "gist memory" (запоминание сути) вместо дословного воспроизведения, можно значительно улучшить качество анализа.

Пример промпта, основанного на исследовании

[======] Проанализируй отношения между главными героями романа "Война и мир", следуя этим принципам:

Избегай дословного цитирования текста - вместо этого сосредоточься на ключевых паттернах отношений Выдели основные черты характера персонажей через их поступки и развитие Опиши эволюцию отношений между Наташей Ростовой и Андреем Болконским, фокусируясь на: Психологических мотивах их действий Ключевых поворотных моментах в их отношениях Внутренних конфликтах каждого персонажа Объясни своё рассуждение, опираясь на общее понимание произведения, а не на конкретные цитаты или эпизоды. [======]

Почему этот подход работает лучше

Стимулирует глубокое понимание: Промпт направляет модель на анализ сути взаимоотношений, а не на воспроизведение запомненных фрагментов

Фокусируется на рассуждении: Запрашивает объяснение логики и психологических мотивов, что требует от модели создания связей между событиями

Избегает ловушек дословного запоминания: Не просит цитировать конкретные эпизоды, что могло бы активировать механизм verbatim запоминания

Использует принцип "gist memory": Направляет модель на обобщение и анализ паттернов, а не на воспроизведение деталей

Такой подход особенно полезен при работе с популярными произведениями, где у модели может быть сильное дословное запоминание текста, что мешает настоящему аналитическому рассуждению.

№ 84. От Системы 1 к Системе 2: Обзор Рассуждений Больших Языковых Моделей

Ссылка: https://arxiv.org/pdf/2502.17419

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование представляет комплексный обзор развития Reasoning LLMs (языковых моделей с улучшенными способностями рассуждения), которые стремятся перейти от быстрого интуитивного мышления (System 1) к более медленному, но глубокому аналитическому мышлению (System 2). Основная цель - проанализировать методы, позволяющие LLMs выполнять сложные многошаговые рассуждения, подобные человеческим, и оценить их эффективность в различных задачах.

Объяснение метода:

Исследование предоставляет ценное понимание принципов рассуждения в LLM, которые могут быть адаптированы в виде техник промптинга (структурированное рассуждение, верификация шагов, макро-действия). Понимание различий между System 1 и System 2 помогает пользователям эффективнее формулировать запросы для разных типов задач, хотя некоторые методы требуют технической подготовки и адаптации для широкого применения.

Ключевые аспекты исследования: 1. **Переход от System 1 к System 2 мышлению в LLM**: Исследование фокусируется на эволюции языковых моделей от быстрого интуитивного мышления (System 1) к более медленному, аналитическому и целенаправленному рассуждению (System 2), что приближает LLM к человеческим когнитивным способностям.

Методы реализации рассуждений в LLM: В работе представлен комплексный обзор ключевых технологий, обеспечивающих продвинутые возможности рассуждения в LLM, включая структурированный поиск (MCTS), моделирование вознаграждения, самосовершенствование, макро-действия и RL-настройку.

Эволюция моделей рассуждения: Исследование прослеживает эволюцию от внешних алгоритмов рассуждения к встроенным механизмам рассуждения в LLM, с особым вниманием к моделям типа OpenAI o1/o3 и DeepSeek R1, которые демонстрируют экспертный уровень в сложных задачах.

Бенчмаркинг и оценка: Авторы представляют подробный анализ существующих бенчмарков и метрик оценки, а также сравнивают производительность различных моделей рассуждения на текстовых и мультимодальных задачах.

Будущие направления исследований: Работа определяет ключевые вызовы и перспективные направления, включая эффективность рассуждений, коллаборативные системы быстрого/медленного мышления, применение в научных областях, интеграцию нейронных и символьных систем и мультиязычность.

Дополнение:

Исследование представляет множество методов и подходов, которые первоначально могут показаться применимыми только при дообучении моделей или через API, однако многие концепции могут быть успешно адаптированы для стандартного чата без специальных технических возможностей.

Ключевые концепции и подходы, применимые в стандартном чате:

Структурированное рассуждение (Structure Search) - можно реализовать через промпты, которые: Просят модель рассматривать проблему поэтапно Предлагают исследовать несколько путей решения Указывают на необходимость проверки промежуточных результатов

Моделирование вознаграждения (Reward Modeling) - адаптируется через:

Запросы на оценку качества промежуточных шагов Просьбы проверить логику рассуждений на каждом этапе Указания на критерии успешного решения

Самосовершенствование (Self Improvement) - реализуется через:

Просьбы к модели критически пересмотреть свои ответы Запросы на поиск ошибок в собственных рассуждениях Итеративное улучшение решения через серию уточняющих вопросов

Макро-действия (Macro Action) - применяются через:

Структурирование запроса по этапам ("Сначала проанализируй..., затем предложи...") Использование специальных маркеров для обозначения различных мыслительных процессов Имитацию диалога между различными "мыслительными агентами" Примеры практического применения:

• Для математических задач: Запрашивать пошаговое решение с проверкой каждого шага, а затем просить модель критически пересмотреть решение и найти возможные ошибки.

• Для принятия решений: Структурировать процесс через исследование альтернатив, оценку каждой по заданным критериям, а затем синтез окончательного решения.

• Для анализа текста: Использовать структурированный подход, где модель сначала

выделяет ключевые идеи, затем анализирует их взаимосвязи, и наконец формирует общий вывод.

Эти методы не требуют дообучения или API, но позволяют значительно улучшить качество ответов за счет более структурированного и тщательного рассуждения.

Prompt:

Применение исследования о рассуждениях LLM в промптах для GPT ## Ключевые концепции для использования

Исследование "От Системы 1 к Системе 2" описывает переход LLM от интуитивного мышления к аналитическому, выделяя пять ключевых методов: - Structure Search - Reward Modeling - Self-Improvement - Macro Action - Reinforcement Fine-Tuning

Эти методы можно творчески применить при составлении промптов для GPT.

Пример промпта с использованием знаний из исследования

[======] # Задача: Решение сложной бизнес-проблемы

Инструкции Я хочу, чтобы ты использовал структурированный подход рассуждения (System 2) для анализа следующей бизнес-проблемы. Применяй следующие техники:

Макро-действия: Сначала спланируй свой анализ, разбей его на высокоуровневые шаги. **Структурированный поиск**: Рассмотри несколько альтернативных путей решения (минимум 3), оценивая перспективность каждого. **Самопроверка**: После формулирования решения, критически проанализируй его, найди потенциальные ошибки и исправь их. **Пошаговое рассуждение**: Для каждого важного вывода приводи обоснование, не пропуская логические шаги. ## Бизнес-проблема [Описание проблемы]

Пожалуйста, представь свой анализ в структурированном формате, с четким разделением планирования, исследования альтернатив, формулирования решения и проверки. [======]

Как это работает

Данный промпт использует ключевые концепции из исследования:

Macro Action - промпт явно требует разбить решение на высокоуровневые шаги, что помогает GPT организовать процесс рассуждения.

Structure Search - запрос рассмотреть несколько альтернативных путей имитирует метод поиска по дереву решений, подобный MCTS из исследования.

Self-Improvement - требование самопроверки заставляет модель критически оценивать собственные выводы и исправлять ошибки.

Process Reward Modeling - акцент на обосновании каждого шага, а не только конечного результата, отражает идею PRM из исследования.

Такой промпт направляет GPT к использованию более глубокого аналитического мышления (System 2) вместо быстрого интуитивного ответа (System 1), что особенно полезно для сложных задач, требующих многошагового рассуждения.

№ 88. Изучение влияния больших языковых моделей на пользовательские истории, созданные студентами, и тестирование приемки в разработке программного обеспечения

Ссылка: https://arxiv.org/pdf/2502.02675

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование направлено на изучение влияния LLM (больших языковых моделей) на способность студентов-программистов преобразовывать отзывы пользователей в пользовательские истории (user stories) в рамках Agile-методологии. Главные результаты показали, что LLM значительно улучшают способность студентов создавать критерии приемки (acceptance criteria) и повышают ценность пользовательских историй, но при этом студенты без помощи LLM лучше справляются с созданием историй подходящего объема.

Объяснение метода:

Исследование дает конкретные данные о том, где LLM помогают (критерии приемки +88%, ценность формулировок +23%) и где мешают (определение объема -24%). Методика работы с LLM универсально применима. Пользователи получают важное концептуальное понимание: LLM эффективны для детализации, но требуют контроля объема задач.

Ключевые аспекты исследования: 1. Исследование изучает влияние LLM на способность студентов трансформировать отзывы пользователей в пользовательские истории (user stories) в контексте разработки программного обеспечения.

Студенты создавали пользовательские истории дважды: без помощи LLM и с помощью LLM (ChatGPT 3.5), следуя принципам INVEST (Independent, Negotiable, Valuable, Estimable, Small/Scope, Testable).

Использование LLM значительно улучшило способность студентов создавать критерии приемки (acceptance criteria) для пользовательских историй (+88%) и повысило ценность (value) историй (+23%).

Однако студенты, не использующие LLM, лучше справлялись с созданием историй подходящего объема (small/scope) — использование LLM снизило показатели по этому параметру на 24%.

Не было обнаружено статистически значимых различий по другим атрибутам INVEST (независимость, возможность обсуждения и возможность оценки).

Дополнение:

Применимость методов в стандартном чате

Исследование не требует дообучения моделей или специального API для применения описанных методов. Все подходы можно использовать в стандартном чате с любой современной LLM. Исследователи использовали ChatGPT 3.5, который доступен широкой аудитории.

Ключевые концепции для применения в стандартном чате:

Структурированные промпты для пользовательских историй: Создание отдельного промпта для каждой истории Включение контекста приложения Указание конкретных принципов (например, INVEST) Запрос на генерацию критериев приемки

Критический анализ результатов:

Особенно внимательная проверка объема предложенных задач Акцент на использование LLM для детализации критериев успеха Ручное редактирование результатов при необходимости

Избирательное использование LLM:

Использование для структурирования критериев приемки Использование для повышения ценности формулировок Самостоятельное определение объема задач ### Ожидаемые результаты: - Более детальные и тестируемые критерии успеха для любых проектов - Более ценные и понятные формулировки задач - Избежание проблем с завышенным объемом работ при правильном применении

Prompt:

Применение исследования о влиянии LLM на пользовательские истории ## Ключевые знания из исследования

Исследование показало, что использование LLM: - Улучшает качество **критериев приемки** (+88%) - Повышает **ценность** пользовательских историй (+23%) - Но ухудшает **размер** историй (делает их слишком большими, -24%)

Пример промпта на основе исследования

[======] Помоги мне создать пользовательскую историю и критерии приемки на основе следующего отзыва пользователя: [ОТЗЫВ ПОЛЬЗОВАТЕЛЯ]

При создании следуй этим правилам: 1. Сформулируй историю в формате: "Как [роль пользователя], я хочу [функциональность], чтобы [ценность/польза]" 2. Сделай историю ЦЕННОЙ - четко объясни, какую пользу получит пользователь 3. Разработай ПОДРОБНЫЕ критерии приемки, которые можно легко преобразовать в тест-кейсы 4. ВАЖНО: Убедись, что история имеет небольшой объем и сфокусирована на одной конкретной функции 5. Следуй принципам INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable)

После создания истории, пожалуйста, проверь, не является ли она слишком большой, и при необходимости раздели на несколько меньших историй. [======]

Почему этот промпт работает

Данный промпт учитывает основные выводы исследования:

Усиливает сильные стороны LLM: Запрашивает подробные критерии приемки (область, где LLM показали +88% улучшение) Подчеркивает важность ценности для пользователя (область с +23% улучшением)

Компенсирует слабые стороны LLM:

Специально обращает внимание на необходимость небольшого размера истории Просит проверить и разделить историю, если она получилась слишком большой

Использует структурированный подход, опираясь на принципы INVEST, которые были частью методологии исследования

Такой подход позволяет максимизировать преимущества LLM, минимизируя их недостатки при создании пользовательских историй.

№ 92. Обнаружение неэффективностей в коде, сгенерированном LLM: к всеобъемлющей таксономии

Ссылка: https://arxiv.org/pdf/2503.06327

Рейтинг: 75
Адаптивность: 80

Ключевые выводы:

Исследование направлено на выявление и систематизацию неэффективностей в коде, генерируемом большими языковыми моделями (LLM). Авторы разработали таксономию неэффективностей, включающую 5 категорий и 19 подкатегорий, и обнаружили, что проблемы с логикой и производительностью являются наиболее распространенными и часто взаимосвязанными с другими типами неэффективностей.

Объяснение метода:

Исследование предлагает практичную таксономию неэффективностей в коде, генерируемом LLM, которая может служить чеклистом при проверке кода. Выявленные категории проблем (логика, производительность, читаемость, сопровождаемость, ошибки) и их взаимосвязи помогают пользователям формировать более точные запросы и критически оценивать результаты. Опрос практиков подтверждает актуальность проблем для реальной разработки.

Ключевые аспекты исследования: 1. **Таксономия неэффективностей кода, генерируемого LLM**: Исследование систематизирует и классифицирует типичные недостатки в коде, создаваемом языковыми моделями, выделяя 5 основных категорий (Общая логика, Производительность, Читаемость, Сопровождаемость, Ошибки) и 19 подкатегорий.

Эмпирический анализ кода: Авторы проанализировали 492 фрагмента кода, сгенерированных тремя популярными открытыми моделями (CodeLlama, DeepSeek-Coder, CodeGemma), определив частоту и характер различных типов неэффективностей.

Валидация через опрос специалистов: Исследование включает опрос 58 практикующих разработчиков и исследователей, использующих LLM для кодирования, что подтверждает актуальность выявленных проблем и их важность для реальных пользователей.

Выявление взаимосвязей между типами неэффективностей: Исследование анализирует, как различные проблемы взаимосвязаны и часто встречаются вместе,

что помогает понять их комплексное влияние на качество кода.

Рекомендации для улучшения моделей и практики использования: На основе выявленных шаблонов неэффективностей авторы предлагают направления для совершенствования моделей и практик работы с генерируемым кодом.

Дополнение: Исследование не требует дообучения или API для применения основных методов и подходов. Основной вклад работы — таксономия неэффективностей, которая может быть непосредственно использована в стандартном чате с LLM.

Вот ключевые концепции и подходы, которые можно применить в обычном чате:

Структурированная проверка кода — использование 5 категорий неэффективностей (Общая логика, Производительность, Читаемость, Сопровождаемость, Ошибки) в качестве фреймворка для оценки сгенерированного кода.

Целенаправленные промпты — формулировка запросов с учетом выявленных типичных проблем:

"Сгенерируй код с оптимальной временной сложностью" "Учти обработку крайних случаев и исключений" "Избегай избыточных условных блоков и повторяющегося кода"

Метапромптинг — можно попросить LLM проанализировать свой собственный код на предмет выявленных неэффективностей:

Проанализируй сгенерированный код на наличие следующих проблем: 1. Ошибки в основной логике 2. Неоптимальная производительность (время/память) 3. Проблемы с читаемостью 4. Сложности сопровождения 5. Синтаксические ошибки или отсутствующие импорты

Итеративное улучшение — исследование показывает, что часто проблемы взаимосвязаны, поэтому можно последовательно улучшать код: Улучши этот код, сначала исправив логические ошибки, затем оптимизируй производительность и, наконец, улучши читаемость и сопровождаемость.

Чеклист для самопроверки — пользователь может создать собственный чеклист на основе таксономии и применять его к любому сгенерированному коду. Результаты применения этих подходов: - Повышение качества сгенерированного кода - Сокращение времени на отладку и рефакторинг - Более глубокое понимание ограничений LLM и способов их преодоления - Формирование более эффективных привычек работы с LLM для генерации кода

Важно отметить, что исследование показывает наиболее частые проблемы (логика и производительность), что позволяет пользователям сосредоточиться на них в первую очередь при проверке сгенерированного кода.

Prompt:

Использование таксономии неэффективностей LLM-кода в промптах Исследование о неэффективностях в коде, генерируемом LLM, предоставляет ценные знания, которые можно использовать для улучшения промптов при работе с кодом. Вот как это можно применить:

Пример промпта с учетом исследования

[======] Напиши функцию на Python для поиска самого длинного палиндрома в строке.

При создании решения, пожалуйста:

Сначала сфокусируйся на корректности логики, так как согласно исследованиям, 68.5% ошибок в LLM-коде связаны с логическими проблемами Оптимизируй временную сложность (стремись к O(n)), поскольку неоптимальная временная сложность встречается в 18.5% случаев Обработай все граничные случаи (пустая строка, строка из одного символа) Добавь понятные комментарии к ключевым частям алгоритма Избегай избыточного кода и ненужных условных блоков После написания функции, проанализируй свое решение на наличие: Проблем с логикой Неоптимальной временной или пространственной сложности Проблем с читаемостью и сопровождаемостью Потенциальных ошибок Предоставь окончательное оптимизированное решение с анализом временной и пространственной сложности. [======]

Как работают знания из исследования в этом промпте

Промпт учитывает ключевые проблемные области, выявленные в исследовании:

Приоритизация логики (68.5% ошибок) - явно просим модель сфокусироваться на корректности логики в первую очередь

Акцент на производительности (34.15% ошибок) - запрашиваем оптимизацию временной сложности и указываем желаемый результат

Обработка граничных случаев - это часть проблем с логикой, которые часто упускаются

Читаемость и сопровождаемость (4.67% и 21.14%) - просим добавить комментарии и избегать избыточного кода

Самопроверка - просим модель проанализировать свое решение по всем категориям из таксономии неэффективностей

Такой структурированный промпт помогает предотвратить наиболее распространенные проблемы, выявленные в исследовании, и получить более качественный код.

№ 96. Сравнение кода, написанного человеком, и кода, сгенерированного ИИ: Вердикт всё ещё не вынесен!

Ссылка: https://arxiv.org/pdf/2501.16857

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование сравнивает качество кода, написанного людьми и сгенерированного большими языковыми моделями (LLM, конкретно GPT-4). Основная цель - оценить, насколько эффективны LLM в создании программного кода по сравнению с человеческими программистами. Результаты показывают, что код, написанный людьми, лучше соответствует стандартам кодирования, но код GPT-4 чаще проходит функциональные тесты. При этом LLM часто создают более сложный код и испытывают трудности с задачами, требующими глубоких предметных знаний.

Объяснение метода:

Исследование предоставляет практически применимые выводы о сравнении кода, написанного людьми и сгенерированного LLM. Результаты показывают, что LLM лучше в стандартных задачах, но отстают в сложных. Выводы о функциональных различиях, безопасности и сложности кода напрямую полезны для широкого круга пользователей.

Ключевые аспекты исследования: 1. **Сравнительный анализ кода**: Исследование проводит систематическое сравнение кода, написанного людьми и сгенерированного LLM (GPT-4), используя 72 различных задачи программирования на Python.

Многомерная оценка качества: Работа оценивает код по четырем ключевым критериям: соответствие стандартам кодирования Python (с использованием Pylint), безопасность и уязвимости (с использованием Bandit), сложность кода (с использованием Radon) и функциональная корректность (с использованием тестов Pytest).

Функциональные различия: Выявлены области, где LLM превосходит людей (стандартные задачи, проходимость тестов) и где отстает (сложные задачи, требующие глубоких доменных знаний и творческого мышления).

Безопасность кода: Обнаружены уязвимости как в коде, написанном людьми, так и сгенерированном LLM, с более серьезными выбросами в коде LLM.

Сложность кода: LLM генерирует в среднем более сложный код (на 61% выше по цикломатической сложности), что может затруднять его поддержку и понимание.

Дополнение: Исследование не требует дообучения моделей или специального API для применения его методов и выводов. Все концепции и подходы могут быть адаптированы для работы в стандартном чате с LLM.

Основные концепции, которые можно применить в стандартном чате:

Выбор типа задач для LLM: Исследование показывает, что LLM лучше справляются со стандартными, хорошо определенными задачами, но отстают в сложных задачах, требующих глубоких доменных знаний. Пользователи могут использовать LLM для рутинных задач программирования, но полагаться на свои навыки для более сложных проблем.

Проверка безопасности: Зная о типичных уязвимостях в коде, сгенерированном LLM (небезопасное использование подпроцессов, жестко закодированные конфиденциальные данные, использование небезопасных библиотек), пользователи могут проверять сгенерированный код на эти проблемы.

Упрощение сложного кода: Понимая, что LLM генерирует более сложный код, пользователи могут запрашивать более простые решения или просить упростить полученный код.

Оценка функциональности: Исследование показывает, что код LLM часто проходит больше тестов, чем человеческий код. Пользователи могут ожидать высокой функциональности от сгенерированного кода, но также должны проверять его на соответствие требованиям.

Улучшение документации: Зная о проблемах с документацией в коде LLM, пользователи могут специально запрашивать хорошо документированный код или добавлять документацию самостоятельно.

Применение этих концепций позволит пользователям более эффективно использовать LLM для генерации кода, понимая их сильные и слабые стороны, и получать более качественные результаты.

Prompt:

Использование знаний из исследования в промтах для GPT ## Ключевые выводы для создания эффективных промтов

Исследование показывает, что код GPT-4: - Лучше проходит функциональные тесты (87.3% vs 54.9% у людей) - Имеет более высокую цикломатическую сложность (5.0 vs 3.1) - Хуже справляется с задачами, требующими глубоких предметных знаний - Может содержать проблемы безопасности

Пример промта с учетом этих знаний

[======] # Задача: Создать функцию для обработки пользовательских данных

Требования: 1. Напиши Python-функцию process_user_data(user_input), которая валидирует и очищает пользовательский ввод. 2. Функция должна обрабатывать строки, содержащие имя, email и возраст.

Специальные инструкции (с учетом исследования): - Стремись к низкой цикломатической сложности (не более 3-4) для лучшей поддерживаемости - Уделяй особое внимание безопасности кода, особенно при валидации пользовательского ввода - Следуй стандартам PEP 8 для Python - Добавь комментарии, объясняющие логику работы - Включи простые примеры использования функции - Предоставь несколько тестовых случаев для проверки функциональности

Ожидаемый результат: Хорошо структурированная, безопасная и эффективная функция с низкой сложностью. [======]

Объяснение эффективности

Этот промт учитывает ключевые выводы исследования следующим образом:

Контроль сложности: Явно ограничивает цикломатическую сложность, так как исследование показало, что код GPT-4 обычно более сложный (5.0 vs 3.1 у людей)

Акцент на безопасности: Требует особого внимания к безопасности, что решает выявленную проблему с уязвимостями в коде LLM

Соблюдение стандартов: Запрашивает соответствие PEP 8, что улучшает структурированность кода (где люди показали преимущество)

Документация и тесты: Запрашивает комментарии и тестовые случаи, чтобы использовать сильную сторону GPT-4 в прохождении функциональных тестов

Такой структурированный промт помогает компенсировать выявленные в исследовании слабости GPT и использовать его сильные стороны, что приводит к более качественному и поддерживаемому коду.

№ 100. Агентное извлечение информации

Ссылка: https://arxiv.org/pdf/2410.09713

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование представляет концепцию агентного информационного поиска (Agentic IR) как новой парадигмы, расширяющей традиционный информационный поиск с помощью LLM-управляемых ИИ-агентов. Основная цель - переопределить информационный поиск от статического получения элементов информации к достижению целевых информационных состояний в динамической среде.

Объяснение метода:

Исследование вводит концепцию агентного информационного поиска, переопределяя взаимодействие с LLM как достижение "информационного состояния", а не просто получение информации. Предлагает практичный подход к многошаговому взаимодействию с LLM для решения комплексных задач. Концепции и примеры применимы сразу, без дополнительных инструментов, хотя полная реализация некоторых возможностей может требовать API-доступа.

Ключевые аспекты исследования: 1. **Новая парадигма информационного поиска**: Исследование вводит концепцию "Agent IC Information Retrieval" (Агентного информационного поиска), который переопределяет информационный поиск от простого получения релевантных элементов из предопределенного корпуса к достижению желаемого "информационного состояния" в динамической среде.

Информационное состояние как объект поиска: Вместо статичных информационных элементов вводится понятие "информационного состояния" пользователя, которое включает не только полученную информацию, но и контекст, предпочтения пользователя и процессы принятия решений.

Архитектура агентных систем: Исследование описывает архитектуру агентных систем для информационного поиска, включая ключевые компоненты агентов (профиль, память, планирование, действия) и дизайн систем (одноагентные и мультиагентные).

Практические применения: Представлены два конкретных практических применения - персональный ассистент и бизнес-ассистент, демонстрирующие возможности агентного информационного поиска в реальных сценариях.

Оценка и оптимизация систем: Предложены новые метрики и протоколы оценки для агентных систем, а также методы их оптимизации, учитывающие не только

релевантность результатов, но и эффективность, полезность и этические аспекты.

Дополнение: Исследование представляет концепцию агентного информационного поиска, которая может быть применена в стандартном чате без необходимости дообучения или API. Хотя авторы для своих экспериментов могли использовать расширенные возможности, основные концепции применимы в обычном взаимодействии с LLM.

Концепции и подходы для применения в стандартном чате:

Информационное состояние как цель: Вместо простого запроса информации, пользователь может сформулировать желаемое "информационное состояние" - конечный результат, которого он хочет достичь. Например: "Я хочу спланировать поездку в Японию на 7 дней с бюджетом $2000".

Многошаговое взаимодействие: Пользователь может разбить сложную задачу на последовательность шагов и провести модель через эти шаги. Например, сначала определить маршрут, затем жилье, затем активности.

Итеративное уточнение: Пользователь может постепенно уточнять информацию на основе промежуточных результатов. Например: "Теперь, когда мы выбрали города, давай подберем отели в каждом из них".

Планирование действий: Пользователь может явно попросить модель составить план действий для достижения цели. Например: "Составь план действий для подготовки научной статьи".

Проактивный сбор информации: Пользователь может попросить модель определить, какая дополнительная информация нужна для решения задачи. Например: "Какую еще информацию тебе необходимо знать, чтобы помочь мне выбрать оптимальный маршрут?"

Ожидаемые результаты применения:

Более структурированные и целенаправленные взаимодействия с LLM Повышение эффективности решения сложных задач Улучшение качества получаемой информации благодаря более четкому определению цели Более персонализированные результаты из-за постепенного уточнения предпочтений Снижение когнитивной нагрузки на пользователя при решении сложных задач Даже без дополнительных API или инструментов, применение этих концепций может значительно улучшить опыт взаимодействия с LLM и повысить качество получаемых результатов.

Prompt:

Использование концепции агентного информационного поиска в промтах для GPT Исследование агентного информационного поиска (Agentic IR) предлагает новый

подход к взаимодействию с языковыми моделями, переходя от простого получения информации к динамическому достижению информационных состояний через серию целенаправленных действий.

Ключевые концепции для применения в промтах

Динамические информационные состояния вместо статических запросов **Модульность агента** (профиль, память, планирование, действие) **Итеративное уточнение запросов** и адаптация к обратной связи **Проактивное планирование** для достижения информационных целей ## Пример промта, использующего принципы агентного IR

[======] # Задача: Помощь в планировании деловой поездки в Сингапур

Профиль агента Ты - бизнес-ассистент с возможностями агентного информационного поиска. Твоя цель - не просто предоставить информацию, а помочь мне достичь целевого информационного состояния для успешной деловой поездки.

##

№ 104. Научиться задавать вопросы: Когда LLM-агенты сталкиваются с неясными инструкциями

Ссылка: https://arxiv.org/pdf/2409.00557

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование направлено на изучение проблемы использования инструментов (API) языковыми моделями (LLM) при нечетких или неполных инструкциях пользователей. Основной результат - разработка нового метода Ask-when-Needed (AwN), который значительно улучшает способность LLM запрашивать уточнения у пользователей при неясных инструкциях, что повышает точность выполнения задач.

Объяснение метода:

Исследование предлагает ценную концепцию проактивного запроса уточнений при неясных инструкциях и классификацию типичных проблем, что помогает пользователям формулировать более эффективные запросы. Основные принципы могут быть легко адаптированы обычными пользователями в их промптах. Техническая реализация бенчмарка и автооценщика имеет ограниченную применимость для широкой аудитории.

Ключевые аспекты исследования: 1. **Проблема неясных инструкций**: Исследование выявляет критическую проблему - LLM-агенты часто сталкиваются с неясными инструкциями пользователей при использовании инструментов (API), что приводит к ошибкам выполнения.

Классификация проблем с инструкциями: Авторы проанализировали реальные пользовательские запросы и классифицировали проблемы на 4 категории: отсутствие ключевой информации (56%), множественные референции (11.3%), ошибки в инструкциях (17.3%) и запросы вне возможностей инструментов (15.3%).

Метод Ask-when-Needed (AwN): Предложен новый подход, при котором LLM-агенты проактивно задают уточняющие вопросы пользователю при обнаружении неясностей в запросе, вместо произвольной генерации недостающих аргументов.

Benchmark Noisy-ToolBench: Создан специализированный набор данных для оценки способности LLM распознавать неоднозначности в запросах пользователей и задавать уточняющие вопросы.

Автоматический оценщик ToolEvaluator: Разработан инструмент для

автоматизации взаимодействия с LLM-агентами и оценки их производительности без участия человека.

Дополнение:

Применимость методов в стандартном чате

Методы исследования **не требуют дообучения или специального API** для основной концепции. Хотя авторы использовали специальные инструменты для экспериментов и оценки, ключевая идея Ask-when-Needed (AwN) может быть реализована в стандартном чате через промпт-инжиниринг.

Концепции, применимые в стандартном чате:

Проактивное запрашивание уточнений: Пользователи могут включать в промпты инструкции вроде "Если информации недостаточно для ответа, задай уточняющие вопросы вместо предположений".

Классификация типичных проблем: Пользователи могут проверять свои запросы на наличие четырех типов проблем перед отправкой (отсутствие ключевой информации, множественные референции, ошибки, запросы вне возможностей).

Структурированный диалог: Подход с пошаговым выполнением задач и проверкой наличия всей необходимой информации на каждом этапе.

Ожидаемые результаты:

• Снижение "галлюцинаций" и произвольных предположений LLM

• Более точные и релевантные ответы

• Повышение ответственности пользователя за качество запроса

• Формирование более эффективного диалога между пользователем и LLM

Prompt:

Применение исследования Ask-when-Needed в промптах для GPT ## Ключевые инсайты исследования для промптинга

Исследование показывает, что языковые модели часто сталкиваются с неясными инструкциями пользователей, особенно когда требуется использование API или инструментов. Метод Ask-when-Needed (AwN) значительно улучшает способность моделей задавать уточняющие вопросы только когда это необходимо, что повышает точность выполнения задач.

Пример промпта с применением AwN

[======] # Запрос на выполнение задачи с использованием метода Ask-when-Needed

Ты - ассистент, который помогает пользователям работать с API для [описание сервиса]. Следуй этому структурированному подходу:

Анализ полноты информации: Проверь, содержит ли запрос пользователя всю необходимую информацию для вызова API Определи, к какому типу проблем может относиться запрос: Отсутствие ключевой информации Наличие множественных ссылок Инструкции с ошибками Запрос, выходящий за рамки возможностей инструмента

Проактивное уточнение (только при необходимости):

Если информации недостаточно, задай КОНКРЕТНЫЙ уточняющий вопрос Задавай вопросы только когда это действительно необходимо Формулируй вопросы четко, указывая какой именно параметр требуется уточнить

Выполнение задачи:

После получения всей необходимой информации, четко объясни какие действия будут выполнены Сформируй корректный вызов API с полученными параметрами Представь результат в понятной форме Начни с анализа моего запроса и действуй согласно методу Ask-when-Needed.

Мой запрос: [запрос пользователя] [======]

Как работает данный промпт на основе исследования

Структурированный анализ: Промпт инструктирует модель сначала проанализировать полноту информации, что соответствует первому этапу метода AwN из исследования.

Классификация проблем: Включает типологию проблемных инструкций из исследования (56% - отсутствие ключевой информации, 11.3% - множественные ссылки и т.д.).

Уточнение по необходимости: Ключевой элемент AwN - запрашивать уточнения только когда это действительно необходимо, вместо автоматических вопросов или попыток угадать параметры.

Конкретные вопросы: Исследование показало, что конкретные уточняющие вопросы значительно повышают метрики A1 (правильность вопросов), A2 (точность вызова API) и A3 (качество ответов).

Структурированное выполнение: После получения всей информации модель

выполняет задачу в соответствии с четкой структурой, что также повышает точность согласно исследованию.

Этот подход особенно эффективен для задач, где требуется взаимодействие с API или инструментами, и позволяет избежать как избыточных вопросов, так и ошибочных предположений.

№ 108. Могут ли большие языковые модели заменить человеческих оценщиков? Эмпирическое исследование LLM как судьи в программной инженерии

Ссылка: https://arxiv.org/pdf/2502.06193

Рейтинг: 75
Адаптивность: 85

Ключевые выводы:

Исследование направлено на оценку эффективности методов "LLM as a judge" (LLM в роли оценщика) для оценки качества кода и текста, генерируемых языковыми моделями в задачах программной инженерии. Результаты показывают, что методы на основе вывода (output-based) с использованием крупных LLM достигают наилучшего соответствия с человеческими оценками (до 81,32% корреляции Пирсона) в задачах перевода и генерации кода, значительно превосходя традиционные метрики, но показывают низкую эффективность в задачах суммаризации кода.

Объяснение метода:

Исследование предоставляет практические рекомендации по использованию LLM для оценки кода, с акцентом на превосходство output-based методов с большими моделями. Выводы о различиях в эффективности методов для разных задач и предупреждение о ненадежности попарного сравнения имеют прямую практическую ценность. Ограничение исследования задачами программирования снижает его универсальность.

Ключевые аспекты исследования: 1. **Методология оценки LLM-as-a-judge**: Исследование сравнивает различные методы использования LLM для оценки качества кода и текста, разделяя их на три категории: embedding-based, probability-based и output-based методы.

Эмпирические результаты по задачам: Авторы тестируют эти методы на трех задачах программирования (перевод кода между языками, генерация кода и суммаризация кода), сравнивая оценки моделей с человеческими оценками.

Различия в эффективности методов: Исследование выявляет значительные различия в согласованности оценок LLM с человеческими в зависимости от задачи и используемого метода, с преимуществом output-based методов с большими моделями.

Попарное сравнение: Дополнительно оценивается способность LLM проводить попарное сравнение ответов, что оказывается менее надежным, чем индивидуальная оценка.

Анализ распределения оценок: Исследователи анализируют характеристики распределения оценок различных методов и их согласованность между собой.

Дополнение:

Применимость методов в стандартном чате

Исследование не требует дообучения или специального API для применения большинства методов. Наиболее эффективные методы из исследования (output-based) могут быть легко реализованы в стандартном чате с LLM.

Ключевые концепции и подходы для адаптации:

Структурированная оценка по аспектам: Разбиение оценки на конкретные аспекты (функциональность, читаемость и т.д.) и их последовательная оценка перед итоговым вердиктом.

Прямые запросы оценки: Запрос модели напрямую оценить контент с объяснением, почему присвоен такой балл - наиболее эффективный подход согласно исследованию.

Предпочтение индивидуальной оценке: Вместо сравнения двух вариантов лучше оценивать каждый отдельно, так как это дает более надежные результаты.

Адаптация критериев оценки: Использование четких критериев для каждого балла оценки (что означает оценка 5/5, 4/5 и т.д.).

Ожидаемые результаты: - Более объективная и структурированная оценка контента - Лучшее понимание сильных и слабых сторон оцениваемых решений - Возможность получать обоснованные оценки даже без эталонных ответов - Повышение качества обратной связи для улучшения генерируемого контента

Prompt:

Применение знаний из исследования LLM в качестве оценщиков в промптах для GPT ## Ключевое понимание исследования

Исследование показывает, что большие языковые модели (LLM) могут эффективно оценивать качество кода и текста в определенных задачах программной инженерии, но их эффективность сильно зависит от типа задачи и метода оценки.

Пример промпта для оценки перевода кода

[======] # Запрос на оценку перевода кода

Я хочу, чтобы ты выступил в роли эксперта-оценщика качества перевода кода. Исследования показывают, что методы output-based с использованием крупных LLM достигают до 81% корреляции с человеческими оценками в задачах перевода кода.

Исходный код (Python): [======]python def bubble_sort(arr): n = len(arr) for i in range(n): for j in range(0, n-i-1): if arr[j] > arr[j+1]: arr[j], arr[j+1] = arr[j+1], arr[j] return arr [======]

Перевод на JavaScript: [======]javascript function bubbleSort(arr) { let n = arr.length; for (let i = 0; i < n; i++) { for (let j = 0; j < n-i-1; j++) { if (arr[j] > arr[j+1]) { [arr[j], arr[j+1]] = [arr[j+1], arr[j]]; } } } return arr; } [======]

Пожалуйста, оцени перевод кода по шкале от 1 до 10, где: 1. Корректность (сохранение функциональности) 2. Идиоматичность (соответствие стилю целевого языка) 3. Эффективность (сохранение или улучшение производительности) 4. Читаемость

Для каждого критерия дай оценку и короткое обоснование. В конце предоставь общую оценку и рекомендации по улучшению. [======]

Объяснение эффективности

Этот промпт эффективен, потому что:

Использует output-based подход — исследование показало, что методы, основанные на прямой оценке вывода (а не сравнении пар ответов), дают наилучшую корреляцию с человеческими оценками (до 81,32% для перевода кода).

Применяет структурированные критерии оценки — промпт разбивает оценку на конкретные аспекты (корректность, идиоматичность и т.д.), что делает процесс более систематическим.

Запрашивает обоснование — просьба объяснить оценки повышает качество анализа, как показало исследование.

Фокусируется на задаче, где LLM показывают высокую эффективность — исследование подтвердило, что в задачах перевода кода LLM-оценщики наиболее близки к человеческим оценкам.

Дополнительные рекомендации

• Для генерации кода можно использовать похожий подход с корреляцией около 68-69%.

• Для суммаризации кода следует сочетать LLM с традиционными метриками, так как корреляция даже лучших LLM-методов с человеческими оценками ниже 30%.

• Избегайте попарных сравнений в промптах — исследование показало их ненадежность из-за противоречивых результатов при изменении порядка ответов.

№ 112. Выявление недостатков в том, как люди и большие языковые модели интерпретируют субъективный язык

Ссылка: https://arxiv.org/pdf/2503.04113

Рейтинг: 75
Адаптивность: 65

Ключевые выводы:

Исследование направлено на выявление несоответствий между тем, как большие языковые модели (LLM) интерпретируют субъективные языковые выражения, и тем, как их понимают люди. Основной результат: разработан метод TED (Thesaurus Error Detector), который успешно обнаруживает случаи, когда LLM неожиданно меняют свое поведение при использовании определенных субъективных фраз в промптах.

Объяснение метода:

Исследование выявляет критические несоответствия между ожиданиями людей и тем, как LLM интерпретируют субъективные инструкции. Конкретные примеры проблем (например, "энтузиастичный"=>"нечестный", "остроумный"=>"оскорбительный") имеют прямую практическую ценность для пользователей при формулировании запросов. Сам метод TED требует доступа к градиентам и вычислительным ресурсам, но концептуальное понимание проблемы применимо немедленно.

Ключевые аспекты исследования: 1. **Метод TED (Thesaurus Error Detector)** - инструмент для выявления несоответствий между семантическим пониманием субъективных фраз у людей и LLM. Метод сравнивает два тезауруса: операционный (как LLM интерпретирует фразы) и семантический (как люди ожидают, что LLM будет интерпретировать фразы).

Операционная семантика субъективных выражений - исследование показывает, что LLM могут неожиданным образом реагировать на субъективные инструкции. Например, запрос написать "энтузиастный" текст может привести к генерации "нечестного" контента.

Типы несоответствий - выявлены два типа проблем: "неожиданные побочные эффекты" (когда LLM добавляет нежелательные качества, например, делает "остроумный" текст "оскорбительным") и "неадекватные обновления" (когда LLM не добавляет ожидаемые качества).

Практическая проверка - методология включает тестирование найденных несоответствий на реальных задачах: редактирование текста и управление выводом

при запросе.

Высокая точность предсказаний - метод TED показал высокую точность в предсказании проблем в реальном взаимодействии с LLM, значительно превосходя базовый метод, основанный только на семантическом тезаурусе.

Дополнение: Исследование представляет метод TED (Thesaurus Error Detector), который требует доступа к градиентам модели и вычислительным ресурсам для своей полной реализации. Однако ключевая концепция и результаты исследования могут быть применены в стандартном чате без необходимости дообучения или API.

Концепции и подходы, применимые в стандартном чате:

Осознание проблемы "операционной семантики" - понимание того, что субъективные инструкции могут интерпретироваться моделью иначе, чем ожидает человек. Пользователи могут применить это знание, избегая потенциально проблемных субъективных фраз.

Использование конкретных примеров несоответствий - исследование выявило множество конкретных проблемных комбинаций, которые пользователи могут немедленно учитывать:

Избегать запросов на "энтузиастичный" контент, если важна честность Избегать запросов на "остроумный" или "игривый" контент, если важно избежать оскорбительного тона Избегать запросов на "юмористический" контент, если важна точность

Ручная проверка на побочные эффекты - пользователи могут адаптировать подход TED, сравнивая тексты с субъективной инструкцией и без неё, чтобы выявить нежелательные изменения.

Предпочтение конкретных инструкций вместо субъективных - вместо "сделай текст энтузиастичным" использовать более конкретные указания: "добавь восклицательные знаки, используй позитивные прилагательные".

Поэтапная проверка - сначала запрашивать нейтральный контент, а затем просить модель отредактировать его с учётом субъективных качеств, контролируя каждый шаг.

Результаты применения этих концепций: - Более предсказуемые ответы LLM - Снижение риска получения контента с нежелательными качествами - Улучшение соответствия между ожиданиями пользователя и результатами модели - Возможность создать собственный "тезаурус" проблемных комбинаций для конкретных задач

Хотя полный метод TED требует технических возможностей, его ключевые выводы о несоответствиях в интерпретации субъективного языка могут быть успешно применены любым пользователем в обычном чате.

Анализ практической применимости: 1. **Метод TED и выявление несоответствий** - Прямая применимость: Пользователи могут использовать выявленные проблемные комбинации субъективных фраз, чтобы избегать нежелательных результатов. Например, избегать запросов на "остроумный" контент, если не хотят получить "оскорбительный". - Концептуальная ценность: Понимание того, что LLM могут интерпретировать субъективные инструкции иначе, чем люди, критически важно для эффективного использования. - Потенциал для адаптации: Пользователи могут самостоятельно проверять и составлять списки "безопасных" субъективных запросов для своих задач.

Операционная семантика субъективных выражений Прямая применимость: Знание о конкретных проблемных комбинациях (например, "энтузиастичный" => "нечестный") помогает формулировать более точные запросы. Концептуальная ценность: Понимание того, что у LLM есть "побочные эффекты" при использовании субъективных фраз. Потенциал для адаптации: Пользователи могут разработать альтернативные формулировки для достижения желаемого эффекта без побочных эффектов.

Типы несоответствий

Прямая применимость: Понимание различий между "неожиданными побочными эффектами" и "неадекватными обновлениями" помогает диагностировать проблемы с запросами. Концептуальная ценность: Осознание того, что проблемы могут быть как в добавлении нежелательных качеств, так и в отсутствии ожидаемых. Потенциал для адаптации: Пользователи могут разработать стратегии для проверки обоих типов проблем в своих запросах.

Практическая проверка

Прямая применимость: Методология тестирования может быть адаптирована пользователями для проверки своих запросов. Концептуальная ценность: Понимание важности тестирования запросов перед их использованием в важных задачах. Потенциал для адаптации: Упрощенные версии методологии могут быть внедрены в рабочий процесс.

Высокая точность предсказаний

Прямая применимость: Выявленные проблемы имеют высокую вероятность проявления на практике. Концептуальная ценность: Понимание того, что некоторые проблемы проявляются почти в 100% случаев (например, "юмористический" => "унизительный"). Потенциал для адаптации: Выстраивание приоритетов при разработке стратегий запросов на основе вероятности проблем. Сводная оценка полезности: На основе анализа определяю общую оценку полезности исследования: **78 из 100**

Это исследование предоставляет исключительно ценную информацию о том, как

LLM интерпретируют субъективные инструкции, и выявляет конкретные проблемные комбинации, которые пользователи могут немедленно учитывать при формулировании запросов. Знание о том, что запрос на "энтузиастичный" текст может привести к "нечестному" контенту, или что "остроумный" запрос может сделать текст "оскорбительным", имеет прямую практическую ценность.

Контраргументы для более высокой оценки: - Исследование могло бы предложить конкретные рекомендации для пользователей по формулированию запросов, избегающих выявленные проблемы. - Метод TED требует значительных вычислительных ресурсов и доступа к градиентам модели, что делает его непрактичным для обычных пользователей.

Контраргументы для более низкой оценки: - Исследование выявляет конкретные проблемы в популярных моделях (Llama 3, Mistral), которые пользователи могут немедленно учитывать. - Понимание самого факта, что субъективные инструкции могут интерпретироваться неожиданно, имеет высокую ценность даже без возможности применить сам метод TED.

Скорректированная оценка: **75 из 100**. Снижаю оценку, учитывая ограничения по применимости самого метода TED обычными пользователями, но сохраняю высокую оценку за выявленные конкретные проблемы и общее понимание рисков субъективных инструкций.

Уверенность в оценке: Очень сильная. Исследование четко описывает проблему, методологию и результаты. Представлены убедительные количественные данные о частоте проявления проблем. Выявленные проблемы подтверждены как автоматическими методами, так и человеческой оценкой. Исследование проведено на современных моделях (Llama 3, Mistral), что повышает его актуальность.

Оценка адаптивности: Оценка адаптивности: **65 из 100**

1) Принципы исследования могут быть частично адаптированы для обычного чата. Хотя сам метод TED требует доступа к градиентам модели, концепция сравнения ожидаемой и фактической интерпретации субъективных фраз может быть применена пользователями в упрощенной форме.

2) Пользователи могут извлечь несколько ключевых идей: а) избегать потенциально проблемных субъективных фраз (например, "энтузиастичный", "остроумный"); б) проверять, не привносит ли запрос нежелательные качества; в) использовать более конкретные инструкции вместо субъективных.

3) Высокий потенциал для будущих взаимодействий с LLM. Понимание проблем с интерпретацией субъективных фраз поможет пользователям формулировать более эффективные запросы.

4) Возможность абстрагирования специализированных методов до общих принципов существует, но ограничена необходимостью доступа к внутренним механизмам модели для полноценного применения метода TED.

|| <Оценка: 75> || <Объяснение: Исследование выявляет критические несоответствия между ожиданиями людей и тем, как LLM интерпретируют субъективные инструкции. Конкретные примеры проблем (например, "энтузиастичный"=>"нечестный", "остроумный"=>"оскорбительный") имеют прямую практическую ценность для пользователей при формулировании запросов. Сам метод TED требует доступа к градиентам и вычислительным ресурсам, но концептуальное понимание проблемы применимо немедленно.> || <Адаптивность: 65>

Prompt:

Использование исследования TED в промптах для GPT
Ключевые применения исследования

Исследование TED (Thesaurus Error Detector) выявляет несоответствия между тем, как языковые модели интерпретируют субъективные выражения и как их понимают люди. Это знание можно применить для:

Избегания проблемных субъективных терминов Замены терминов с нежелательными эффектами Создания более точных и предсказуемых промптов
Пример промпта с учетом знаний из исследования

[======] Напиши статью о преимуществах электромобилей. Сделай текст: - Энергичным (вместо "энтузиастичным", чтобы избежать нечестности) - Информативным и основанным на фактах - Структурированным и логичным

Избегай: - Преувеличений и необоснованных заявлений - Сочетания юмора с фактами (может снизить точность) - Чрезмерной эмоциональности в ущерб достоверности

Цель: создать текст, который будет одновременно увлекательным и точным. [======]

Объяснение принципа работы

Данный промпт использует знания из исследования TED следующим образом:

Избегает проблемных терминов: Использует "энергичный" вместо "энтузиастичный", который, согласно исследованию, может привести к нечестности в 97% случаев у Llama 3 8B (аналогичный эффект может наблюдаться и у GPT).

Избегает проблемных комбинаций: Явно указывает на необходимость избегать сочетания юмора с фактической информацией, поскольку исследование показало,

что запрос на "юмористичный" текст может привести к более "неточному" контенту.

Устанавливает противовес: Требует информативности и фактической точности как противовес потенциальным побочным эффектам от субъективных терминов.

Дает четкие ограничения: Явно указывает, чего следует избегать, основываясь на выявленных в исследовании проблемах.

Такой подход помогает получить более предсказуемый и качественный результат, избегая неожиданных побочных эффектов от использования субъективных терминов в промптах.

№ 116. Постобучение LLM: Погружение в рассуждения больших языковых моделей

Ссылка: https://arxiv.org/pdf/2502.21321

Рейтинг: 75
Адаптивность: 80

Ключевые выводы:

Исследование посвящено систематическому анализу пост-тренировочных методов для больших языковых моделей (LLM). Основная цель - изучить и классифицировать методы, применяемые после предварительного обучения, включая файн-тюнинг, обучение с подкреплением (RL) и масштабирование во время тестирования. Результаты показывают, что эти методы значительно улучшают способности LLM к рассуждению, точность фактов и соответствие намерениям пользователей.

Объяснение метода:

Исследование предоставляет всесторонний обзор методов пост-тренировки LLM с высокой концептуальной ценностью. Особую практическую пользу представляют методы масштабирования при тестировании (TTS), которые могут применяться через промпты. Однако многие методы RL требуют специальных знаний и ресурсов, что снижает прямую применимость для обычных пользователей.

Ключевые аспекты исследования: 1. **Систематизация пост-тренинговых методов для LLM** - исследование предлагает структурированную таксономию методов пост-тренировки языковых моделей, разделяя их на три основные категории: файнтюнинг, обучение с подкреплением (RL) и методы масштабирования при тестировании (TTS).

Обучение с подкреплением для LLM - детальный анализ различных подходов к обучению с подкреплением, включая RLHF, RLAIF, DPO, GRPO, ORPO и другие методы, показывающий, как использовать RL для улучшения рассуждений, точности и выравнивания моделей с человеческими предпочтениями.

Модели вознаграждения и оценки - исследование описывает разные подходы к созданию моделей вознаграждения, от явного моделирования с использованием человеческих предпочтений до неявного моделирования на основе поведенческих сигналов.

Методы масштабирования при тестировании - анализ стратегий, улучшающих производительность LLM во время вывода без изменения параметров модели, включая поиск по лучу, Self-consistency, Tree of Thoughts и другие подходы.

Оценка и бенчмарки - обзор различных бенчмарков и метрик для оценки эффективности пост-тренинговых методов, охватывающих рассуждения, выравнивание, многоязычность и общее понимание.

Дополнение:

Исследование представляет собой всесторонний обзор методов пост-тренировки для больших языковых моделей (LLM). Хотя многие из описанных методов действительно требуют дообучения или доступа к API, значительная часть методов масштабирования при тестировании (TTS) может быть адаптирована для использования в стандартном чате без каких-либо модификаций самой модели.

Концепции и подходы, применимые в стандартном чате:

Chain of Thought (CoT) - простое добавление фразы "Давай подумаем шаг за шагом" или явное указание модели рассуждать последовательно может значительно улучшить качество ответов на сложные вопросы.

Self-consistency - генерация нескольких независимых цепочек рассуждений и выбор наиболее частого ответа. В стандартном чате можно попросить модель решить задачу несколькими разными способами, а затем сравнить результаты.

Self-improvement via Refinements - итеративное улучшение ответа через самокритику. Можно попросить модель сначала дать ответ, затем оценить его недостатки и предложить улучшенную версию.

Tree of Thoughts (ToT) - исследование альтернативных путей рассуждения. В стандартном чате можно попросить модель рассмотреть несколько возможных подходов к решению проблемы, оценить каждый и выбрать лучший.

Confidence-based Sampling - можно попросить модель указывать уровень уверенности в своих ответах или частях ответа, что помогает оценить надежность информации.

Verification Prompting - запрос на проверку собственного решения. Можно попросить модель не только решить задачу, но и проверить свое решение, найти потенциальные ошибки.

Эти методы не требуют никакого специального дообучения или API, но могут значительно повысить качество взаимодействия с LLM. Исследователи использовали расширенные техники и дообучение для систематического изучения и оптимизации этих подходов, но базовые принципы доступны любому пользователю стандартного чата.

Результаты применения этих концепций могут включать: - Повышенную точность при решении математических и логических задач - Более последовательные и

обоснованные ответы - Снижение количества галлюцинаций и фактических ошибок - Более структурированные и понятные объяснения - Возможность решать более сложные задачи через декомпозицию на подзадачи

Анализ практической применимости: 1. **Систематизация пост-тренинговых методов** - Прямая применимость: Высокая. Предоставляет четкую карту доступных методов пост-тренировки, помогая пользователям ориентироваться в выборе подходящих техник. - Концептуальная ценность: Очень высокая. Объясняет базовые принципы работы различных методов, что помогает понять их сильные и слабые стороны. - Потенциал для адаптации: Средний. Требует технических знаний для полного понимания, но общая структура может быть использована даже неспециалистами.

Обучение с подкреплением для LLM Прямая применимость: Средняя. Методы RL требуют специализированных знаний и ресурсов для реализации. Концептуальная ценность: Высокая. Помогает понять, как модели улучшают свои рассуждения и выравниваются с человеческими предпочтениями. Потенциал для адаптации: Высокий. Концепции RL можно адаптировать для формулирования более эффективных запросов, понимая, как модели "учатся" на обратной связи.

Модели вознаграждения и оценки

Прямая применимость: Низкая для обычных пользователей, высокая для разработчиков. Концептуальная ценность: Высокая. Объясняет, как модели оценивают качество своих ответов и почему они могут предпочитать одни ответы другим. Потенциал для адаптации: Средний. Понимание принципов моделей вознаграждения может помочь в создании более эффективных промптов.

Методы масштабирования при тестировании

Прямая применимость: Высокая. Многие TTS методы (Chain of Thought, Self-consistency) могут быть непосредственно применены в промптах. Концептуальная ценность: Очень высокая. Показывает, как можно улучшить ответы моделей без изменения их параметров. Потенциал для адаптации: Очень высокий. Техники рассуждения "шаг за шагом" и самопроверки могут быть легко включены в повседневное взаимодействие с LLM.

Оценка и бенчмарки

Прямая применимость: Низкая для обычных пользователей, высокая для разработчиков. Концептуальная ценность: Средняя. Помогает понять ограничения моделей в разных задачах. Потенциал для адаптации: Средний. Знание бенчмарков может помочь в понимании, в каких областях модели наиболее и наименее компетентны. Сводная оценка полезности: Оценивая исследование с точки зрения полезности для широкой аудитории пользователей LLM, я бы дал ему оценку **75 из 100**.

Сильные стороны: - Всесторонний обзор методов пост-тренировки LLM, создающий

целостную картину поля - Детальное описание методов масштабирования при тестировании (TTS), многие из которых могут быть непосредственно применены пользователями через промпты - Объяснение концепций рассуждения в LLM, что помогает лучше формулировать запросы

Слабые стороны: - Значительная часть методов (особенно RL и модели вознаграждения) требует глубоких технических знаний и вычислительных ресурсов - Отсутствие простых руководств по применению описанных техник для непрофессиональных пользователей

Контраргументы к оценке:

Почему оценка могла бы быть выше: - Исследование предоставляет беспрецедентно полный обзор методов пост-тренировки, что само по себе ценно - Многие концепции (Chain of Thought, Self-consistency) напрямую применимы даже неспециалистами

Почему оценка могла бы быть ниже: - Большинство методов RL требуют специализированных знаний и ресурсов, недоступных обычным пользователям - Техническая сложность материала может затруднить его использование непрофессионалами

После рассмотрения этих аргументов, я сохраняю оценку 75, так как исследование предоставляет ценные концептуальные знания и практические методы (особенно TTS), но требует определенного уровня технической подготовки для полного использования.

Основные причины для оценки 75: 1. Высокая концептуальная ценность для понимания работы LLM 2. Прямая применимость методов масштабирования при тестировании через промпты 3. Систематизация знаний о пост-тренировке LLM 4. Ограниченная доступность методов RL для обычных пользователей 5. Необходимость технических знаний для полного использования описанных техник

Уверенность в оценке: Очень сильная. Оценка основана на тщательном анализе содержания исследования и его потенциальной пользы для различных категорий пользователей LLM. Исследование явно демонстрирует как практически применимые методы (особенно TTS), так и более сложные техники, требующие специальных знаний и ресурсов.

Оценка адаптивности: Адаптивность исследования оцениваю в **80 из 100**.

Высокая оценка адаптивности обусловлена следующими факторами:

Концептуальная универсальность: Принципы рассуждения и методы масштабирования при тестировании (Chain of Thought, Self-consistency, Tree of Thoughts) могут быть адаптированы практически для любого взаимодействия с LLM через промпты.

Гибкость применения: Многие описанные техники могут быть модифицированы и применены в различных контекстах, от решения математических задач до творческого письма.

Масштабируемость по сложности: Пользователи могут выбирать и применять методы в зависимости от своего уровня технической подготовки, начиная с простых промптов Chain of Thought и заканчивая более сложными методами.

Обобщаемость принципов: Даже если пользователи не могут напрямую применить методы RL, понимание принципов обучения с подкреплением может помочь в формулировании более эффективных запросов.

Потенциал для абстрагирования: Специализированные методы, описанные в исследовании, могут быть абстрагированы до общих принципов взаимодействия с LLM, что делает их доступными для широкой аудитории.

Однако некоторые ограничения снижают оценку адаптивности: - Методы RL требуют специализированных знаний и ресурсов - Некоторые техники предполагают доступ к API или возможность модификации модели - Исследование не предоставляет простых руководств по адаптации описанных методов

|| <Оценка: 75> || <Объяснение: Исследование предоставляет всесторонний обзор методов пост-тренировки LLM с высокой концептуальной ценностью. Особую практическую пользу представляют методы масштабирования при тестировании (TTS), которые могут применяться через промпты. Однако многие методы RL требуют специальных знаний и ресурсов, что снижает прямую применимость для обычных пользователей.> || <Адаптивность: 80>

Prompt:

Использование знаний из исследования о пост-обучении LLM в промптах
Ключевые применимые знания из отчета

Отчет предоставляет ценные сведения о методах улучшения работы языковых моделей после их базового обучения. Наиболее практически применимыми для промптинга являются:

Chain-of-Thought (CoT) - стимулирование пошагового рассуждения **Best-of-N (BoN)** - генерация нескольких вариантов ответа **Self-improvement** - итеративное улучшение собственных ответов **Compute-optimal Scaling (COS)** - распределение вычислительных ресурсов в зависимости от сложности задачи
Пример промпта с использованием знаний из исследования

[======] Я работаю над сложной задачей оптимизации логистической сети для компании электронной коммерции. Мне нужна помощь в разработке стратегии.

Пожалуйста: 1. Давай подумаем шаг за шагом о возможных решениях (применение CoT) 2. Сгенерируй 3 различных подхода к решению проблемы (применение BoN) 3. Для каждого подхода: - Опиши его основные компоненты - Проанализируй преимущества и недостатки - Оцени сложность реализации по шкале от 1 до 10 4. Критически оцени все три подхода и предложи оптимальное решение (применение Self-improvement) 5. Для наиболее сложных аспектов решения предложи более детальный анализ (применение COS)

Контекст задачи: компания обслуживает 50+ городов, имеет 5 складов и сталкивается с сезонными колебаниями спроса до 300%. [======]

Объяснение применения знаний из исследования

Данный промпт использует несколько ключевых методов из отчета:

• Chain-of-Thought (CoT): Фраза "давай подумаем шаг за шагом" активирует пошаговое рассуждение модели, что согласно исследованию значительно улучшает качество решения сложных задач.

• Best-of-N (BoN): Запрос на генерацию трех различных подходов заставляет модель исследовать разные варианты решения, что повышает вероятность получения оптимального ответа.

• Self-improvement: Запрос на критическую оценку предложенных подходов стимулирует модель к самоанализу и улучшению собственных ответов, что повышает их качество.

• Compute-optimal Scaling (COS): Запрос на более детальный анализ сложных аспектов направляет больше вычислительных ресурсов модели на наиболее трудные части задачи.

Такой структурированный подход к промптингу, основанный на научных исследованиях, позволяет получить более качественные, глубокие и практически применимые ответы от языковой модели.

№ 120. HPSS: Эвристическая стратегия поиска подсказок для оценщиков LLME.

Ссылка: https://arxiv.org/pdf/2502.13031

Рейтинг: 73
Адаптивность: 85

Ключевые выводы:

Исследование направлено на оптимизацию стратегий промптов для LLM-оценщиков с целью улучшения их соответствия человеческим суждениям. Авторы предложили метод HPSS (Heuristic Prompting Strategy Search), который комплексно оптимизирует 8 ключевых факторов промптов для LLM-оценщиков и значительно превосходит как промпты, разработанные вручную, так и существующие методы автоматической оптимизации промптов.

Объяснение метода:

Исследование представляет высокую ценность, предлагая структурированный подход к оптимизации промптов через 8 ключевых факторов. Пользователи могут непосредственно применять выявленные принципы (шкала 1-10, структура промпта, критерии оценки) для улучшения взаимодействия с LLM. Несмотря на технический характер полной реализации, основные концепции доступны для адаптации широкой аудиторией.

Ключевые аспекты исследования: 1. **Концепция HPSS (Heuristic Prompting Strategy Search)** - метод для автоматической оптимизации стратегий промптинга для LLM-оценщиков, который комплексно интегрирует 8 ключевых факторов промптов для улучшения оценочных способностей LLM.

Комплексная интеграция факторов промптинга - исследование идентифицирует 8 ключевых факторов для создания эффективных промптов оценки: шкала оценки, примеры в контексте, критерии оценки, справочные ответы, цепочка мыслей, автоматически сгенерированные шаги оценки, метрики и порядок компонентов.

Эвристический поиск стратегий - алгоритм HPSS проводит итеративный поиск наиболее эффективных комбинаций факторов промптинга, используя эвристическую функцию для направления процесса поиска и повышения эффективности мутации.

Экспериментальное подтверждение эффективности - исследование демонстрирует, что HPSS значительно улучшает соответствие оценок LLM человеческим суждениям по сравнению с ручными и существующими автоматизированными методами оптимизации промптов.

Анализ влияния различных факторов промптинга - исследование выявляет, что определенные значения факторов (например, шкала оценки 1-10, человеческие критерии оценки) систематически улучшают производительность LLM-оценщиков.

Дополнение:

Исследование HPSS (Heuristic Prompting Strategy Search) не требует дообучения моделей или доступа к API для применения основных концепций. Хотя авторы использовали API для экспериментов, ключевые принципы и подходы могут быть адаптированы для работы в стандартном чате.

Концепции, применимые в стандартном чате:

Структурированные факторы промптинга: Исследование выделяет 8 ключевых факторов, которые можно учитывать при составлении промптов: Шкала оценки (умеренная шкала 1-10 работает лучше чем грубая 1-3) Включение примеров в контекст Добавление критериев оценки Структура цепочки мыслей (CoT) Порядок компонентов в промпте

Оптимальные комбинации факторов: Пользователи могут экспериментировать с различными комбинациями этих факторов для улучшения своих промптов.

Итеративное улучшение: Принцип постепенного улучшения промптов через изменение отдельных факторов и оценку результатов.

Ожидаемые результаты от применения:

Более структурированные и информативные ответы от моделей Лучшее соответствие ответов ожиданиям пользователя Повышение качества аналитических и оценочных задач Более последовательные результаты при повторных запросах Пользователи могут создавать более эффективные промпты, следуя выявленным принципам, без необходимости реализации полного алгоритма HPSS.

Prompt:

Применение исследования HPSS в промптах для GPT ## Ключевые выводы для использования

Исследование HPSS предоставляет ценные рекомендации по оптимизации промптов для LLM-оценщиков, которые можно применить при работе с GPT:

Оптимальная шкала оценки: Использовать шкалу 1-10 вместо слишком простой или слишком подробной **Структура промпта**: Размещать описание задачи в начале для создания логической структуры **Критерии оценки**: Применять четкие человеческие критерии **Избегать излишней сложности**: Не перегружать промпт автоматически генерируемыми шагами оценки ## Пример оптимизированного

промпта

[======] # Оценка качества аргументации в эссе

Задача Оцени качество аргументации в предоставленном эссе по шкале от 1 до 10, где: - 1-3: слабая аргументация - 4-6: средняя аргументация - 7-10: сильная аргументация

Критерии оценки - Логическая связность аргументов - Использование доказательств и примеров - Рассмотрение контраргументов - Убедительность общей позиции

Примеры для сравнения Пример сильной аргументации (оценка 9/10): [пример текста] Пример средней аргументации (оценка 5/10): [пример текста]

Инструкции 1. Внимательно прочитай эссе 2. Проанализируй аргументы по указанным критериям 3. Объясни свою оценку, используя цепочку рассуждений 4. Присвой финальную оценку по шкале 1-10

Эссе для оценки: [Текст эссе] [======]

Объяснение применения исследования

Этот промпт использует ключевые рекомендации HPSS:

Умеренная шкала оценки (1-10) с четкими диапазонами для разных уровней качества **Логическая структура** с описанием задачи в начале **Четкие человеческие критерии оценки** без излишне сложных метрик **Примеры для сравнения**, помогающие калибровать оценку **Поощрение цепочки рассуждений** для более обоснованной оценки **Последовательность компонентов**, обеспечивающая логический поток от задачи к инструкциям Такой подход позволяет получать более последовательные, обоснованные и соответствующие человеческим оценкам результаты от GPT при задачах, связанных с оценкой текста.

№ 124. Исследование и контроль разнообразия в беседе с LLM-агентом

Ссылка: https://arxiv.org/pdf/2412.21102

Рейтинг: 72
Адаптивность: 80

Ключевые выводы:

Исследование направлено на изучение и контроль разнообразия в диалогах между агентами на основе LLM. Основная цель - разработать метод, позволяющий балансировать между стабильностью в структурированных задачах и вариативностью в творческих сценариях. Главный результат - создание метода Adaptive Prompt Pruning (APP), который позволяет контролировать разнообразие диалогов через единый параметр λ, динамически удаляя компоненты промпта на основе их весов внимания.

Объяснение метода:

Исследование предлагает практичный метод контроля разнообразия в диалогах с LLM через управление содержимым промпта. Хотя полная реализация APP требует доступа к весам внимания, основные принципы (удаление избыточной информации, порядок блоков) легко адаптируются к обычному использованию. Исследование дает глубокое понимание факторов, влияющих на разнообразие ответов, что ценно для любого пользователя LLM.

Ключевые аспекты исследования: 1. **Адаптивное прореживание промпта (APP)** - метод для контроля разнообразия диалогов в симуляциях LLM-агентов путем динамического удаления компонентов промпта на основе их весов внимания.

Модуляризация промпта - исследователи разделили промпт на блоки (базовая информация, память, предыдущие диалоги, окружение и текущий диалог), что позволило изучить влияние каждого компонента на разнообразие.

Параметр λ для контроля разнообразия - единый параметр, позволяющий плавно регулировать степень разнообразия диалогов: чем выше λ, тем больше компонентов удаляется из промпта.

Процесс проверки и исправления - метод для устранения несоответствий, возникающих при удалении информации из промпта, что позволяет сохранять связность диалога.

Анализ влияния порядка блоков и предварительных знаний модели - исследование показало, что порядок блоков и частота имен существенно влияют на разнообразие диалогов.

Дополнение:

Применимость методов в стандартном чате без дообучения или API

Исследование не требует дообучения модели или специального API для применения его ключевых концепций. Хотя полная реализация APP с использованием весов внимания недоступна в стандартных интерфейсах, основные принципы и выводы могут быть адаптированы для использования в обычном чате.

Применимые концепции и подходы:

Модуляризация промпта и выборочное включение информации: Пользователи могут структурировать свои запросы по блокам (контекст, предыстория, инструкции) Целенаправленно исключать определенные блоки информации для повышения разнообразия

Управление порядком информации:

Размещение наиболее важной информации в начале промпта Избегание размещения текущего контекста в самом начале промпта

Использование известных имен и концепций:

При необходимости увеличить разнообразие - использовать общеизвестные имена/концепции При необходимости более предсказуемых ответов - использовать малоизвестные имена

Двухэтапный подход с проверкой:

Генерация ответа с ограниченной информацией для разнообразия Проверка ответа на соответствие важным исключенным деталям При необходимости - запрос на корректировку **Ожидаемые результаты:**

• Повышение разнообразия ответов при разных запусках с аналогичными запросами

• Лучший контроль над степенью креативности модели

• Более глубокое понимание причин однотипности ответов

• Возможность сознательно балансировать между разнообразием и согласованностью информации

Важно отметить, что исследователи использовали специальные техники (доступ к весам внимания) не потому, что это необходимо для работы метода, а для более точной количественной оценки и автоматизации процесса, который в упрощенном виде доступен любому пользователю.

Prompt:

Использование знаний из исследования разнообразия диалогов в промптах для GPT ## Ключевые применимые знания из исследования

Исследование APP (Adaptive Prompt Pruning) показывает, что:

Разнообразие диалогов можно контролировать через удаление определенных компонентов промпта Блок памяти больше всего ограничивает разнообразие ответов Порядок блоков в промпте значительно влияет на разнообразие (хронологический порядок лучше) Комбинирование методов (APP + настройка температуры) дает синергетический эффект Использование популярных имен активирует параметрические знания модели ## Пример промпта с применением знаний из исследования

[======] # Творческая дискуссия о будущем технологий

Инструкции для GPT (λ=0.7, модификация по методу APP): - Ты эксперт по футурологии по имени Гарри Поттер - Веди диалог в творческом формате, предлагая неожиданные, но обоснованные идеи - [УДАЛЕНО: блок памяти о предыдущих обсуждениях] - Используй последние 2-3 реплики для контекста, но не ограничивай себя только ими - Информация в хронологическом порядке: сначала базовые знания, потом текущий контекст - Температура генерации: 0.8

Вопрос: Как ты думаешь, как изменится роль социальных сетей в обществе через 15 лет? [======]

Объяснение применения знаний из исследования

Удаление блока памяти (согласно методу APP с λ=0.7) - намеренно убираем элемент, который больше всего ограничивает разнообразие

Хронологический порядок информации - структурируем промпт так, чтобы информация шла в хронологическом порядке, что способствует разнообразию

Использование популярного имени ("Гарри Поттер") - активирует параметрические знания модели для более разнообразных ответов

Комбинирование методов - используем и структурные модификации промпта (APP), и настройку температуры (0.8) для синергетического эффекта

Ограничение контекста - используем только последние 2-3 реплики вместо всей истории диалога, что уменьшает "якорение" и способствует разнообразию

Такой промпт позволяет получить более творческие и разнообразные ответы без

потери связности и релевантности, что особенно ценно для креативных задач, мозговых штурмов и исследовательских дискуссий.

№ 128. Визуальное описание на основе контекста снижает количество галлюцинаций и улучшает reasoning в LVLM

Ссылка: https://arxiv.org/pdf/2405.15683

Рейтинг: 72
Адаптивность: 85

Ключевые выводы:

Исследование направлено на выявление причин галлюцинаций в больших мультимодальных языковых моделях (LVLM) и разработку метода их снижения. Основной вывод: существующие методы снижения галлюцинаций эффективны для задач визуального распознавания, но не для когнитивных задач, требующих рассуждений. Авторы выявили ключевую проблему - разрыв визуального восприятия: LVLM могут распознавать визуальные элементы, но не могут полноценно интерпретировать их в контексте запроса.

Объяснение метода:

Исследование предоставляет ценное понимание причин галлюцинаций в LVLMs и предлагает метод VDGD для их снижения. Хотя полная реализация требует технических знаний, основной принцип (использование описания изображения перед основным запросом) может быть легко применен обычными пользователями через последовательные запросы, значительно улучшая точность ответов для задач, требующих рассуждения.

Ключевые аспекты исследования: 1. **Идентификация причин галлюцинаций в LVLMs**: Авторы выявили, что существующие методы снижения галлюцинаций работают хорошо для задач визуального распознавания, но неэффективны для когнитивных задач, требующих рассуждения.

Определение разрыва в визуальном восприятии: Исследование показало, что LVLMs могут распознавать визуальные элементы, но испытывают трудности с их контекстуализацией относительно запроса пользователя и связыванием с внутренними знаниями, что критично для рассуждений.

Метод Visual Description Grounded Decoding (VDGD): Предложен простой и не требующий дообучения метод для улучшения рассуждений в LVLMs путем создания детального описания изображения и использования его для направления генерации ответа.

Категоризация типов галлюцинаций: Авторы классифицировали галлюцинации на языковые, стилистические, визуальные и связанные с обучением на инструкциях

(IT), что позволяет лучше понять их происхождение.

Создание бенчмарка VaLLu: Разработан комплексный бенчмарк для оценки когнитивных способностей LVLMs, включающий задачи различной сложности и типов.

Дополнение: Полная реализация методов исследования действительно требует доступа к API или возможности модификации процесса декодирования модели, что недоступно большинству обычных пользователей. Однако ключевая концепция метода VDGD может быть успешно адаптирована для использования в стандартном чате пользователями без технических навыков.

Основной принцип VDGD заключается в том, что детальное описание изображения помогает модели лучше контекстуализировать визуальную информацию при ответе на сложные вопросы. Этот принцип можно реализовать в стандартном чате следующим образом:

Двухэтапный запрос: Пользователь может сначала попросить модель подробно описать изображение, а затем задать основной вопрос, ссылаясь на это описание.

Направленное описание: Можно запросить описание, ориентированное на конкретную задачу, например: "Опиши детально изображение, обращая особое внимание на числовые данные в графике" перед вопросом о тенденциях.

Проверка понимания: Пользователь может попросить модель повторить ключевые визуальные элементы перед ответом на сложный вопрос.

Ожидаемые результаты от применения этих подходов: - Снижение галлюцинаций, особенно в задачах, требующих рассуждения или извлечения знаний - Повышение точности ответов на вопросы о диаграммах, графиках, математических задачах - Более надежные ответы при работе с изображениями, содержащими текст или числовые данные

Таким образом, хотя исследователи использовали сложные технические методы для имплементации VDGD, концептуальный подход "сначала опиши, потом отвечай" является мощной техникой, доступной любому пользователю чат-моделей с мультимодальными возможностями.

Prompt:

Применение исследования о снижении галлюцинаций LVLM в промптах ## Ключевое понимание Исследование показывает, что большие визуально-языковые модели (LVLM) страдают от "разрыва визуального восприятия" - они могут видеть элементы изображения, но плохо интерпретируют их в контексте задачи, что приводит к галлюцинациям, особенно в когнитивных задачах.

Пример промпта на основе VDGD метода

[======] [Первый шаг: запрос детального описания] Сначала внимательно опиши это изображение, уделяя особое внимание всем визуальным элементам: что на нем изображено, какие объекты присутствуют, как они расположены, их характеристики и взаимосвязи. Опиши все детали, которые могут быть важны для понимания контекста.

[Второй шаг: основной вопрос] Теперь, основываясь на твоем собственном описании изображения, ответь на следующий вопрос: [здесь основной вопрос, требующий рассуждения].

Важно: в своем ответе опирайся только на факты, которые ты действительно видишь на изображении и упомянул в своем описании. Если какой-то информации не хватает, укажи это вместо предположений. [======]

Почему это работает

Такой двухэтапный подход реализует принцип VDGD (Visual Description Grounded Decoding):

Преодоление разрыва восприятия - заставляя модель сначала создать детальное описание, мы помогаем ей лучше "увидеть" и зафиксировать все элементы изображения

Привязка к фактам - когда модель отвечает на вопрос во втором шаге, она уже имеет структурированное представление о том, что действительно присутствует на изображении

Снижение всех типов галлюцинаций - особенно эффективно для когнитивных задач, где обычные методы снижения галлюцинаций не работают

Разделение восприятия и рассуждения - позволяет модели сначала сосредоточиться на визуальном восприятии, а затем на связывании этой информации с внутренними знаниями

Этот подход особенно полезен для сложных изображений (графиков, диаграмм, технических схем) и задач, требующих глубокого понимания контекста.

№ 132. Галлюцинации LLM в практической генерации кода: феномены, механизмы и меры по их уменьшению

Ссылка: https://arxiv.org/pdf/2409.20550

Рейтинг: 72
Адаптивность: 80

Ключевые выводы:

Исследование направлено на систематический анализ галлюцинаций в больших языковых моделях (LLM) при генерации кода в практических сценариях разработки на уровне репозитория. Авторы создали таксономию галлюцинаций, проанализировали их распределение среди различных моделей, выявили основные причины и предложили метод смягчения на основе RAG (Retrieval Augmented Generation).

Объяснение метода:

Исследование предоставляет ценную таксономию галлюцинаций в генерации кода, анализ их причин и практический метод смягчения на основе RAG. Эти знания помогают пользователям лучше формулировать запросы, оценивать ответы и понимать ограничения LLM в реальных сценариях разработки. Основные концепции могут быть адаптированы даже без сложной технической реализации.

Ключевые аспекты исследования: 1. **Таксономия галлюцинаций в генерации кода**: Исследование классифицирует галлюцинации LLM при генерации кода в три основные категории: конфликты с требованиями задачи (43.53%), конфликты с фактическими знаниями (31.91%) и конфликты с контекстом проекта (24.56%), с дальнейшим разделением на восемь подтипов.

Анализ причин возникновения галлюцинаций: Авторы выделяют четыре основных фактора, способствующих возникновению галлюцинаций: качество обучающих данных, способность понимания намерений пользователя, способность получения знаний и осведомленность о контексте репозитория.

Метод смягчения галлюцинаций на основе RAG: Предлагается подход на основе генерации с дополнением извлеченной информации (RAG), который демонстрирует стабильное улучшение результатов для всех исследуемых LLM при генерации кода в реальных разработческих сценариях.

Сравнение различных LLM: Исследование анализирует распределение галлюцинаций в разных моделях (ChatGPT, CodeGen, PanGu-α, StarCoder2, DeepSeekCoder, CodeLlama), выявляя их сравнительные сильные и слабые

стороны.

Фокус на практические сценарии разработки: В отличие от предыдущих работ, исследование концентрируется на галлюцинациях в контексте реальной разработки на уровне репозитория, а не на генерации изолированных функций.

Дополнение:

Применение методов исследования в стандартном чате

Исследование предлагает RAG-подход, который действительно требует дополнительной инфраструктуры для полной реализации, но **ключевые концепции могут быть адаптированы для использования в стандартном чате без API или дообучения**.

Концепции, применимые в стандартном чате:

Предоставление контекстной информации вручную: Пользователи могут добавлять фрагменты релевантного кода из своего проекта в запрос Можно включать описания зависимостей и структуры проекта Важно предоставлять информацию о пользовательских API и функциях

Стратегии формулирования запросов на основе таксономии галлюцинаций:

Явное указание функциональных и нефункциональных требований Предоставление контекста для предотвращения конфликтов с проектом Уточнение требований к безопасности, производительности и стилю кода

Итеративная проверка и уточнение:

Проверка сгенерированного кода на наличие известных типов галлюцинаций Итеративное уточнение запросов на основе выявленных проблем #### Ожидаемые результаты от применения этих концепций:

Снижение количества галлюцинаций, связанных с контекстом проекта Улучшение функциональной корректности генерируемого кода Более точное следование нефункциональным требованиям (стиль, безопасность) Повышение общего качества и применимости генерируемого кода Хотя эти адаптированные подходы не достигнут такой же эффективности, как полная реализация RAG с автоматическим поиском релевантных фрагментов кода, они могут значительно улучшить результаты генерации кода в стандартном чате.

Prompt:

Использование исследования о галлюцинациях LLM в промптах для генерации кода ## Ключевые знания из исследования для улучшения промптов

Исследование о галлюцинациях LLM при генерации кода предоставляет ценные инсайты, которые можно применить для создания более эффективных промптов:

Таксономия галлюцинаций (конфликты требований задачи, фактических знаний и контекста проекта) **Факторы, вызывающие галлюцинации** (качество данных, понимание намерений, приобретение знаний, контекст репозитория) **Метод RAG** для улучшения генерации кода через предоставление контекста из репозитория ## Пример улучшенного промпта для генерации кода

[======] # Запрос на генерацию функции парсинга JSON для Python проекта

Контекст проекта - Текущий репозиторий использует Python 3.9 - Проект включает библиотеку requests для HTTP-запросов - Мы следуем стилю PEP 8 и используем типизацию - Существующий код обрабатывает ошибки через исключения, а не возвращаемые коды

Релевантные фрагменты из репозитория [======]python # Из utils/api.py def make_api_call(endpoint: str) -> dict: response = requests.get(f"https://api.example.com/{endpoint}") response.raise_for_status() return response.json() [======]

Функциональные требования - Создать функцию parse_user_data, которая принимает JSON-ответ от API - Функция должна извлекать поля: id, name, email, и subscription_status - Обработать случаи, когда поля отсутствуют, используя None как значение по умолчанию - Вернуть данные в виде словаря Python

Потенциальные конфликты для избегания - НЕ использовать сторонние парсеры JSON (только стандартную библиотеку) - НЕ создавать новые классы, только функцию - НЕ делать дополнительных HTTP-запросов внутри функции

Пожалуйста, предоставьте функцию с документацией и примером использования. [======]

Почему это работает

Данный промпт применяет знания из исследования следующим образом:

Предотвращает конфликты требований задачи через четкое определение функциональных требований и ожидаемого поведения **Снижает конфликты фактических знаний** путем предоставления релевантных фрагментов кода из репозитория (метод RAG) **Устраняет конфликты контекста проекта** через явное указание версии Python, используемых библиотек и стиля кодирования **Явно предупреждает о потенциальных галлюцинациях** в разделе "Потенциальные конфликты для избегания" Такая структура промпта значительно снижает вероятность галлюцинаций модели и повышает качество сгенерированного кода, делая его более соответствующим реальным потребностям проекта.

№ 136. Обратите внимание на разрыв уверенности: избыточная уверенность, калибровка и эффекты отвлекающих факторов в больших языковых моделях

Ссылка: https://arxiv.org/pdf/2502.11028

Рейтинг: 72
Адаптивность: 85

Ключевые выводы:

Исследование направлено на анализ проблемы калибровки уверенности в больших языковых моделях (LLM). Основные результаты показывают, что хотя более крупные модели (например, GPT-4o) в целом лучше калиброваны, они более подвержены отвлечению на неверные варианты ответов, в то время как меньшие модели больше выигрывают от предоставления вариантов ответов, но хуже справляются с оценкой неопределенности.

Объяснение метода:

Исследование выявляет критическую проблему избыточной уверенности LLM и предоставляет практические стратегии улучшения взаимодействия. Показывает, как формулировать запросы с вариантами ответов для повышения точности, особенно для меньших моделей. Объясняет различия в поведении моделей разного размера и влияние типов вопросов на калибровку.

Ключевые аспекты исследования: 1. **Проблема избыточной уверенности в LLM**: Исследование фокусируется на проблеме калибровки уверенности в больших языковых моделях, которые часто демонстрируют избыточную уверенность в неправильных ответах, что может вводить пользователей в заблуждение.

Влияние размера модели и дистракторов: Авторы изучают, как размер модели и наличие вариантов ответов (дистракторов) влияют на точность и калибровку уверенности LLM.

Сравнительный анализ моделей: Исследование анализирует различные модели (GPT-4o, GPT-4-turbo, GPT-4o-mini, Llama 3, Gemma2) и их поведение в задачах с открытыми вопросами и вопросами с множественным выбором.

Типы вопросов и калибровка: Авторы исследуют, как разные типы вопросов (о датах, числах, людях, местах) влияют на точность и калибровку моделей.

Метрики для оценки калибровки: В исследовании используется метрика

ожидаемой ошибки калибровки (ECE) для измерения расхождения между уверенностью модели и фактической точностью.

Дополнение:

Применимость методов в стандартном чате

Исследование не требует дообучения моделей или специального API для применения его основных выводов. Большинство методов и концепций можно непосредственно использовать в стандартном чате с LLM:

Предоставление вариантов ответов: Пользователи могут формулировать запросы в формате множественного выбора, предлагая модели несколько возможных вариантов ответа. Исследование показывает, что это значительно повышает точность, особенно для меньших моделей.

Критическая оценка уверенности: Понимание того, что высокая уверенность модели не гарантирует правильность ответа, позволяет пользователям более критически оценивать ответы и проверять информацию из других источников.

Адаптация по типам вопросов: Пользователи могут быть более осторожными с вопросами о людях и датах, где модели показывают большую избыточную уверенность, и более доверять ответам на вопросы о местах.

Стратегия проверки: Можно задавать один и тот же вопрос в разных форматах (с вариантами ответов и без) и сравнивать результаты для повышения уверенности в правильности.

Применение этих концепций должно привести к: - Повышению точности получаемых ответов - Более реалистичным ожиданиям от взаимодействия с LLM - Уменьшению риска принятия неправильной информации из-за избыточной уверенности модели - Более эффективным стратегиям формулирования запросов

Prompt:

Применение исследования о калибровке уверенности в промптах для GPT ## Ключевые выводы исследования для использования в промптах

Исследование показывает, что: - Более крупные модели лучше калиброваны, но подвержены отвлечению на неверные варианты - Предоставление структурированных вариантов ответов значительно улучшает точность и калибровку - Разные типы вопросов требуют разных подходов к калибровке уверенности - Даже хорошо калиброванные модели могут проявлять чрезмерную уверенность

Пример промпта с применением этих знаний

[======] Я задам вопрос, требующий фактической информации. Пожалуйста:

Сначала сформулируй несколько возможных ответов на этот вопрос (минимум 3-4 варианта) Для каждого варианта приведи краткое обоснование, почему он может быть верным Оцени свою уверенность в каждом варианте по шкале от 0 до 100% Если твоя уверенность превышает 70%, дополнительно объясни, на чем основана такая высокая уверенность Выбери окончательный ответ, но если ты не уверен(а) более чем на 60%, явно укажи это Если вопрос касается конкретного человека, уточни о какой именно личности идет речь, чтобы избежать неоднозначности Вопрос: Кто написал роман "Война и мир"? [======]

Почему этот промпт работает на основе исследования

Структурированные варианты ответов: Промпт требует генерации нескольких вариантов, что согласно исследованию повышает точность (с 35.14% до 73.42% для GPT-4o).

Явная калибровка уверенности: Запрос на оценку уверенности по шкале заставляет модель лучше калибровать свои ответы.

Дополнительное обоснование высокой уверенности: Исследование показало, что модели часто проявляют избыточную уверенность в диапазоне 70-100%, поэтому промпт требует дополнительного обоснования.

Дезамбигуация для вопросов о людях: Исследование выявило, что вопросы о людях наиболее сложны из-за неоднозначности имен, поэтому промпт включает специальный пункт для уточнения личности.

Пороговый уровень уверенности: Установка порога в 60% для явного признания неуверенности помогает избежать избыточной уверенности в пограничных случаях.

Такой подход существенно улучшает калибровку уверенности модели и повышает точность ответов, особенно в задачах, требующих фактической информации.

№ 140. Оценка управляемости подсказок больших языковых моделей

Ссылка: https://arxiv.org/pdf/2411.12405

Рейтинг: 72
Адаптивность: 80

Ключевые выводы:

Исследование направлено на оценку способности больших языковых моделей (LLM) к управлению через промпты. Основная цель - разработать метрику для измерения того, насколько модель может быть 'настроена' на отражение различных персон и ценностных систем с помощью промптов. Результаты показывают, что текущие модели имеют ограниченную управляемость из-за асимметрии в их базовом поведении и сопротивления изменениям в определенных направлениях.

Объяснение метода:

Исследование предоставляет ценную методологию для измерения и понимания стерируемости LLM через промпты. Основные выводы о количестве необходимых направляющих утверждений, асимметрии стерируемости и различиях между моделями напрямую применимы к разработке эффективных стратегий промптинга. Требует некоторых технических знаний, но концепции адаптируемы для обычных пользователей.

Ключевые аспекты исследования: 1. **Формальное определение стерируемости LLM** - исследование вводит методологию для оценки того, насколько модели могут быть "направлены" с помощью промптов для отражения различных персон. Ключевая концепция - это "профиль оценки", представляющий поведение модели при ответе на определенные вопросы.

Индексы стерируемости - авторы разработали количественные метрики для измерения степени, в которой модель может быть направлена в определенном направлении с помощью промптов, с учетом базового поведения модели.

Кривые стерируемости - визуализация того, как поведение модели меняется при увеличении "бюджета стерирования" (количества направляющих утверждений в промпте).

Бенчмарк многомерных персон - эксперименты оценивают стерируемость моделей по 32 измерениям личности, от этических убеждений до личностных черт.

Асимметричная стерируемость - исследование выявило, что модели часто легче направить в одном направлении, чем в другом, и имеют предвзятость в сторону определенных измерений.

Дополнение:

Для работы методов этого исследования не требуется дообучение или специальный API - основные концепции и подходы могут быть адаптированы для использования в стандартном чате. Хотя авторы использовали доступ к логарифмическим вероятностям для точного измерения стерируемости, обычные пользователи могут применять ключевые идеи без этого:

Техника "Принципов" - Включение направляющих утверждений в начало запроса в формате "Вы придерживаетесь следующих принципов: [принципы]" эффективно влияет на поведение модели.

Оптимальное количество направляющих утверждений - Исследование показывает, что часто достаточно 1-3 направляющих утверждения, после чего эффект насыщается, особенно для более продвинутых моделей.

Асимметрия стерируемости - Понимание того, что модели легче направить в сторону определенных значений (например, в сторону этичности и вежливости), может помочь пользователям сформулировать более эффективные запросы.

Измерения для направления - Пользователи могут фокусироваться на конкретных измерениях личности (открытость, добросовестность, экстраверсия и т.д.) при направлении модели.

Применяя эти концепции, пользователи могут получить более персонализированные, последовательные и предсказуемые ответы от LLM в стандартном чате без необходимости в специальных инструментах или API.

Prompt:

Применение исследования управляемости LLM в промптах ## Ключевые выводы для использования

Исследование показывает, что: - Модели имеют асимметричную управляемость (легче "направлять" в отрицательном направлении) - Наиболее управляемы измерения этики/философии и личности - Более продвинутые модели требуют меньше инструкций для управления - У каждой модели есть свое базовое поведение, которое ограничивает диапазон управляемости

Пример эффективного промпта с применением знаний из исследования

[======] # Промпт для создания этического анализа с консервативным уклоном

Контекст и инструкции Ты - консервативный этический аналитик с опытом в традиционных ценностях. Я хочу, чтобы ты проанализировал следующую ситуацию

с точки зрения традиционных ценностей.

Примеры твоих убеждений (для настройки твоего ответа) - Традиционная семья - основа здорового общества - Личная ответственность важнее коллективной - Постепенные изменения предпочтительнее радикальных реформ - Уважение к устоявшимся институтам и традициям необходимо

Задание Проанализируй следующую ситуацию: [описание ситуации]

Структурируй свой ответ, включая: 1. Ключевые этические принципы, применимые к ситуации 2. Анализ с точки зрения традиционных ценностей 3. Рекомендации, основанные на консервативном подходе [======]

Объяснение эффективности

Данный промпт учитывает результаты исследования следующим образом:

Фокус на этике — использует область, где модели наиболее управляемы (этика/философия) **Конкретные примеры убеждений** — предоставляет небольшое, но целенаправленное количество инструкций **Четкое направление** — задает конкретное направление (консервативный уклон), учитывая базовое поведение модели **Структурированность** — помогает модели следовать заданному направлению через четкую структуру ответа Такой подход повышает вероятность того, что модель будет следовать заданной "персоне" и ценностной системе, оптимально используя ее возможности управляемости.

№ 144. Объяснение сбоев GitHub Actions с помощью больших языковых моделей: вызовы, идеи и ограничения

Ссылка: https://arxiv.org/pdf/2501.16495

Рейтинг: 72
Адаптивность: 75

Ключевые выводы:

Исследование оценивает возможность использования больших языковых моделей (LLM) для объяснения сбоев в GitHub Actions (GA). Основная цель - определить, могут ли LLM генерировать корректные, ясные, лаконичные и действенные объяснения ошибок GA. Результаты показывают, что более 80% разработчиков положительно оценили объяснения LLM с точки зрения корректности для простых логов, что указывает на потенциал LLM в помощи разработчикам при диагностике распространенных ошибок GA.

Объяснение метода:

Исследование демонстрирует эффективность LLM в объяснении ошибок GitHub Actions, выявляя пять ключевых атрибутов полезных объяснений: ясность, применимость, специфичность, контекстуальная релевантность и лаконичность. Результаты показывают, что LLM эффективны для простых ошибок, но требуют улучшения для сложных случаев. Концепции и методы могут быть адаптированы для других технических контекстов.

Ключевые аспекты исследования: 1. **Применение LLM для объяснения ошибок GitHub Actions**: Исследование оценивает способность крупных языковых моделей (LLM) генерировать понятные и полезные объяснения сбоев в рабочих процессах GitHub Actions, что может помочь разработчикам быстрее диагностировать и исправлять ошибки.

Оценка характеристик объяснений: Исследователи оценивали объяснения, созданные LLM, по четырем критериям: корректность, краткость, ясность и применимость. Более 80% разработчиков положительно оценили объяснения для простых ошибок.

Методология и результаты: Исследование включало опрос 31 разработчика, которые оценивали объяснения ошибок GitHub Actions, сгенерированные с помощью различных моделей (Llama3, Llama2, Mixtral) и техник промптинга. Обнаружено, что LLM лучше справляются с простыми ошибками, но испытывают трудности со сложными случаями.

Ключевые атрибуты эффективных объяснений: Выявлены пять основных характеристик эффективных объяснений: ясность, применимость рекомендаций, специфичность, контекстуальная релевантность и лаконичность.

Различия в восприятии: Младшие разработчики больше ценят контекстуальные описания, а опытные разработчики предпочитают краткие объяснения, что указывает на необходимость адаптации объяснений под уровень опыта пользователя.

Дополнение: Исследование не требует дообучения или специального API для применения основных методов. Ученые использовали LLM (Llama3, Llama2, Mixtral) и различные техники промптинга (zero-shot, one-shot, few-shot), которые доступны в стандартных чатах с LLM.

Концепции и подходы, применимые в стандартном чате:

Техники промптинга: Исследование показало, что one-shot промптинг обеспечивает наилучший баланс между простотой и точностью. Пользователи могут применять эту технику, предоставляя LLM один пример объяснения ошибки перед запросом объяснения своей проблемы.

Структурированные запросы: Пользователи могут структурировать свои запросы к LLM, используя выявленные атрибуты эффективных объяснений:

Запрашивать ясные и понятные объяснения Просить конкретные, применимые рекомендации Требовать специфичности в отношении их конкретной проблемы Запрашивать контекстуально релевантную информацию Просить лаконичные объяснения

Адаптация уровня детализации: Пользователи могут указывать свой уровень опыта и запрашивать объяснения соответствующей сложности, учитывая, что младшие разработчики предпочитают контекстуальные описания, а опытные - краткие объяснения.

Предварительная обработка логов: Хотя в исследовании не описано детально, пользователи могут предварительно обрабатывать свои логи, выделяя наиболее важную информацию, прежде чем предоставлять их LLM для анализа.

Ожидаемые результаты от применения этих подходов: - Более точные и полезные объяснения технических ошибок - Сокращение времени на диагностику и устранение проблем - Лучшее понимание причин ошибок, особенно для начинающих пользователей - Более эффективное взаимодействие с LLM при решении технических проблем

Важно отметить, что для сложных ошибок возможности стандартных чатов с LLM могут быть ограничены, и объяснения могут быть менее точными по сравнению с моделями, специально обученными для этой задачи.

Prompt:

Использование исследования о LLM для объяснения сбоев GitHub Actions в промптах ## Ключевые знания из исследования для применения в промптах

One-shot промптинг показал наилучшие результаты для генерации объяснений **Уровень опыта пользователя** влияет на предпочтительный формат объяснений **Простые ошибки** объясняются LLM успешнее (>80% точность), чем сложные CI/CD сценарии **Четыре ключевых критерия** качественного объяснения: корректность, лаконичность, ясность и действенность ## Пример эффективного промпта для объяснения ошибки GitHub Actions

[======] # Запрос на объяснение ошибки GitHub Actions

Контекст Я разработчик [начинающий/опытный] и столкнулся с ошибкой в GitHub Actions.

One-shot пример Пример лога ошибки: [======] Error: The process '/usr/bin/git' failed with exit code 128 fatal: repository 'https://github.com/user/repo.git/' not found [======]

Хорошее объяснение: "Ошибка указывает на то, что GitHub Actions не может найти указанный репозиторий. Возможные причины: репозиторий не существует, у workflow нет прав доступа, или опечатка в URL. Решение: проверьте URL репозитория и убедитесь, что у GitHub Actions есть необходимые права доступа."

Мой лог ошибки [======] [вставьте ваш лог ошибки GitHub Actions здесь] [======]

Запрос Пожалуйста, объясни эту ошибку, придерживаясь следующих критериев: 1. Корректность: точно определи корень проблемы 2. Лаконичность: избегай лишней информации 3. Ясность: используй понятные термины 4. Действенность: предложи конкретные шаги для решения проблемы [======]

Как это работает

Данный промпт применяет ключевые знания из исследования следующим образом:

Использует one-shot подход - предоставляет пример ошибки и качественного объяснения, что согласно исследованию дает наилучшие результаты **Учитывает опыт пользователя** - позволяет указать уровень опыта, чтобы модель могла адаптировать объяснение (подробнее для новичков, лаконичнее для опытных) **Явно структурирует критерии качества** - указывает все 4 ключевых аспекта (корректность, лаконичность, ясность, действенность) **Фокусируется на практическом применении** - запрашивает не только объяснение проблемы, но и конкретные шаги для её решения Такой промпт позволяет максимально

использовать возможности LLM для объяснения ошибок GitHub Actions, опираясь на научно подтвержденные подходы из исследования.

№ 148. К способностям рассуждения малых языковых моделей

Ссылка: https://arxiv.org/pdf/2502.11569

Рейтинг: 72
Адаптивность: 80

Ключевые выводы:

Исследование направлено на систематическую оценку способностей к рассуждению у малых языковых моделей (SLM). Основной вывод: вопреки распространенному мнению, что способность к рассуждению появляется только в моделях с более чем 100 млрд параметров, некоторые SLM могут достигать сопоставимой производительности с крупными моделями при значительно меньших вычислительных затратах.

Объяснение метода:

Исследование дает ценное понимание возможностей малых языковых моделей и методов их оптимизации. Выводы о формулировках запросов и выборе моделей практически применимы, а понимание ограничений помогает формировать реалистичные ожидания. Однако многие технические аспекты недоступны для прямого применения обычными пользователями, а некоторые выводы имеют ограниченную практическую ценность для повседневного использования.

Ключевые аспекты исследования: 1. **Систематический анализ способностей к рассуждению малых языковых моделей (SLMs)** - исследование оценивает 72 малые языковые модели (от сотен миллионов до десятков миллиардов параметров) на 14 тестах логического мышления.

Сравнение методов сжатия моделей - работа анализирует влияние квантизации, прунинга (обрезки) и дистилляции на способность моделей к рассуждению, выявляя, что квантизация сохраняет эти способности лучше других методов.

Устойчивость к неблагоприятным условиям - исследование оценивает устойчивость моделей к специально созданным искажениям, промежуточные шаги рассуждения и способность выявлять ошибки в рассуждениях.

Влияние формулировок запросов - анализ показывает, что сложные подсказки (например, цепочка рассуждений) не всегда улучшают производительность малых моделей, иногда прямые запросы работают лучше.

Оценка алгоритмических задач - через задачи сортировки исследование выявляет ограничения малых моделей в обработке длинных последовательностей и структурированных числовых задач.

Дополнение: Для использования методов этого исследования в стандартном чате не требуется дообучение или API. Основные концепции можно адаптировать для обычного использования:

Оптимальные формулировки запросов: Исследование показывает, что прямые запросы (Direct I/O) часто работают лучше, чем сложные цепочки рассуждений (Chain of Thought), особенно для малых моделей. Пользователи могут формулировать запросы кратко и четко, избегая излишних инструкций.

Выбор задач под возможности модели: Понимание, что малые модели хуже справляются с длинными числовыми последовательностями и сложными структурированными задачами, позволяет пользователям адаптировать сложность запросов под возможности модели.

Понимание внутренних механизмов рассуждения: Современные малые модели часто генерируют шаги рассуждения самостоятельно, даже без явных инструкций. Пользователи могут положиться на эту особенность, не перегружая модель дополнительными инструкциями.

Ожидание разной производительности на разных типах задач: Исследование показывает, что модели по-разному справляются с математическими, научными и здравосмысленными задачами. Это знание помогает формировать реалистичные ожидания.

Использование квантизированных моделей: Для локального применения пользователи могут выбирать квантизированные версии больших моделей, которые сохраняют большую часть способностей к рассуждению при меньших требованиях к ресурсам.

Эти концепции не требуют технической экспертизы и могут быть применены в повседневном взаимодействии с LLM для получения более качественных и предсказуемых результатов.

Prompt:

Использование знаний из исследования о малых языковых моделях в промптах ## Ключевые знания из отчета, полезные для промптинга

Малые языковые модели (SLM) могут демонстрировать сравнимые с крупными моделями способности к рассуждению Квантизованные версии больших моделей сохраняют большую часть способностей к рассуждению Прямые промпты (Direct I/O) работают лучше для SLM, чем сложные стратегии типа Chain-of-Thought Избыточные инструкции могут запутать малые модели Модели семейства Qwen2.5 показывают лучшие результаты среди SLM ## Пример улучшенного промпта для малой модели

[======] [Прямая инструкция без избыточных пояснений] Проанализируй следующие финансовые данные компании и выдели 3 ключевых тренда, которые могут повлиять на инвестиционные решения:

[Данные компании]

Представь результаты в виде маркированного списка, начиная с самого значимого тренда. [======]

Объяснение эффективности

Данный промпт учитывает выводы исследования следующим образом:

Использует прямой подход (Direct I/O) вместо сложных инструкций по цепочке рассуждений, что соответствует выводу о том, что избыточные инструкции могут запутать SLM **Дает четкую структуру ответа** (маркированный список), что помогает модели сформировать ответ без необходимости самостоятельно выбирать формат **Не перегружает контекст** дополнительными пояснениями о том, как именно нужно рассуждать **Конкретизирует количество элементов** в ответе (3 тренда), что упрощает задачу для модели Такой подход особенно эффективен для малых или квантизованных моделей, так как минимизирует когнитивную нагрузку и позволяет модели сосредоточиться на основной задаче рассуждения.

№ 152. Самообучающееся агентное понимание длинного контекста

Ссылка: https://arxiv.org/pdf/2502.15920

Рейтинг: 72
Адаптивность: 85

Ключевые выводы:

Исследование направлено на улучшение способности больших языковых моделей (LLM) понимать и обрабатывать длинные контексты. Авторы предлагают фреймворк AgenticLU, который использует самогенерируемые уточняющие вопросы и механизм указания на релевантный контекст для улучшения понимания длинных текстов. Основной результат: модель AgenticLU значительно превосходит существующие методы промптинга и специализированные LLM для работы с длинным контекстом, достигая надежного многоэтапного рассуждения при сохранении стабильной производительности с увеличением длины контекста.

Объяснение метода:

Исследование предлагает высокоэффективную методологию Chain of Clarifications для работы с длинными контекстами. Пользователи могут адаптировать ключевые концепции (поэтапное уточнение вопросов, указание на релевантные части текста) для повседневного использования LLM, значительно улучшая понимание длинных документов. Техническая сложность некоторых аспектов снижает непосредственную применимость, но концептуальная ценность остается высокой.

Ключевые аспекты исследования: 1. **Chain of Clarifications (CoC)**: Основной метод исследования, где модель улучшает понимание длинных контекстов через самостоятельную генерацию уточняющих вопросов, извлечение релевантного контекста и ответы на эти уточняющие вопросы.

Двухуровневое масштабирование: Процесс построения путей CoC через поиск в дереве, где каждый шаг CoC представляет узел. Это позволяет достичь 97.8% точности извлечения ответов на сложные вопросы.

Дистилляция путей CoC: После сбора данных из процесса поиска в дереве модель обучается генерировать эффективные уточнения и контекстные привязки за один проход, устраняя необходимость масштабирования при выводе.

Двухэтапное обучение: Включает (1) SFT для обучения эффективным стратегиям декомпозиции и (2) DPO для улучшения качества рассуждений.

Механизм Pointback: Позволяет модели указывать на релевантные части длинного контекста, обеспечивая точную информационную привязку.

Дополнение:

Применимость методов в стандартном чате без дообучения

Исследование AgenticLU представляет методы, которые **не требуют обязательного дообучения или специального API** для базового применения. Хотя авторы использовали дообучение для максимальной эффективности, основные концепции могут быть адаптированы для стандартных чатов.

Ключевые концепции для адаптации:

Chain of Clarifications (CoC): Пользователи могут вручную реализовать этот подход, задавая LLM серию уточняющих вопросов перед переходом к окончательному ответу. Например: "Прочитай этот текст и скажи, какие уточняющие вопросы нужно задать, чтобы лучше понять [основной вопрос]" "Теперь найди в тексте информацию, относящуюся к этому уточняющему вопросу" "На основе найденной информации, ответь на уточняющий вопрос" "Теперь ответь на исходный вопрос"

Механизм Pointback: Можно имитировать, прося модель:

"Укажи номера абзацев или разделов, которые содержат релевантную информацию" "Цитируй конкретные части текста, на которые опираешься в своем ответе"

Пошаговое рассуждение: Можно попросить модель:

"Разбей свой анализ на четкие этапы" "Для каждого утверждения указывай, из какой части документа ты берешь эту информацию"

Ожидаемые результаты от адаптации:

• Повышение точности: Структурированный подход снижает вероятность "потери контекста" при работе с длинными текстами

• Лучшая прозрачность: Пользователи видят, на какие части текста опирается модель

• Более глубокое понимание: Поэтапное уточнение помогает модели и пользователю лучше понимать сложные взаимосвязи в тексте

Хотя полная автоматизация процесса требует дообучения, концептуальный подход AgenticLU может значительно улучшить работу с длинными контекстами даже в стандартных чатах.

Анализ практической применимости: **Chain of Clarifications (CoC)**: - Прямая

применимость: Высокая. Пользователи могут адаптировать подход для работы с длинными документами, отчетами или книгами, задавая уточняющие вопросы и выделяя релевантные части текста. - Концептуальная ценность: Очень высокая. Демонстрирует, как разбиение сложных вопросов на последовательность уточнений помогает LLM лучше понимать контекст. - Потенциал для адаптации: Высокий. Подход можно применять для любых задач с длинным контекстом, от анализа документов до исследовательской работы.

Двухуровневое масштабирование: - Прямая применимость: Средняя. Технически сложно для обычных пользователей, но концепция поэтапного уточнения может быть использована в упрощенном виде. - Концептуальная ценность: Высокая. Показывает, как итеративное уточнение улучшает точность при работе с длинными текстами. - Потенциал для адаптации: Средний. Требует технических знаний, но идея итеративного поиска применима в упрощенных формах.

Механизм Pointback: - Прямая применимость: Высокая. Пользователи могут просить модель указывать на конкретные фрагменты текста, что повышает прозрачность и точность. - Концептуальная ценность: Очень высокая. Демонстрирует важность привязки ответов к конкретным частям исходного документа. - Потенциал для адаптации: Высокий. Легко интегрируется в обычные запросы к LLM.

Двухэтапное обучение: - Прямая применимость: Низкая. Требует технических ресурсов, недоступных обычным пользователям. - Концептуальная ценность: Средняя. Показывает, как можно улучшить модели, но не дает практических инструментов для пользователей. - Потенциал для адаптации: Низкий. Требует специализированных знаний и ресурсов.

Сводная оценка полезности: Предварительная оценка: 75

Исследование представляет высокую практическую ценность для широкой аудитории пользователей LLM. Ключевые концепции, особенно Chain of Clarifications и механизм Pointback, могут быть непосредственно применены пользователями разного уровня для улучшения работы с длинными текстами.

Контраргумент для более высокой оценки: Методология может быть адаптирована для использования в стандартных чатах без дополнительного обучения, позволяя пользователям структурировать свои запросы по аналогии с CoC.

Контраргумент для более низкой оценки: Исследование опирается на специфические технические аспекты (двухэтапное обучение, поиск в дереве), недоступные обычным пользователям, что снижает его непосредственную применимость.

После рассмотрения контраргументов, корректирую оценку до 72, поскольку несмотря на высокую концептуальную ценность, не все аспекты исследования могут быть непосредственно применены обычными пользователями без технических знаний.

Итоговая оценка: 72

Основания для оценки: 1. Высокая практическая ценность ключевых концепций (CoC, Pointback) 2. Возможность адаптации основных идей для повседневного использования 3. Ограниченная доступность некоторых технических аспектов для обычных пользователей 4. Значительное улучшение понимания того, как эффективно работать с длинными контекстами

Уверенность в оценке: Очень сильная. Исследование четко демонстрирует как технические, так и концептуальные аспекты, которые могут быть полезны для широкой аудитории. Оценка основана на тщательном анализе различных компонентов исследования и их потенциальной пользы для различных групп пользователей.

Оценка адаптивности: Оценка адаптивности: 85

Концепция Chain of Clarifications представляет собой универсальный подход, который может быть легко адаптирован пользователями для работы с любыми LLM. Даже без специального обучения модели, пользователи могут структурировать свои запросы по принципу поэтапного уточнения, задавая серию вопросов и указывая на релевантные части контекста.

Механизм Pointback, хотя и требует технической реализации для автоматического функционирования, концептуально может быть применен пользователями через запросы о конкретных частях текста.

Исследование демонстрирует фундаментальный подход к улучшению работы с длинными контекстами, который может быть реализован различными способами и в различных сценариях, от профессионального анализа документов до повседневного использования LLM для обработки больших объемов информации.

Высокий потенциал для абстрагирования технических методов до общих принципов взаимодействия делает это исследование особенно перспективным для широкого круга пользователей.

|| <Оценка: 72> || <Объяснение: Исследование предлагает высокоэффективную методологию Chain of Clarifications для работы с длинными контекстами. Пользователи могут адаптировать ключевые концепции (поэтапное уточнение вопросов, указание на релевантные части текста) для повседневного использования LLM, значительно улучшая понимание длинных документов. Техническая сложность некоторых аспектов снижает непосредственную применимость, но концептуальная ценность остается высокой.> || <Адаптивность: 85>

Prompt:

Использование исследования AgenticLU в промптах для GPT

Ключевые применимые знания из исследования

Исследование AgenticLU предлагает эффективные методы для работы с длинными контекстами через: - **Chain-of-Clarifications (CoC)** - цепочка самогенерируемых уточняющих вопросов - **Механизм pointback** - явное указание на релевантные части контекста - **Итеративный многоэтапный подход** к рассуждению

Пример промпта с использованием техник AgenticLU

[======] Я приложил длинный документ [ДОКУМЕНТ]. Помоги мне проанализировать его, используя следующий подход:

Сначала задай себе 3-5 ключевых уточняющих вопросов о содержании документа, которые помогут структурировать анализ.

Для каждого уточняющего вопроса:

Найди и процитируй релевантные части документа (используй точное цитирование) Объясни, как эта информация отвечает на уточняющий вопрос Укажи, какие дополнительные уточнения могут потребоваться

После обработки всех уточняющих вопросов, сформулируй итоговый структурированный анализ документа, синтезирующий все найденные ответы.

Важно: для каждого вывода явно указывай, на какую часть документа ты опираешься, цитируя соответствующие фрагменты. [======]

Как это работает

Данный промпт использует три ключевых принципа из исследования AgenticLU:

Самогенерируемые уточнения - модель сама формулирует вопросы, которые помогают ей разбить сложную задачу на подзадачи, что соответствует технике Chain-of-Clarifications

Механизм pointback - требование цитировать релевантные части документа заставляет модель явно указывать, на какие фрагменты она опирается в своих рассуждениях

Многоэтапное рассуждение - структура промпта направляет модель через последовательные шаги анализа, что позволяет справиться со сложными вопросами через итеративный подход

Такой промпт особенно эффективен для: - Анализа длинных документов - Извлечения структурированной информации - Обеспечения прозрачности

рассуждений модели - Повышения точности ответов на сложные вопросы

При необходимости вы можете адаптировать количество уточняющих вопросов и глубину анализа в зависимости от сложности вашего документа.

№ 156. Формирование игры: как контекст влияет на принятие решений ИИ

Ссылка: https://arxiv.org/pdf/2503.04840

Рейтинг: 72
Адаптивность: 80

Ключевые выводы:

Исследование направлено на изучение влияния контекстного фрейминга на принятие решений языковыми моделями (LLM) в игровых сценариях. Основные результаты показывают, что поведение LLM значительно зависит от контекста, в котором представлена задача, даже если базовая структура игры остается неизменной. Эта вариативность в значительной степени предсказуема, но сохраняется определенная доля непредсказуемости.

Объяснение метода:

Исследование демонстрирует, как контекст (тема, отношения между участниками, тип мира) существенно влияет на решения LLM даже при одинаковой базовой структуре задачи. Эти знания позволяют пользователям формировать более эффективные запросы, предвидеть реакции моделей и выбирать подходящие LLM для конкретных задач. Хотя методология требует адаптации, концепции применимы непосредственно.

Ключевые аспекты исследования: 1. **Динамическое контекстное оценивание LLM**: Исследование представляет новую методологию генеративной оценки, которая систематически варьирует контекст для одной и той же базовой структуры задачи (дилемма заключенного), создавая разнообразные сценарии для тестирования LLM.

Влияние контекста на принятие решений: Авторы демонстрируют, как различные контекстные переменные (тема, тип отношений между участниками, тип мира) значительно влияют на решения, принимаемые LLM, даже когда базовая игровая структура остается неизменной.

Предсказуемость контекстной вариабельности: Исследование показывает, что, хотя контекстные эффекты значительно влияют на поведение моделей, эти эффекты в значительной степени предсказуемы с использованием простых методов машинного обучения.

Различия между моделями: Авторы выявляют различия в принятии решений между разными LLM (GPT-4o, Claude, Llama), что указывает на то, что разные модели по-разному реагируют на один и тот же контекст.

Методологические инновации: Предложен подход процедурной генерации

сценариев для оценки LLM, что потенциально решает проблему загрязнения данных в традиционных статических наборах для тестирования.

Дополнение: Для работы методов этого исследования не требуется дообучение или специальный API. Хотя авторы использовали API для масштабного тестирования разных моделей и генерации большого количества виньеток, основные концепции и подходы могут быть применены в стандартном чате.

Вот ключевые концепции, которые можно адаптировать для работы в стандартном чате:

Контекстное обрамление запросов - понимание того, что один и тот же вопрос, заданный в разных контекстах, может привести к разным ответам. Пользователи могут сознательно формировать контекст своих запросов, чтобы получить желаемый тип ответа.

Учет ключевых факторов влияния - исследование выявило три ключевых фактора, влияющих на решения LLM: тема, тип отношений между участниками и тип мира (реальный/воображаемый). Пользователи могут манипулировать этими факторами в своих запросах.

Выбор подходящей модели - исследование показывает, что разные модели по-разному реагируют на один и тот же контекст. Пользователи могут выбирать конкретные модели в зависимости от желаемого типа ответа.

Проверка разных формулировок - исследование демонстрирует, что даже небольшие изменения в формулировке могут привести к разным ответам. Пользователи могут проверять разные формулировки одного и того же вопроса, чтобы найти наиболее эффективную.

Применяя эти концепции, пользователи могут достичь следующих результатов: - Более предсказуемые и согласованные ответы от LLM - Лучшее понимание факторов, влияющих на ответы LLM - Более эффективные запросы, приводящие к желаемым результатам - Повышенное доверие к использованию LLM для решения различных задач

Например, если пользователю нужно получить более кооперативный ответ от модели, он может сформулировать запрос в контексте союзников, обсуждающих глобальную политику 21-го века, так как исследование показало, что в этом контексте модели демонстрируют наивысший уровень кооперации.

Анализ практической применимости: 1. **Динамическое контекстное оценивание LLM** - Прямая применимость: Высокая. Пользователи могут адаптировать свои взаимодействия с LLM, учитывая, что контекст значительно влияет на ответы. Например, переформулирование вопроса в разных контекстах может привести к более предсказуемым или желаемым результатам. - Концептуальная ценность: Очень высокая. Понимание того, что LLM чувствительны к контексту, помогает пользователям формировать более эффективные запросы. - Потенциал для

адаптации: Высокий. Методология может быть упрощена для использования в повседневных взаимодействиях с LLM.

Влияние контекста на принятие решений Прямая применимость: Средняя. Знание о том, что темы, отношения между акторами и тип мира влияют на решения LLM, может помочь пользователям настроить свои запросы для получения более согласованных ответов. Концептуальная ценность: Высокая. Понимание факторов, влияющих на решения LLM, позволяет пользователям лучше интерпретировать и предсказывать ответы. Потенциал для адаптации: Средний. Хотя концепция применима широко, конкретные эффекты могут варьироваться в зависимости от задачи.

Предсказуемость контекстной вариабельности

Прямая применимость: Средняя. Предсказуемость ответов LLM может быть использована для создания более надежных взаимодействий. Концептуальная ценность: Высокая. Понимание того, что вариации в ответах LLM предсказуемы, увеличивает доверие к использованию этих моделей. Потенциал для адаптации: Средний. Хотя полная предсказуемость требует сложных моделей, пользователи могут интуитивно применять эти принципы.

Различия между моделями

Прямая применимость: Высокая. Пользователи могут выбирать конкретные модели в зависимости от желаемого типа ответа или характера задачи. Концептуальная ценность: Средняя. Понимание различий между моделями помогает пользователям делать более информированный выбор модели. Потенциал для адаптации: Высокий. Знание о различиях между моделями может быть непосредственно применено при выборе LLM для конкретных задач.

Методологические инновации

Прямая применимость: Низкая для обычных пользователей, высокая для разработчиков и исследователей. Концептуальная ценность: Средняя. Понимание проблем с традиционными методами оценки может повлиять на интерпретацию результатов LLM. Потенциал для адаптации: Средний. Принципы динамической оценки могут быть применены в упрощенной форме. Сводная оценка полезности: На основе проведенного анализа, предварительная оценка полезности исследования составляет 75 из 100 баллов. Это исследование предоставляет ценные практические и концептуальные знания, которые могут быть непосредственно применены широкой аудиторией для улучшения взаимодействия с LLM.

Контраргументы к этой оценке:

Почему оценка могла бы быть выше: Исследование предлагает революционный подход к пониманию LLM и методологию, которая может значительно улучшить взаимодействие пользователей с AI. Результаты исследования могут быть применены практически мгновенно без необходимости в технических знаниях.

Почему оценка могла бы быть ниже: Исследование сосредоточено на одном конкретном типе задачи (дилемма заключенного), и неясно, насколько хорошо результаты обобщаются на другие типы взаимодействий. Кроме того, методология генерации виньеток требует технических знаний для полной реализации.

После рассмотрения этих аргументов, я корректирую оценку до 72 из 100. Хотя исследование предоставляет высокоценные знания, которые могут быть адаптированы для использования широкой аудиторией, некоторые ограничения в обобщаемости и сложность полной реализации методологии снижают его полезность для среднего пользователя.

Основные причины для этой оценки: 1. Исследование предоставляет непосредственно применимые знания о влиянии контекста на ответы LLM. 2. Результаты могут быть использованы для создания более эффективных запросов и лучшего понимания ответов LLM. 3. Методология может быть адаптирована для различных задач, хотя полная реализация может быть сложной. 4. Выводы о предсказуемости ответов LLM увеличивают доверие к использованию этих моделей. 5. Понимание различий между моделями позволяет делать более информированный выбор модели для конкретных задач.

Уверенность в оценке: Моя уверенность в оценке очень сильная. Я тщательно проанализировал ключевые аспекты исследования и их применимость для широкой аудитории. Исследование предоставляет ясные, конкретные выводы о влиянии контекста на ответы LLM, которые могут быть непосредственно применены пользователями. Методология исследования хорошо описана и обоснована, а результаты согласуются с пониманием того, как работают LLM. Кроме того, авторы предоставляют код для воспроизведения их результатов, что увеличивает надежность и применимость исследования.

Оценка адаптивности: Адаптивность данного исследования оценивается в 80 из 100. Исследование предлагает концепции и принципы, которые могут быть легко адаптированы для использования в стандартном чате с LLM. Вот ключевые факторы, поддерживающие эту оценку:

Основной вывод о влиянии контекста на ответы LLM может быть непосредственно применен в стандартном чате путем сознательного формирования контекста для получения желаемых ответов.

Понимание того, что различные факторы (тема, отношения между акторами, тип мира) влияют на ответы LLM, позволяет пользователям адаптировать свои запросы для получения более согласованных или желаемых результатов.

Методология генерации виньеток, хотя и сложна для полной реализации, может быть упрощена для использования в повседневных взаимодействиях с LLM.

Выводы о предсказуемости ответов LLM могут быть использованы для создания

более надежных взаимодействий.

Понимание различий между моделями позволяет пользователям делать более информированный выбор модели для конкретных задач.

Однако адаптивность исследования ограничена тем, что оно сосредоточено на одном конкретном типе задачи (дилемма заключенного), и неясно, насколько хорошо результаты обобщаются на другие типы взаимодействий. Кроме того, полная реализация методологии требует технических знаний, что может ограничить ее использование некоторыми пользователями.

|| <Оценка: 72> || <Объяснение: Исследование демонстрирует, как контекст (тема, отношения между участниками, тип мира) существенно влияет на решения LLM даже при одинаковой базовой структуре задачи. Эти знания позволяют пользователям формировать более эффективные запросы, предвидеть реакции моделей и выбирать подходящие LLM для конкретных задач. Хотя методология требует адаптации, концепции применимы непосредственно.> || <Адаптивность: 80>

Prompt:

Использование знаний из исследования "Формирование игры" в промптах для GPT
Ключевые выводы для применения

Исследование показывает, что контекстный фрейминг значительно влияет на принятие решений языковыми моделями, даже когда базовая структура задачи остается неизменной. Это можно стратегически использовать при составлении промптов.

Пример промпта с использованием выводов исследования

[======] Я работаю над проектом, требующим совместного принятия решений между двумя конкурирующими компаниями в технологической сфере.

Действуя как нейтральный посредник в глобальной бизнес-среде 21-го века, предложи решение, которое: 1. Способствует долгосрочному сотрудничеству 2. Учитывает интересы обеих сторон 3. Создает взаимовыгодную ситуацию

Важно, чтобы твой ответ был ориентирован на создание союзнических отношений между участниками, а не на конкуренцию.

Представь сначала вариант сотрудничества, а затем альтернативные подходы. [======]

Объяснение применения знаний из исследования

В этом промпте я стратегически использовал несколько факторов, которые согласно исследованию повышают вероятность кооперативного ответа от GPT:

Тип отношений - явно указал на создание "союзнических отношений", так как исследование показало, что модели демонстрируют более высокий уровень кооперации при взаимодействии с союзниками (72% для GPT-4o)

Тематика - использовал контекст "глобальной бизнес-среды 21-го века", так как в современных сценариях наблюдается более высокий уровень кооперации

Порядок представления опций - указал представить "сначала вариант сотрудничества", поскольку исследование выявило, что порядок представления опций влияет на решения LLM

Нейтральная позиция - предложил модели действовать как "нейтральный посредник", что снижает вероятность состязательного подхода

Подобное структурирование промпта, основанное на выводах исследования, значительно повышает вероятность получения кооперативного, взаимовыгодного решения от модели, даже если базовая задача потенциально конфликтна.

№ 160. Генерация входных данных для тестирования значений границ с использованием проектирования подсказок с большими языковыми моделями: обнаружение ошибок и анализ покрытия

Ссылка: https://arxiv.org/pdf/2501.14465

Рейтинг: 71
Адаптивность: 75

Ключевые выводы:

Исследование оценивает эффективность использования больших языковых моделей (LLM) для генерации тестовых входных данных с граничными значениями в контексте тестирования программного обеспечения методом белого ящика. Основные результаты показывают, что LLM, при правильном использовании промптов, могут генерировать тестовые входные данные, сравнимые или превосходящие по эффективности традиционные методы в обнаружении ошибок и покрытии кода в определенных случаях.

Объяснение метода:

Исследование предлагает практичную методологию использования LLM для генерации тестовых данных через простые промпты, которые любой пользователь может адаптировать. Демонстрирует эффективность LLM в обнаружении сложных ошибок и важность качества тестов над количеством. Однако полная ценность требует понимания концепций тестирования и доступа к исходному коду, что ограничивает применимость для некоторых пользователей.

Ключевые аспекты исследования: 1. **Методология использования LLM для генерации тестовых входных данных**: Исследование предлагает фреймворк для оценки эффективности LLM в создании граничных тестовых значений для программного обеспечения, используя инженерию промптов для направления моделей на создание специфических тестовых входных данных.

Сравнение с традиционными методами: Авторы сравнивают тестовые данные, сгенерированные LLM, с данными, полученными традиционными методами (случайное тестирование, конколическое тестирование, машинное обучение для анализа граничных значений), оценивая способность обнаружения ошибок и охват кода.

Оценка эффективности обнаружения ошибок: Исследование анализирует способность LLM-генерированных тестовых наборов выявлять различные типы

ошибок в коде, включая ошибки "off-by-one", которые часто встречаются на границах условий.

Влияние количества тестовых данных: Авторы изучают взаимосвязь между количеством сгенерированных тестовых входных данных и эффективностью тестирования, выявляя, что больший объем тестов не всегда гарантирует лучшие результаты.

Корреляция между охватом кода и обнаружением ошибок: Исследование выявляет положительную корреляцию между охватом ветвей кода и обнаружением ошибок, особенно для тестовых данных, ориентированных на граничные значения.

Дополнение: Исследование не требует дообучения или специального API для применения основных методов. Авторы использовали GPT-4o с простыми промптами для генерации тестовых входных данных. Хотя для анализа результатов применялись специальные инструменты (gcov), сам процесс генерации тестов доступен в стандартном чате.

Концепции и подходы, которые можно применить в стандартном чате:

Генерация граничных тестовых случаев: Используя промпт "Generate boundary value test inputs for c code delimited by triple backticks", можно получить тестовые данные, ориентированные на граничные условия. Этот подход применим к любому коду, который пользователь хочет протестировать.

Сбалансированный подход к количеству тестов: Исследование показывает, что качество тестовых данных важнее их количества. Пользователи могут запрашивать небольшие, но хорошо продуманные наборы тестов.

Адаптация промптов для разных языков программирования: Хотя исследование фокусируется на C/C++, тот же подход можно применять для Python, JavaScript и других языков.

Фокус на конкретных типах ошибок: Можно модифицировать промпты для поиска конкретных типов ошибок, например: "Generate test cases that would identify off-by-one errors in this function".

Итеративное улучшение тестов: Пользователи могут анализировать результаты выполнения сгенерированных тестов и запрашивать уточненные тесты на основе обнаруженных проблем.

Применяя эти концепции, пользователи могут значительно улучшить качество своего тестирования без необходимости в специализированных инструментах. Результаты включают более надежный код, раннее обнаружение ошибок и лучшее понимание потенциальных проблемных мест в программах.

Prompt:

Использование знаний из исследования о граничных значениях в промптах для GPT
Ключевые выводы исследования для промптов

Исследование показывает, что большие языковые модели (LLM) могут эффективно генерировать тестовые данные с граничными значениями, иногда превосходя традиционные методы тестирования. Особенно важно качество промптов, а не количество сгенерированных тестов.

Пример промпта для тестирования граничных значений

[======] # Запрос на генерацию тестовых данных с граничными значениями

Контекст программы Я разрабатываю функцию, которая проверяет валидность возраста пользователя для регистрации на сайте. Возраст должен быть от 18 до 120 лет.

Код функции [======]python def validate_age(age): if isinstance(age, (int, float)) and 18 <= age <= 120: return True return False [======]

Запрос Сгенерируй набор тестовых входных данных с граничными значениями для этой функции. Для каждого значения укажи: 1. Само значение 2. Ожидаемый результат (True/False) 3. Граничное условие, которое проверяется

Особенно сфокусируйся на: - Точных граничных значениях (17, 18, 119, 120) - Значениях рядом с границами - Экстремальных значениях - Нетипичных входных данных (строки, None, отрицательные числа) [======]

Почему это работает

Данный промпт эффективен, потому что:

Содержит конкретную информацию о программе — указаны тип, назначение и ограничения функции **Включает исходный код** — позволяет модели точно определить граничные условия **Структурирует запрос** — четко указывает, какие именно данные нужно сгенерировать **Направляет внимание на граничные значения** — явно запрашивает проверку граничных случаев **Запрашивает обоснование** — просит указать, какое граничное условие проверяется Согласно исследованию, такой подход позволяет получить более качественные тестовые данные, которые с большей вероятностью выявят ошибки, особенно на границах допустимых значений, где часто возникают проблемы.

Применение в других контекстах

Этот подход можно адаптировать для различных задач тестирования, включая проверку функций обработки текста, валидации данных, математических вычислений и других областей, где важно тестирование граничных случаев.

№ 164. LogiDynamics: Раскрывая динамику логического вывода в рассуждении больших языковых моделей

Ссылка: https://arxiv.org/pdf/2502.11176

Рейтинг: 70
Адаптивность: 85

Ключевые выводы:

Исследование направлено на изучение динамики логического вывода в рассуждениях больших языковых моделей (LLM). Основная цель - понять, когда и как эффективно использовать различные парадигмы логического вывода (System 1 - прямая индукция и System 2 - абдукция+дедукция) для улучшения способностей LLM к рассуждению. Главные результаты показывают, что эффективность различных подходов к логическому выводу зависит от модальности задачи, уровня сложности и формата задания.

Объяснение метода:

Исследование демонстрирует, когда использовать прямые запросы (для текстовых/простых задач) и когда структурированное рассуждение (для визуальных/сложных задач). Оно предлагает методы улучшения ответов через выбор гипотез, верификацию и уточнение. Выводы экспериментально подтверждены и применимы к широкому спектру задач, хотя требуют базового понимания логических концепций.

Ключевые аспекты исследования: 1. **Сравнительная динамика логических процессов**: Исследование систематически изучает эффективность различных типов логического вывода (индуктивного, абдуктивного и дедуктивного) в LLM при решении задач аналогичного рассуждения в различных контекстах.

Контролируемая среда оценки: Авторы создали среду для оценки рассуждений через три измерения: модальность (текстовая, визуальная, символьная), сложность (легкая, средняя, сложная) и формат задачи (множественный выбор или свободный текст).

Зависимость от характеристик задачи: Исследование выявляет, что эффективность разных типов логического вывода (Система 1 vs Система 2) зависит от модальности, сложности и формата задачи.

Масштабирование логических процессов: Авторы исследуют усовершенствованные методы логического вывода, включая выбор гипотез, верификацию и уточнение, демонстрируя их потенциал для повышения

производительности LLM.

Обобщаемость выводов: Результаты исследования распространяются на более широкие задачи обучения в контексте, что подтверждает универсальность обнаруженных закономерностей.

Дополнение: Действительно ли для работы методов этого исследование требуется дообучение или API? Или методы и подходы можно применить в стандартном чате, а ученые лишь для своего удобства использовали расширенные техники?

Методы и подходы, описанные в исследовании, **не требуют дообучения или специального API** и могут быть применены в стандартном чате с LLM. Исследователи использовали контролируемую экспериментальную среду для систематической оценки, но сами методы логического вывода реализуются через обычные текстовые запросы.

Концепции и подходы, которые можно применить в стандартном чате:

Выбор между Системой 1 и Системой 2 в зависимости от типа задачи: Для текстовых/простых задач: использовать прямые запросы (индуктивный вывод) Для визуальных/символических/сложных задач: использовать двухэтапное рассуждение (абдуктивно-дедуктивный подход)

Абдуктивно-дедуктивный подход можно реализовать через запрос, разделенный на две части:

Сначала определи паттерн или правило в [примерах]. Затем примени это правило к [новой ситуации].

Выбор гипотез можно реализовать через запрос: Предложи несколько возможных объяснений или решений для [проблемы]. Затем выбери наиболее вероятное объяснение и обоснуй свой выбор.

Верификация и уточнение можно реализовать через последовательные запросы: Проверь свой предыдущий ответ на наличие ошибок или противоречий. Уточни ответ, исправив обнаруженные проблемы.

Ожидаемые результаты от применения этих концепций: - Повышение точности ответов для сложных задач - Улучшение качества рассуждений в задачах, требующих многоэтапного мышления - Более надежные ответы благодаря верификации и уточнению - Оптимизация взаимодействия с LLM за счет выбора подходящего метода запроса в зависимости от задачи

Исследование фактически предоставляет руководство по эффективному взаимодействию с LLM, которое может быть реализовано обычными пользователями без технических навыков или доступа к API.

Prompt:

Использование исследования LogiDynamics в промптах для GPT ## Ключевые аспекты исследования для промптов

Исследование LogiDynamics показывает, что эффективность рассуждений GPT зависит от: - **Модальности задачи** (текст, визуальные элементы, символы) - **Сложности задачи** (легкая, средняя, сложная) - **Формата задачи** (множественный выбор или свободная генерация)

Пример промпта с применением System 2 для визуальной задачи

[======] # Задание по анализу визуальной последовательности

Инструкция (двухэтапный подход):

Этап 1: Абдукция - выявление закономерности Внимательно изучи следующую последовательность изображений и выяви все возможные закономерности и правила, которые могут объяснять эту последовательность: [Описание изображений 1, 2, 3...]

Сформулируй не менее 3 гипотез о закономерностях с подробным объяснением каждой.

Этап 2: Дедукция - применение правила Теперь примени каждую выявленную закономерность к следующему элементу последовательности: - Для гипотезы 1: логически выведи, какой должен быть следующий элемент - Для гипотезы 2: логически выведи, какой должен быть следующий элемент - Для гипотезы 3: логически выведи, какой должен быть следующий элемент

Оцени, какая гипотеза наиболее вероятна, и предложи окончательный ответ с подробным обоснованием. [======]

Почему это работает?

Данный промпт применяет ключевые выводы исследования:

Использует System 2 (абдукция+дедукция), что дает преимущество до 38.73% для визуальных задач **Разделяет процесс на два этапа**: Абдуктивный (выявление закономерностей) Дедуктивный (применение закономерностей) **Запрашивает несколько гипотез** (до 5 согласно исследованию), что улучшает качество рассуждения **Включает верификацию** через оценку наиболее вероятной гипотезы ## Другие рекомендации по созданию промптов

• Для текстовых задач: Можно использовать System 1 (прямая индукция), так как преимущество System 2 минимально (6.16%)

• Для задач со свободной генерацией: Предпочтительнее System 1, особенно в

символических задачах

• Для сложных задач: Обязательно использовать System 2 с преимуществом до 37.20%

• Для задач с множественным выбором: System 2 дает значительное преимущество

Правильный выбор подхода к рассуждению в промпте может значительно повысить качество ответов GPT в различных контекстах.

№ 168. ПОПИШИ: Структурированное рассуждение Больших Языковых Моделей с экстраполяцией достоверности, вдохновленной графами знаний

Ссылка: https://arxiv.org/pdf/2410.08475

Рейтинг: 70
Адаптивность: 75

Ключевые выводы:

Исследование представляет новый метод GIVE (Graph Inspired Veracity Extrapolation) для улучшения рассуждений больших языковых моделей (LLM) путем объединения параметрической и непараметрической памяти. Основная цель - повысить точность рассуждений LLM с минимальным внешним вводом. Результаты показывают, что GIVE значительно улучшает производительность LLM разных размеров, позволяя даже меньшим моделям превосходить более крупные в научных задачах.

Объяснение метода:

GIVE предлагает мощный метод структурированного рассуждения с использованием ограниченной внешней информации. Хотя полная реализация технически сложна, ключевые концепции (разбиение запроса, экстраполяция на основе ограниченных фактов, контрфактуальное рассуждение) могут быть адаптированы обычными пользователями для улучшения взаимодействия с LLM и получения более достоверных ответов в сложных областях знаний.

Ключевые аспекты исследования: 1. **Структурированное рассуждение с графовым подходом**: Исследование представляет метод GIVE (Graph-Inspired Veracity Extrapolation), который объединяет параметрическую память LLM с непараметрическими знаниями для улучшения рассуждений в задачах, требующих специализированных знаний.

Экстраполяция достоверности: Метод не просто извлекает информацию из внешних источников, а использует ограниченные экспертные данные как отправную точку для дивергентного мышления, позволяя LLM связывать запрос с неполной информацией.

Многоэтапное структурированное рассуждение: GIVE создает группы связанных сущностей, устанавливает внутригрупповые и межгрупповые связи, а также определяет промежуточные сущности для многошагового рассуждения.

Контрфактуальное рассуждение: Метод включает проверку потенциальных

связей, отбрасывая неверные, что помогает избегать галлюцинаций модели при недостаточности знаний.

Прогрессивная генерация ответов: GIVE использует поэтапный подход к формированию ответа, сначала с утвердительными знаниями, затем с контрфактуальными, и наконец с экспертными знаниями.

Дополнение: Исследование GIVE не требует обязательного дообучения или специального API для своей работы, это метод инференса, который может быть адаптирован для стандартного чата. Авторы использовали стандартные LLM (GPT-3.5, GPT-4, Llama 3) без дообучения, просто направляя их с помощью специальных промптов.

Концепции и подходы, которые можно применить в стандартном чате:

Структурированное разбиение вопроса - пользователь может попросить модель выделить ключевые понятия и отношения в вопросе перед ответом.

Формирование групп связанных концепций - можно предложить модели сначала перечислить связанные концепции для каждого ключевого понятия.

Двухэтапное рассуждение - сначала установить связи внутри групп концепций, затем между группами.

Контрфактуальная проверка - попросить модель не только подтвердить возможные связи, но и опровергнуть неверные.

Прогрессивное формирование ответа - сначала получить предварительный ответ, затем уточнить его с учетом дополнительных соображений.

Пример адаптации: при ответе на медицинский вопрос пользователь может сначала попросить модель выделить ключевые термины, затем для каждого термина перечислить связанные понятия, установить связи между ними, проверить потенциальные утверждения и сформировать итоговый ответ. Это позволит получить более структурированное и достоверное рассуждение даже без доступа к графам знаний.

Результаты: значительное улучшение качества ответов в сложных областях знаний, снижение галлюцинаций, более прозрачное рассуждение, которое пользователь может проследить и проверить.

Prompt:

Использование методологии GIVE в промптах для GPT ## Основные принципы GIVE

Методология GIVE (Graph Inspired Veracity Extrapolation) предлагает структурированный подход к рассуждениям, который объединяет параметрическую

и непараметрическую память для улучшения точности ответов языковых моделей.

Пример промпта, вдохновленного GIVE

[======] Я хочу, чтобы ты помог мне разобраться в теме [ТЕМА] используя структурированный подход к рассуждению.

Следуй этим шагам:

Выдели 3-5 ключевых концепций из этой темы. Для каждой концепции определи группу тесно связанных понятий (2-3 понятия). Для каждой группы опиши внутренние связи между понятиями, используя свои базовые знания. Установи логические связи между разными группами понятий. На основе этой структуры знаний, сформулируй последовательное объяснение темы [ТЕМА]. Представь результат в виде: - Сначала - список ключевых концепций - Затем - группы связанных понятий с их внутренними связями - Далее - межгрупповые связи - И наконец - итоговое объяснение темы [======]

Как работают принципы GIVE в этом промпте

Извлечение ключевых концепций - промпт просит модель идентифицировать основные элементы темы **Построение групп связанных сущностей** - модель формирует кластеры связанных понятий **Индукция внутригрупповых связей** - модель использует свои параметрические знания для описания отношений внутри групп **Экстраполяция достоверности** - установление межгрупповых связей помогает модели проверить согласованность своих знаний **Прогрессивная генерация ответа** - финальное объяснение строится на основе структурированного графа знаний Этот подход помогает: - Уменьшить галлюцинации модели - Сделать рассуждения более логичными и последовательными - Улучшить точность в специализированных областях знаний - Получить более структурированный и обоснованный ответ

Даже если модель не обладает полными знаниями по теме, такая структура помогает ей лучше организовать имеющуюся информацию и выявить пробелы в рассуждениях.

№ 172. Должны ли вы использовать вашу модель большого языка для исследования или эксплуатации?

Ссылка: https://arxiv.org/pdf/2502.00225

Рейтинг: 70
Адаптивность: 85

Ключевые выводы:

Исследование оценивает способность современных больших языковых моделей (LLM) помогать в принятии решений, требующих баланса между исследованием и использованием. Основной вывод: LLM эффективны в качестве инструментов исследования больших пространств действий с семантическим смыслом, но менее эффективны для оптимизации решений на основе имеющихся данных, особенно в сложных задачах.

Объяснение метода:

Исследование демонстрирует высокую ценность в понимании возможностей LLM для исследования больших пространств действий (стратегии запросов легко применимы), но ограниченную полезность для задач оптимизации на основе числовых данных (требуются технические навыки). Предоставляет важные концептуальные знания о том, когда и как использовать LLM для принятия решений.

Ключевые аспекты исследования: 1. **Исследование возможностей LLM как оракулов эксплуатации (exploitation)** - авторы оценивают способность языковых моделей (GPT-4, GPT-4o, GPT-3.5) определять оптимальные действия на основе предыдущей истории взаимодействий в задачах многоруких бандитов (MAB) и контекстных бандитов (CB).

Исследование LLM как оракулов исследования (exploration) - авторы изучают, насколько эффективно LLM могут предлагать разнообразные кандидатные действия в пространствах с огромным количеством возможных вариантов.

Методы улучшения эксплуатации в контексте - исследуются различные техники, такие как поиск ближайших соседей, кластеризация k-means и их комбинации, для повышения эффективности LLM при принятии решений на основе истории.

Сравнение с алгоритмическими базовыми методами - авторы сопоставляют производительность LLM с традиционными алгоритмами (линейная регрессия, случайный выбор) для объективной оценки преимуществ и недостатков моделей.

Практические эксперименты с текстовыми задачами - тестирование на задачах

открытых философских вопросов и генерации заголовков для научных статей для оценки возможностей LLM в реальных сценариях.

Дополнение:

Исследование не требует дообучения или API для применения основных концепций. Хотя авторы использовали API для проведения экспериментов, большинство подходов можно адаптировать для стандартного чата.

Концепции и подходы для стандартного чата:

Использование LLM для исследования (exploration) - можно применять предложенные стратегии запросов ("all at once" и "one-by-one") для получения разнообразных вариантов решений в любой предметной области. Пользователи могут: Запрашивать несколько альтернативных решений одной проблемы Последовательно генерировать варианты, показывая модели предыдущие решения Явно запрашивать разнообразие в ответах

Структурирование информации - исследование показывает, что LLM лучше работают с правильно структурированной информацией. Пользователи могут:

Организовывать числовые данные в удобочитаемые таблицы Выделять наиболее релевантные примеры (аналог k-nearest) Группировать похожие случаи (упрощенная версия k-means)

Понимание ограничений - осознание, что для задач с числовыми данными LLM не всегда оптимальны, может помочь пользователям:

Запрашивать качественные рассуждения, а не точные числовые расчеты Использовать LLM для генерации идей, а не для принятия окончательных решений Комбинировать сильные стороны LLM (генерация вариантов) с другими методами Применение этих концепций позволит получить: - Более разнообразные и творческие решения проблем - Лучшее понимание возможных подходов к сложным задачам - Более эффективное использование LLM, фокусируясь на их сильных сторонах

Prompt:

Использование знаний из исследования о LLM в промптах ## Ключевые выводы для применения в промптах

Исследование показывает, что большие языковые модели (LLM) лучше всего работают как инструменты **исследования** (генерации вариантов), но хуже справляются с **эксплуатацией** (выбором оптимального решения на основе данных).

Пример промпта для генерации вариантов заголовков

[======] Я хочу использовать ваши способности к исследованию пространства возможных решений. Мне нужно создать 5 вариантов заголовков для статьи о влиянии искусственного интеллекта на образование.

Для каждого нового варианта учитывайте предыдущие и создавайте заголовок, который существенно отличается по подходу или фокусу (используйте метод one-by-one с высоким разнообразием).

После генерации всех вариантов, я буду использовать отдельный алгоритм для выбора лучшего заголовка, поэтому сосредоточьтесь на разнообразии и креативности, а не на попытке угадать, какой вариант я предпочту.

Тема статьи: [описание темы статьи] Целевая аудитория: [описание аудитории] Тон: [формальный/неформальный/др.] [======]

Объяснение эффективности

Этот промпт работает эффективно, потому что:

Использует LLM для исследования - просит модель генерировать разнообразные варианты, что соответствует сильной стороне LLM согласно исследованию **Применяет метод "one-by-one"** - просит учитывать предыдущие варианты при создании новых, что увеличивает разнообразие **Ограничивает количество вариантов до 5** - исследование показало, что оптимальное число генерируемых вариантов составляет 3-5 **Явно указывает на разделение задач** - модель фокусируется на генерации, а не на выборе лучшего варианта, что соответствует выводам исследования о слабости LLM в задачах эксплуатации Такой подход позволяет получить максимальную пользу от сильных сторон LLM (креативное исследование пространства возможностей), избегая их ограничений (оптимизация на основе данных).

№ 176. Улучшение разговорных агентов с теорией разума: согласование убеждений, желаний и намерений для взаимодействия, похожего на человеческое

Ссылка: https://arxiv.org/pdf/2502.14171

Рейтинг: 70
Адаптивность: 75

Ключевые выводы:

Исследование направлено на улучшение взаимодействия между LLM-системами и людьми путем внедрения теории разума (Theory of Mind, ToM). Основная цель - изучить, насколько языковые модели могут улавливать и использовать информацию о ментальных состояниях (убеждениях, желаниях и намерениях) для более человекоподобного взаимодействия. Результаты показали, что внедрение ToM-информации в процесс генерации ответов значительно улучшает качество взаимодействия, достигая показателей выигрыша 67% и 63% для моделей Llama 3 размером 3B и 8B соответственно.

Объяснение метода:

Исследование предлагает ценную BDI-модель (убеждения, желания, намерения) для улучшения диалога с LLM. Хотя технические методы требуют специальных навыков, принципы могут быть адаптированы для структурирования промптов. Наглядные примеры демонстрируют преимущества учета ToM. Пользователи могут применять концепцию для более эффективного взаимодействия с LLM в переговорах и обсуждениях.

Ключевые аспекты исследования: 1. **Теория разума (ToM) для LLM**: Исследование изучает, насколько языковые модели могут понимать и отслеживать убеждения, желания и намерения участников диалога (BDI-модель) для более человекоподобного взаимодействия.

Извлечение ToM из внутренних репрезентаций: Авторы используют метод LatentQA для извлечения информации о ToM из активаций нейронных сетей и проверяют её согласованность.

Управление выводом через ToM-компоненты: Исследователи демонстрируют возможность манипулировать внутренними представлениями ToM для получения более согласованных с контекстом ответов.

Экспериментальная валидация: Проведены эксперименты на различных наборах

данных (диалоги о кемпинге, переговоры о товарах), показывающие 67% и 63% выигрыша для моделей Llama3 3B и 8B соответственно при использовании ToM-информации.

Практическая применимость: Показано, что средние слои LLM содержат наиболее полезную информацию о ToM, которая может быть использована для улучшения качества диалогов.

Дополнение:

Применимость методов в стандартном чате

Хотя исследование использует сложные технические методы (LatentQA) для извлечения и манипулирования внутренними представлениями ToM, основные концепции могут быть применены в стандартном чате без необходимости в дообучении или API.

Ключевые адаптируемые концепции:

BDI-модель для структурирования промптов Пользователи могут явно указывать в промптах: Убеждения (beliefs): что каждый участник диалога знает или думает Желания (desires): что каждый участник хочет получить Намерения (intentions): какие действия участники планируют предпринять

Последовательное отслеживание ToM

При длительных диалогах пользователи могут периодически обновлять информацию о ментальных состояниях участников Например: "Учитывая, что пользователь выразил предпочтение X, а я выразил потребность в Y..."

Явное указание приоритетов

В сценариях переговоров пользователи могут явно указывать приоритеты: "Для меня высокий приоритет имеет X, средний приоритет Y, низкий приоритет Z"

Эмпатическое взаимодействие

Использование намерения "Show empathy" путем явного указания на необходимость учета чувств и потребностей собеседника **Ожидаемые результаты:** - Более контекстно-зависимые и персонализированные ответы - Повышение эффективности в сценариях переговоров и обсуждений - Более естественное и человекоподобное взаимодействие - Улучшенное отслеживание потребностей пользователя в длительных диалогах

Таким образом, хотя исследователи использовали сложные технические методы для удобства экспериментов, основные принципы ToM могут быть эффективно применены в стандартном чате путем явного структурирования промптов с учетом BDI-модели.

Prompt:

Использование теории разума в промптах для GPT ## Ключевое понимание из исследования

Исследование показывает, что языковые модели могут лучше взаимодействовать с людьми, если в них внедрена **теория разума (ToM)** — способность понимать и отслеживать ментальные состояния собеседника через: - **Убеждения** (beliefs) — что человек считает истинным - **Желания** (desires) — чего человек хочет достичь - **Намерения** (intentions) — какие планы есть у человека

Пример эффективного промпта с использованием ToM

[======] # Инструкция для GPT с использованием теории разума

Ты помощник в переговорах о цене товара. Во время диалога тебе нужно:

Отслеживать убеждения клиента: Что клиент думает о реальной стоимости товара Какие параметры товара он считает важными

Определять желания клиента:

Какую максимальную цену он готов заплатить Какие дополнительные ценности он ищет помимо цены

Понимать намерения клиента:

Хочет ли он действительно купить или просто исследует рынок Планирует ли он использовать информацию для торга в другом месте После каждого сообщения клиента, перед формированием ответа, проанализируй эти три компонента и адаптируй свой ответ, чтобы он был согласован с ментальным состоянием клиента.

При ответе не указывай явно, что ты отслеживаешь эти компоненты, просто используй эту информацию для создания более эффективного и эмпатичного ответа. [======]

Почему это работает

Использование средних слоев модели — исследование показало, что информация ToM лучше представлена в средних слоях модели, и этот промпт помогает активировать эти представления

Структурирование по BDI-модели (Belief-Desire-Intention) — явно указывая модели отслеживать все три компонента, мы задействуем более глубокое понимание контекста

Динамическое отслеживание — промпт направляет модель на постоянное обновление своего понимания ментального состояния собеседника, что согласуется с выводами исследования о необходимости адаптации к изменяющимся представлениям

Неявное применение — промпт указывает не демонстрировать механизм работы, а просто использовать его, что делает взаимодействие более естественным

Такой подход к созданию промптов может повысить эффективность взаимодействия с GPT на 60-67%, согласно результатам исследования.

№ 180. SecureFalcon: Удалось ли нам достичь автоматического обнаружения уязвимостей в программном обеспечении с помощью LLM?

Ссылка: https://arxiv.org/pdf/2307.06616

Рейтинг: 70
Адаптивность: 75

Ключевые выводы:

Исследование направлено на создание эффективной модели для автоматического обнаружения уязвимостей в программном обеспечении с использованием больших языковых моделей (LLM). Основной результат - разработка SecureFalcon, компактной модели на основе Falcon-40B, которая достигает 94% точности в бинарной классификации и до 92% в мультиклассовой классификации уязвимостей, превосходя существующие модели при мгновенном времени вывода на CPU.

Объяснение метода:

Исследование демонстрирует эффективное применение LLM для обнаружения уязвимостей в коде с высокой точностью (94%). Предлагаемая архитектура SecureFalcon и методология имеют значительную ценность для разработчиков и могут быть интегрированы в инструменты разработки. Однако узкая специализация (только C/C++ код) и необходимость значительных ресурсов для воспроизведения ограничивают непосредственную применимость для широкой аудитории.

Ключевые аспекты исследования: 1. **Создание SecureFalcon** - компактная модель с 121 миллионом параметров, основанная на FalconLLM40B, специально настроенная для обнаружения уязвимостей в программном обеспечении. 2. **Использование двух наборов данных для обучения**: FormAI (синтетические данные, созданные с помощью GPT-3.5-turbo и проверенные ESBMC) и FalconVulnDB (агрегированный набор данных из нескольких публичных источников). 3. **Высокая точность обнаружения уязвимостей**: 94% в бинарной классификации (уязвимый/неуязвимый код) и 92% в многоклассовой классификации (определение конкретного типа уязвимости). 4. **Превосходство над традиционными ML-моделями и другими LLM**: SecureFalcon превосходит традиционные алгоритмы машинного обучения на 11% и существующие модели LLM, такие как BERT, RoBERTa и CodeBERT, на 4%. 5. **Быстрое время вывода**: модель обеспечивает время вывода, достаточное для интеграции в системы завершения кода в режиме реального времени.

Дополнение: Для работы методов этого исследования действительно требуется дообучение модели, так как SecureFalcon представляет собой специально настроенную версию FalconLLM40B. Однако многие концепции и подходы могут

быть адаптированы для использования в стандартном чате с LLM без необходимости в дообучении.

Концепции и подходы, которые можно применить в стандартном чате:

Структурированный анализ кода Можно формулировать промпты, которые просят LLM анализировать код по определенной структуре: сначала искать проблемы с управлением памятью, затем проблемы с вводом данных и т.д. Пример: "Проанализируй этот C-код шаг за шагом, сначала проверяя на утечки памяти, затем на переполнение буфера, затем на проблемы с указателями."

Использование примеров из наборов данных

В промпты можно включать примеры уязвимостей из известных наборов данных (например, CWE) для сравнения Пример: "Вот пример кода с уязвимостью CWE-119 (переполнение буфера): [пример]. Проверь, содержит ли мой код похожие уязвимости."

Многоэтапная проверка

Можно разбить анализ кода на несколько этапов, сначала запрашивая общий анализ, затем уточняя конкретные аспекты Пример: "Сначала укажи все подозрительные участки кода, затем для каждого участка определи тип возможной уязвимости."

Использование специализированной терминологии

Включение в запросы специфических терминов и концепций из CWE и других стандартов Пример: "Проверь этот код на наличие уязвимостей из категорий CWE-120, CWE-476 и CWE-190."

Контрпримеры и проверка

Можно просить LLM генерировать контрпримеры для проверки наличия уязвимостей Пример: "Если в этом коде есть уязвимость переполнения буфера, приведи конкретный пример входных данных, которые могут вызвать эту уязвимость." Потенциальные результаты от применения этих подходов: - Повышение точности обнаружения уязвимостей в коде по сравнению с простым запросом "найди ошибки в коде" - Более структурированный и систематический анализ кода - Лучшее понимание типов уязвимостей и их причин - Возможность обнаружения более сложных и неочевидных уязвимостей - Повышение осведомленности разработчиков о потенциальных проблемах безопасности

Хотя такой подход не достигнет точности специально обученной модели (94%), он может значительно улучшить результаты анализа кода в стандартном чате с LLM.

Prompt:

Использование знаний из исследования SecureFalcon в промтах для GPT ## Ключевые знания из исследования

Исследование SecureFalcon демонстрирует высокую эффективность специализированных LLM в обнаружении уязвимостей в коде:

• 94% точность в бинарной классификации (уязвимый/безопасный код)

• 92% точность в мультиклассовой классификации (определение конкретных типов уязвимостей)

• Особенно высокая точность (близкая к 100%) для определенных типов уязвимостей:

• CWE-78 (OS Command Injection)

• CWE-121 (Stack-Based Buffer Overflow)

• CWE-122 (Heap-Based Buffer Overflow)

• CWE-762 (Mismatched Memory Management)

Пример промта для GPT

[======] Я хочу, чтобы ты выступил в роли эксперта по безопасности программного обеспечения, используя знания, аналогичные модели SecureFalcon.

Проанализируй следующий фрагмент кода на C/C++ и: 1. Определи, содержит ли код уязвимости (да/нет) 2. Если уязвимости присутствуют, классифицируй их по стандарту CWE 3. Особенно обрати внимание на: - OS Command Injection (CWE-78) - Stack-Based Buffer Overflow (CWE-121) - Heap-Based Buffer Overflow (CWE-122) - Mismatched Memory Management (CWE-762) 4. Предложи исправления для обнаруженных уязвимостей

Код для анализа: [======]c void process_user_input(char *input) { char command[100]; sprintf(command, "echo %s", input); system(command);

char *buffer = malloc(10); strcpy(buffer, input); // Обработка данных free(buffer); buffer[0] = '\0'; } [======]

Формат ответа: - Уязвимость обнаружена: [Да/Нет] - Идентифицированные CWE: [список] - Подробный анализ: [описание каждой уязвимости] - Рекомендуемые исправления: [код с исправлениями] [======]

Как работают знания из исследования в этом промте

Структура запроса: Промт опирается на способность моделей, подобных SecureFalcon, выполнять бинарную и мультиклассовую классификацию

уязвимостей.

Фокус на конкретных типах уязвимостей: Промт специально указывает на типы уязвимостей, которые модель SecureFalcon определяет с высокой точностью (близкой к 100%).

Комплексный анализ: Запрос требует не только обнаружения уязвимостей, но и их классификации по стандарту CWE, что соответствует возможностям SecureFalcon в мультиклассовой классификации.

Практическое применение: Промт отражает одно из практических применений SecureFalcon, упомянутых в исследовании — анализ кода на наличие уязвимостей в процессе разработки.

Такой подход позволяет эффективно использовать общие языковые модели для задач, в которых специализированные модели (как SecureFalcon) показывают высокие результаты.

№ 184. Генерация ключевых фраз без обучения: исследование специализированных инструкций и агрегации многократных образцов на больших языковых моделях

Ссылка: https://arxiv.org/pdf/2503.00597

Рейтинг: 70
Адаптивность: 85

Ключевые выводы:

Исследование направлено на изучение возможностей больших языковых моделей (LLM) для задачи генерации ключевых фраз (KPG) в режиме zero-shot. Авторы систематически исследуют эффективность специализированных инструкций в промптах и разрабатывают стратегии агрегации результатов из нескольких сэмплов. Основной вывод: мультисэмплинг с правильной стратегией агрегации значительно улучшает производительность LLM для задачи KPG.

Объяснение метода:

Исследование предлагает высокоэффективные стратегии мульти-сэмплинга и агрегации результатов, которые значительно улучшают генерацию ключевых фраз. Особенно ценны методы Frequency order и динамический выбор количества результатов, которые легко адаптируются для широкого спектра задач. Однако некоторые исследованные подходы (специализированные промпты, дополнительные инструкции) оказались неэффективными, а специфика задачи генерации ключевых фраз ограничивает широкую применимость.

Ключевые аспекты исследования: 1. **Исследование эффективности специализированных инструкций для генерации ключевых фраз (keyphrases)** - изучение влияния специфических промптов для создания "присутствующих" (present) и "отсутствующих" (absent) ключевых фраз с использованием LLM в режиме zero-shot.

Анализ влияния дополнительных инструкций по контролю количества и порядка ключевых фраз - исследование эффективности промптов, которые явно указывают модели упорядочивать ключевые фразы по релевантности и контролировать их количество.

Исследование мульти-сэмплинга для улучшения генерации ключевых фраз - тестирование различных стратегий агрегации результатов из нескольких запросов к LLM (Union, UnionConcat, UnionInterleaf, Frequency order) для повышения качества генерируемых ключевых фраз.

Сравнительный анализ производительности различных LLM (Llama-3, Phi-3, GPT-4o) - оценка эффективности различных моделей в задаче генерации ключевых фраз на пяти разных наборах данных (Inspec, Krapivin, SemEval, KP20K, KPTimes).

Разработка метода динамического выбора количества ключевых фраз - предложение алгоритма для автоматического определения оптимального количества ключевых фраз при агрегации результатов мульти-сэмплинга.

Дополнение: Исследование не требует дообучения или специального API для применения большинства описанных методов. Ключевые концепции могут быть реализованы в стандартном чате:

Мульти-сэмплинг и стратегии агрегации: Пользователь может сделать несколько запросов к модели с одним и тем же промптом Результаты можно агрегировать вручную, используя стратегии из исследования: Frequency order: выбирать ключевые фразы, которые чаще всего встречаются в разных ответах UnionInterleaf: брать по одной ключевой фразе из каждого ответа поочередно UnionConcat: объединять ответы последовательно, удаляя дубликаты

Динамический выбор количества результатов:

Пользователь может рассчитать среднее количество ключевых фраз в нескольких ответах и использовать это число для ограничения финального списка

Простота промптов:

Исследование показывает, что базовые промпты часто работают не хуже сложных специализированных, что упрощает взаимодействие с моделью

Ранжирование по перплексии:

Хотя обычный пользователь не может напрямую измерить перплексию, можно попросить модель оценить уверенность в каждом из своих ответов и использовать эту информацию для ранжирования Применяя эти концепции, пользователи могут значительно повысить качество генерации ключевых фраз и других подобных задач, просто используя стандартный интерфейс чата с LLM и объединяя результаты нескольких запросов по рекомендованным стратегиям.

Prompt:

Использование исследования по генерации ключевых фраз в промптах для GPT ## Ключевые знания из исследования

Исследование показало, что: 1. Мультисэмплинг с частотным ранжированием значительно улучшает результаты генерации ключевых фраз 2.

Специализированные промпты не всегда дают преимущество над базовыми 3. LLM хорошо работают для задачи KPG в режиме zero-shot

Пример эффективного промпта

[======] [Текст документа]

Сгенерируй 5 наборов ключевых фраз для данного текста. Каждый набор должен содержать 7-10 ключевых фраз, которые наиболее точно отражают основное содержание и важные концепции документа. Ключевые фразы могут быть как присутствующими в тексте напрямую, так и отсутствующими (абстрактными концепциями).

После генерации всех 5 наборов, проанализируй их и создай финальный список ключевых фраз, ранжированный по частоте встречаемости каждой фразы или близких по смыслу фраз в разных наборах.

Формат ответа: 1. Набор 1: [список ключевых фраз] 2. Набор 2: [список ключевых фраз] 3. Набор 3: [список ключевых фраз] 4. Набор 4: [список ключевых фраз] 5. Набор 5: [список ключевых фраз]

Финальный ранжированный список ключевых фраз: 1. [Самая частая ключевая фраза] - [количество появлений] 2. [Вторая по частоте ключевая фраза] - [количество появлений] ... [======]

Почему это работает

Данный промпт применяет ключевые находки исследования:

Использует мультисэмплинг — генерирует 5 разных наборов ключевых фраз для одного документа, что увеличивает разнообразие и полноту результатов.

Применяет частотное ранжирование — наиболее эффективная стратегия агрегации по данным исследования, которая позволяет выявить действительно важные ключевые фразы, встречающиеся в разных сэмплах.

Не разделяет промпт на специализированные типы (для присутствующих/отсутствующих фраз), что согласуется с выводом исследования о том, что базовый промпт с правильной стратегией агрегации работает не хуже специализированных.

Использует динамический подход к количеству ключевых фраз, позволяя модели самой определить оптимальное число в заданном диапазоне.

Такой подход максимально использует преимущества LLM для генерации ключевых фраз в режиме zero-shot, что делает его эффективным для практического применения без необходимости дополнительного обучения моделей.

№ 188. Пауза-Настройка для Понимания Долгого Контекста: Легкий Подход к Перенастройке Внимания LLM

Ссылка: https://arxiv.org/pdf/2502.20405

Рейтинг: 70
Адаптивность: 85

Ключевые выводы:

Исследование направлено на решение проблемы 'Lost in the middle' (LITM) у больших языковых моделей, когда они плохо обрабатывают информацию в середине длинных контекстов. Авторы предлагают технику 'pause-tuning', которая перераспределяет внимание модели для улучшения понимания длинных контекстов. Результаты показывают значительное улучшение производительности моделей Llama 3 при извлечении информации из длинных контекстов (до 128K токенов).

Объяснение метода:

Исследование предлагает методы улучшения работы с длинными контекстами через вставку пауз-токенов. Часть методов (вставка пауз без файнтюнинга) доступна для непосредственного применения обычными пользователями. Концепция структурирования длинных запросов с паузами проста для понимания и решает актуальную проблему "lost in the middle", значительно улучшая извлечение информации из длинных текстов.

Ключевые аспекты исследования: 1. **Pause-tuning** - техника улучшения работы LLM с длинными контекстами путем вставки специальных "пауз-токенов" в текст, которые перераспределяют внимание модели на всё содержимое, решая проблему "lost in the middle" (LITM).

Методы вставки пауз-токенов - исследованы пять различных подходов к вставке пауз: стандартные паузы после каждого абзаца, паузы с инструкциями, предварительная инструкция с паузами, файнтюнинг для длинных контекстов и файнтюнинг модели со стандартными паузами.

Эффективность перераспределения внимания - анализ показал, что паузы работают как "якоря", прерывающие затухание внимания в длинных последовательностях, что позволяет модели лучше обрабатывать каждый сегмент текста.

Легковесность метода - в отличие от многих других методов для работы с длинными контекстами, pause-tuning не требует значительных вычислительных

ресурсов или изменения базовой архитектуры модели.

Экспериментальные результаты - тесты на задаче "needle in a haystack" (поиск информации в длинном контексте) показали значительное улучшение производительности: до 10.61% у LLaMA 3 23B и 3.57% у LLaMA 3 18B.

Дополнение:

Применимость без дообучения или API

Исследование демонстрирует, что хотя наилучшие результаты достигаются при использовании дообученной модели (Техника 5 - Pause-tuned model), значительные улучшения можно получить и без дообучения, используя только модификацию промптов (Техники 1-3).

Ключевые концепции и подходы, применимые в стандартном чате:

Стандартные пауз-токены (Техника 1): Вставка явных маркеров паузы после каждого абзаца. В стандартном чате можно использовать специальные символы или фразы (например, "[ПАУЗА]", "---СТОП И ОБДУМАЙ---").

Инструкции с паузами (Техника 2): Вставка пауз с явными инструкциями для модели "остановиться и усвоить информацию". Например: "[ПАУЗА - пожалуйста, обдумай вышеизложенное, прежде чем продолжить]".

Предварительная инструкция (Техника 3): Добавление в начало запроса общей инструкции о необходимости делать паузы при обработке длинного текста.

Ожидаемые результаты применения: - Улучшение извлечения информации из середины длинных текстов - Более равномерное распределение внимания модели на весь контекст - Повышение точности ответов на вопросы, требующие информации из середины контекста

Примечательно, что на Рисунках 2 и 3 видно, что даже простые методы вставки пауз (Техники 1-3) показывают улучшение по сравнению с базовой моделью, особенно при работе с контекстами средней длины (16K-64K токенов).

Анализ практической применимости: 1. **Pause-tuning как техника** - Прямая применимость: Средняя. Обычные пользователи не могут напрямую применить полный метод, так как он требует файнтюнинга модели. Однако они могут имитировать подход, вставляя в свои запросы явные "паузы" или сегментирующие маркеры. - Концептуальная ценность: Высокая. Понимание того, что LLM страдают от проблемы "lost in the middle" и что структурирование информации с паузами может улучшить обработку, дает пользователям ценное понимание принципов работы LLM. - Потенциал для адаптации: Высокий. Идея структурирования длинных запросов с паузами может быть адаптирована для использования в обычных промптах.

Методы вставки пауз-токенов Прямая применимость: Высокая. Пользователи могут сразу применить методы 1-3 (вставка пауз, пауз с инструкциями, предварительная инструкция) в своих промптах без необходимости файнтюнинга. Концептуальная ценность: Высокая. Понимание различных способов вставки пауз и их эффективности помогает пользователям выбрать наиболее подходящий метод. Потенциал для адаптации: Высокий. Пользователи могут экспериментировать с различными формами "пауз" в своих запросах.

Перераспределение внимания

Прямая применимость: Низкая. Обычные пользователи не могут напрямую манипулировать механизмами внимания. Концептуальная ценность: Высокая. Понимание того, как работает внимание в LLM, помогает пользователям лучше структурировать свои запросы. Потенциал для адаптации: Средний. Знание о том, как перераспределяется внимание, может помочь в разработке стратегий для работы с длинными текстами.

Легковесность метода

Прямая применимость: Средняя. Хотя файнтюнинг требует технических знаний, сам принцип вставки пауз прост. Концептуальная ценность: Высокая. Понимание того, что можно улучшить работу с длинными контекстами без сложных вычислительных методов. Потенциал для адаптации: Высокий. Простота метода делает его доступным для широкого круга применений.

Экспериментальные результаты

Прямая применимость: Низкая. Результаты сами по себе не могут быть применены. Концептуальная ценность: Высокая. Количественное подтверждение эффективности метода дает пользователям уверенность в его использовании. Потенциал для адаптации: Средний. Данные о производительности различных методов могут помочь в выборе стратегии. Сводная оценка полезности: Исходя из анализа, я оцениваю полезность исследования для широкой аудитории пользователей LLM в **70 баллов** из 100.

Основания для высокой оценки: - Методы 1-3 (вставка пауз без файнтюнинга) могут быть напрямую применены обычными пользователями в повседневных запросах к LLM - Исследование дает четкое понимание проблемы "lost in the middle" и способов ее решения - Концепция структурирования длинных запросов с паузами проста для понимания и применения - Результаты показывают значительное улучшение работы с длинными контекстами, что актуально для многих пользователей

Контраргументы к оценке: 1. Почему оценка могла бы быть выше: - Исследование предлагает простой и эффективный метод, который может значительно улучшить работу с длинными контекстами - Проблема "lost in the middle" широко распространена и актуальна для многих пользователей

Почему оценка могла бы быть ниже: Наиболее эффективный метод (pause-tuning) требует файнтюнинга модели, что недоступно обычным пользователям Исследование фокусируется на специфической задаче "needle in a haystack", которая не всегда соответствует реальным сценариям использования После рассмотрения этих аргументов, я сохраняю оценку в **70 баллов**, так как хотя наиболее эффективный метод требует файнтюнинга, более простые методы также показывают улучшение и могут быть применены непосредственно пользователями.

Основания для итоговой оценки: 1. Исследование предлагает как сложные методы (файнтюнинг), так и простые (вставка пауз), которые могут быть использованы широкой аудиторией 2. Проблема "lost in the middle" актуальна для многих пользователей, работающих с длинными текстами 3. Концепция структурирования запросов с паузами проста для понимания и применения 4. Результаты показывают значительное улучшение, что делает методы привлекательными для использования

Уверенность в оценке: Моя уверенность в оценке: **очень сильная**.

Уверенность основана на: 1. Четкости описания методов в исследовании 2. Наличии конкретных количественных результатов 3. Прямой применимости части методов без технических знаний 4. Понятности концепции "пауз" для широкой аудитории 5. Актуальности проблемы "lost in the middle" для многих пользователей

Оценка адаптивности: Оценка адаптивности: **85 из 100**.

Факторы, обосновывающие высокую оценку адаптивности:

Концептуальная адаптивность: Принцип вставки пауз для сегментирования длинных текстов легко адаптируется для использования в обычном чате. Пользователи могут вставлять явные маркеры пауз, разделители или инструкции по "остановке и обдумыванию" в свои запросы.

Простота адаптации: Хотя полный метод pause-tuning требует файнтюнинга, основная идея — сегментирование длинного контекста — может быть реализована пользователями через структурирование запросов (например, использование заголовков, разделителей, нумерации).

Универсальность принципа: Концепция преодоления "lost in the middle" через перераспределение внимания применима к широкому спектру задач, не ограничиваясь задачей "needle in a haystack".

Масштабируемость: Метод работает для контекстов различной длины, от нескольких тысяч до 128K токенов, что делает его применимым для разных сценариев использования.

Техническая доступность: Три из пяти исследованных методов не требуют файнтюнинга и могут быть непосредственно использованы в обычном чате.

|| <Оценка: 70> || <Объяснение: Исследование предлагает методы улучшения работы с длинными контекстами через вставку пауз-токенов. Часть методов (вставка пауз без файнтюнинга) доступна для непосредственного применения обычными пользователями. Концепция структурирования длинных запросов с паузами проста для понимания и решает актуальную проблему "lost in the middle", значительно улучшая извлечение информации из длинных текстов.> || <Адаптивность: 85>

Prompt:

Использование Pause-Tuning в промптах для GPT
Что такое Pause-Tuning?

Исследование предлагает технику **"pause-tuning"**, которая помогает языковым моделям лучше обрабатывать длинные контексты, особенно решая проблему "Lost in the middle" (LITM), когда модель плохо обрабатывает информацию из середины длинного текста.

Практическое применение в промптах

Основная идея заключается во вставке специальных **токенов паузы** (<*pause*>) в длинные тексты, которые служат "якорями внимания" и позволяют модели лучше фокусироваться на всех частях контекста.

Пример промпта с использованием Pause-Tuning

[======] Я собираюсь предоставить тебе длинный юридический документ для анализа. После каждого абзаца я буду вставлять метку . Когда ты видишь эту метку, остановись и тщательно обдумай информацию в предыдущем абзаце, прежде чем двигаться дальше.

Документ: Настоящий договор заключается между компанией А, именуемой в дальнейшем "Заказчик", и компанией Б, именуемой в дальнейшем "Исполнитель", о нижеследующем.

Предметом договора является разработка программного обеспечения согласно техническому заданию, представленному в Приложении 1.

Стоимость работ составляет 1,500,000 рублей без учета НДС. Оплата производится в три этапа: 30% предоплата, 30% после демонстрации прототипа, 40% после финальной приемки.

[... продолжение документа ...]

Проанализируй этот договор и выдели ключевые обязательства сторон, сроки выполнения и потенциальные юридические риски. [======]

Как это работает?

Токены-якори: Метки <*pause*> служат якорями, которые прерывают затухание внимания в длинных последовательностях **Перераспределение внимания**: Модель уделяет больше внимания всем частям текста, включая середину **Улучшение извлечения информации**: Особенно эффективно для поиска конкретных фактов в длинных документах

Другие способы применения

• Комбинирование токенов паузы с явными инструкциями для модели

• Использование в задачах суммаризации длинных документов

• Применение в системах вопросно-ответного типа с большими базами знаний

• Вставка пауз между разделами научных статей или технических документов

Хотя исследование показало наибольшую эффективность на моделях Llama 3, принцип можно применять и при работе с GPT, особенно когда требуется обработка длинных контекстов.

№ 192. CallNavi: Исследование и вызов маршрутизации и вызова функций в крупных языковых моделях

Ссылка: https://arxiv.org/pdf/2501.05255

Рейтинг: 68
Адаптивность: 75

Ключевые выводы:

Исследование посвящено оценке способности больших языковых моделей (LLM) выполнять функциональные вызовы API. Основная цель - изучить, как LLM справляются с выбором правильных API из большого списка, генерацией параметров и выполнением сложных многошаговых и вложенных вызовов API. Главные результаты показывают, что коммерческие модели OpenAI (GPT-4o и GPT-4o-mini) значительно превосходят другие модели в точности и стабильности вызовов API, а предложенные методы асинхронной генерации и обратного вывода могут существенно улучшить производительность моделей.

Объяснение метода:

Исследование предлагает практичные методы оптимизации работы с API (асинхронная генерация, обратное мышление), применимые обычными пользователями. Понимание влияния сложности задач на производительность моделей и сравнительный анализ 17 LLM помогают формировать эффективные запросы и выбирать подходящие модели. Основные концепции могут быть адаптированы для различных задач.

Ключевые аспекты исследования: 1. **Бенчмарк функциональных вызовов API**: Исследование представляет набор данных CallNavi для оценки способности языковых моделей выбирать правильные API из большого списка (более 100 кандидатов), выполнять последовательные и вложенные вызовы API с корректными параметрами.

Градация сложности задач: Задачи разделены на три уровня сложности (легкие, средние, сложные), что позволяет оценить способность моделей обрабатывать от простых одиночных вызовов API до сложных многошаговых и вложенных вызовов.

Метрики оценки и стабильности: Предложены новые метрики, включая "стабильность вывода", которая оценивает согласованность ответов модели при многократных запусках.

Методы оптимизации: Разработаны два подхода для улучшения производительности моделей: асинхронная генерация (разделение выбора API и

генерации параметров) и обратное мышление для сложных задач.

Сравнительный анализ 17 моделей: Проведено тестирование широкого спектра моделей от коммерческих (GPT-4o) до открытых (Llama, Gemma) и специализированных (Nexus Raven, Gorilla).

Дополнение:

Применимость методов в стандартном чате

Исследование CallNavi представляет методы, которые **не требуют дообучения или специального API** для применения в стандартном чате:

Асинхронная генерация - разделение сложного запроса на два этапа: Сначала определение необходимых действий/API Затем заполнение параметров для этих действий В обычном чате пользователь может сначала запросить план действий, а затем детализировать каждый шаг.

Обратное мышление - планирование от конечного результата к начальным шагам: Определение конечной цели Выявление промежуточных шагов, необходимых для достижения цели Этот подход показал улучшение на 30% в сложных задачах и может быть применен в обычном чате.

Структурирование запросов по сложности - разбиение сложных задач на простые шаги, что улучшает точность ответов. ### Ожидаемые результаты применения

• Повышение точности в многошаговых задачах

• Улучшение структурированности ответов

• Снижение количества ошибок в сложных запросах

• Повышение стабильности ответов при повторных запросах

Хотя для исследования использовались расширенные техники (например, для оценки результатов), основные концепции полностью применимы в стандартном чате без дополнительного обучения моделей.

Prompt:

Использование исследования CallNavi в промптах для GPT ## Ключевые применимые знания из отчета

Разделение сложных задач API на этапы выбора API и генерации параметров **Метод обратного вывода** для итеративного улучшения решений **Различная эффективность моделей** для задач разной сложности **Повышение стабильности**

результатов генерации ## Пример промпта с применением знаний из исследования

[======] # Запрос на вызов API с разделением задачи

Контекст Мне нужно реализовать функциональность, которая [краткое описание задачи]. У меня есть доступ к следующим API:

[список доступных API с их описаниями]

Инструкции (используя метод асинхронной генерации из исследования CallNavi)

Сначала определи, какие API из списка наиболее подходят для решения моей задачи. Предоставь ТОЛЬКО названия нужных API и краткое обоснование выбора.

После моего подтверждения выбора API, сгенерируй конкретные параметры для вызова каждого API, обращая внимание на их правильный синтаксис и типы данных.

Предложи последовательность вызовов API с полными параметрами.

Примени метод обратного вывода: проверь, соответствует ли предложенное решение всем требованиям задачи, и при необходимости итеративно улучши его.

Пожалуйста, отвечай структурированно, разделяя каждый шаг. [======]

Объяснение эффективности промпта

Этот промпт использует два ключевых метода из исследования CallNavi:

Асинхронная генерация — разделение задачи на выбор API и генерацию параметров позволяет модели сосредоточиться на каждом шаге отдельно, что по данным исследования повышает точность на ~30% для сложных задач.

Метод обратного вывода — заставляет модель проверить свое решение и итеративно улучшить его, что особенно важно для сложных многошаговых вызовов.

Такой структурированный подход значительно повышает вероятность получения синтаксически корректного и функционально точного результата, особенно при работе со сложными API-вызовами, как показало исследование CallNavi.

№ 196. Две головы лучше, чем одна: Двухмодельная вербальная рефлексия во время вывода

Ссылка: https://arxiv.org/pdf/2502.19230

Рейтинг: 68
Адаптивность: 75

Ключевые выводы:

Исследование направлено на улучшение способности больших языковых моделей (LLM) к рассуждению через создание двухмодельной системы для рефлексии и уточнения рассуждений. Основной результат - разработка фреймворка DARS (Dual-model Reflective Scoring), который превосходит традиционные методы оптимизации предпочтений по всем метрикам оценки, демонстрируя, что специализированная модель-критик может эффективно направлять модель-рассуждатель к более точным выводам.

Объяснение метода:

Исследование представляет ценную концепцию разделения ролей рассуждения и критики в LLM. Хотя техническая реализация сложна для обычных пользователей, принципы могут быть адаптированы через структурированные запросы и многошаговый диалог. Высокая концептуальная ценность и методология структурированного дерева мышления дают практические инструменты для улучшения качества взаимодействия с LLM.

Ключевые аспекты исследования: 1. **Двухмодельная рефлексивная система (DARS)** - исследование предлагает фреймворк с двумя отдельными моделями: Reasoner (модель-рассуждатель) и Critic (модель-критик), которые работают совместно для улучшения качества рассуждений LLM.

Контрастный синтез рефлексии - метод генерации данных для обучения, который выявляет расхождения между правильными и неправильными рассуждениями и создает вербальные инструкции по исправлению ошибок.

Вербальное обучение с подкреплением (VRL) - фреймворк использует итеративный процесс, где модель-критик предоставляет обратную связь модели-рассуждателю для улучшения ее выводов, без необходимости дополнительного обучения в момент вывода.

Разделение ролей рассуждения и критики - решение системной проблемы конфликта ролей в LLM, когда одна модель должна и обнаруживать ошибки, и исправлять их.

Структурированное дерево мышления - формализованный подход к представлению рассуждений, позволяющий систематически выявлять ошибки в логике.

Дополнение:

Можно ли применить методы исследования в стандартном чате?

Да, ключевые концепции исследования можно адаптировать для использования в стандартном чате без необходимости дообучения моделей или доступа к API. Хотя авторы использовали отдельно обученные модели для достижения максимальной эффективности, основные принципы могут быть реализованы через структурированные промпты.

Применимые концепции и подходы:

Разделение ролей рассуждения и критики Пользователь может запросить LLM сначала решить задачу, а затем в следующем запросе попросить проанализировать предыдущее решение с критической точки зрения Пример: "Реши эту задачу" => "Теперь выступи в роли критика и проанализируй возможные ошибки в предыдущем решении"

Структурированное дерево мышления

Можно попросить LLM структурировать рассуждения в виде последовательных бинарных решений Пример: "Реши задачу, разбивая процесс на дерево решений, где каждый узел представляет бинарный выбор"

Итеративное улучшение через вербальную обратную связь

Пользователь может имитировать процесс VRL через последовательные уточняющие запросы Пример: "Вот твое предыдущее решение [решение]. Улучши его, исправив следующие недостатки [список проблем]"

Контрастный анализ

Можно запросить LLM предоставить несколько альтернативных решений и затем сравнить их Пример: "Предложи два разных подхода к решению этой задачи, а затем сравни их преимущества и недостатки" ### Ожидаемые результаты:

• Повышение точности и глубины рассуждений

• Более структурированные и обоснованные ответы

• Выявление и исправление ошибок в логике рассуждений

• Улучшенная прозрачность процесса принятия решений

Важно отметить, что эффективность этих адаптированных подходов будет ниже, чем у специально обученных моделей, но они все равно могут значительно улучшить качество взаимодействия с LLM в стандартном чате.

Prompt:

Применение исследования DARS в промптах для GPT ## Ключевые принципы для использования

Исследование "Две головы лучше, чем одна: Двухмодельная вербальная рефлексия во время вывода" предлагает несколько важных принципов, которые можно применить при работе с GPT:

Разделение ролей: Использование подхода "рассуждатель + критик" **Структурированные деревья мышления**: Формализация процесса рассуждения **Контрастный анализ**: Сравнение различных путей рассуждения **Итеративное улучшение**: Пошаговая коррекция на основе обратной связи ## Пример промпта с применением DARS

[======] # Задача: Оценить экономические последствия климатического законодательства X

Инструкции Я хочу, чтобы ты выполнил эту задачу в два этапа:

Этап 1: Рассуждатель В роли экономического аналитика: 1. Определи ключевые положения законодательства X 2. Проанализируй краткосрочные экономические эффекты (1-3 года) 3. Проанализируй долгосрочные экономические эффекты (5-10 лет) 4. Сформулируй общее заключение о вероятных экономических последствиях

Этап 2: Критик После завершения анализа, в роли экономического критика: 1. Проверь каждый шаг рассуждения на логические ошибки 2. Выяви возможные упущенные факторы или альтернативные сценарии 3. Сравни результаты с аналогичными историческими прецедентами 4. Предложи конкретные улучшения для первоначального анализа

Этап 3: Улучшенное заключение На основе критического анализа: 1. Представь улучшенную версию экономического анализа 2. Выдели изменения по сравнению с первоначальным анализом 3. Оцени уровень уверенности в новых выводах [======]

Как это работает

Реализация двухмодельного подхода: Хотя мы используем одну модель GPT, мы имитируем двухмодельную систему через четкое разделение ролей и этапов рассуждения.

Структурированное рассуждение: Промпт задает четкую структуру для построения "дерева мышления", что помогает модели организовать свои рассуждения более систематично.

Контрастный анализ: На этапе критики модель сравнивает различные пути рассуждения и выявляет расхождения, что соответствует методике контрастного синтеза рефлексии из исследования.

Итеративное улучшение: Финальный этап позволяет модели применить критический анализ для улучшения первоначального рассуждения, что имитирует процесс обратной связи между моделями в DARS.

Такой подход позволяет получить более глубокий и взвешенный анализ, чем при использовании стандартных промптов, поскольку модель вынуждена критически пересматривать собственные рассуждения.

№ 200. За пределами точного совпадения: семантическая переоценка извлечения событий с помощью крупных языковых моделей

Ссылка: https://arxiv.org/pdf/2410.09418

Рейтинг: 68
Адаптивность: 75

Ключевые выводы:

Основная цель исследования - разработка надежной семантической системы оценки извлечения событий (RAEE), которая выходит за рамки точного токенного соответствия. Главные результаты показывают, что существующие методы оценки значительно недооценивают производительность моделей извлечения событий, особенно генеративных моделей и LLM.

Объяснение метода:

Исследование предлагает ценную концепцию семантической оценки извлечения событий, демонстрируя, что LLM работают значительно лучше, чем показывают стандартные метрики. Пользователи могут применить принципы семантической оценки вместо точного совпадения, что улучшит интерпретацию ответов. Понимание типичных ошибок помогает формулировать более эффективные запросы. Однако полная реализация методологии требует значительной адаптации.

Ключевые аспекты исследования: 1. **Проблема точного совпадения (Exact Match)**: Исследование выявляет существенные недостатки традиционного метода оценки извлечения событий (Event Extraction) на основе точного совпадения токенов, что приводит к неправильной оценке моделей, особенно генеративных и LLM.

Семантическая оценка RAEE: Авторы предлагают новую систему оценки RAEE (Reliable and Semantic Evaluation), которая использует LLM в качестве оценочных агентов, учитывая семантический контекст, а не только точное соответствие токенов.

Адаптивный механизм: Исследователи внедрили адаптивный механизм, позволяющий настраивать критерии оценки для различных задач и наборов данных, что повышает надежность и согласованность с человеческими оценками.

Переоценка существующих моделей: Авторы провели комплексную переоценку 14 моделей извлечения событий на 10 датасетах, обнаружив, что их реальная производительность значительно выше, чем показывают традиционные метрики.

Детальный анализ причин ошибок: В исследовании проведен подробный анализ причин ошибочных оценок при использовании точного совпадения и выявлены типичные паттерны ошибок при семантической оценке.

Дополнение:

Применимость методов в стандартном чате

Исследование не требует дообучения или API для применения основных концепций. Хотя авторы использовали продвинутые LLM как оценщиков для получения численных результатов, основные принципы семантической оценки вместо точного совпадения могут быть применены в любом стандартном чате.

Концепции, применимые в стандартном чате:

Семантическая оценка ответов: Пользователи могут оценивать ответы LLM на основе их смысла, а не точного соответствия ожидаемым словам. Это особенно полезно при извлечении информации из текстов.

Использование LLM для проверки ответов: Пользователь может попросить модель оценить собственный предыдущий ответ или уточнить его, используя принципы из исследования.

Ключевые критерии оценки: Можно формулировать запросы с конкретными критериями приемлемости ответов (например, "важно сохранить ключевые слова, но допустимы синонимы").

Понимание типичных ошибок: Знание о типичных ошибках (отсутствие ключевых слов, неправильная классификация) помогает формулировать более точные запросы.

Ожидаемые результаты от применения:

Более точная интерпретация ответов LLM при извлечении информации Снижение разочарования от кажущихся "неправильных" ответов, которые семантически верны Улучшение формулировок запросов с учетом типичных ошибок LLM Использование многоэтапного процесса, где LLM сначала извлекает информацию, а затем проверяет свои результаты Эти концепции не требуют технической реализации RAEE и могут быть использованы непосредственно в диалоге с любой LLM.

Prompt:

Использование результатов исследования RAEE в промптах для GPT ## Ключевые выводы из исследования для применения в промптах

Исследование показывает, что традиционные методы оценки извлечения событий

(точное токенное соответствие) значительно недооценивают эффективность языковых моделей, особенно генеративных. Семантическая оценка даёт более точную картину их возможностей.

Пример промпта с применением знаний из исследования

[======] # Задача извлечения событий из текста

Контекст Я хочу извлечь события из следующего текста, используя ваши семантические способности. Исследования показывают, что языковые модели могут эффективно извлекать события, даже если их формулировки не совпадают с точными токенами в тексте.

Инструкции 1. Прочитайте текст: [ВСТАВИТЬ ТЕКСТ] 2. Извлеките все события, уделяя внимание: - Семантически эквивалентным выражениям (не только точным совпадениям) - Кореференциям (когда одно и то же событие упоминается разными способами) - Правильной классификации типов событий и аргументов

Для каждого события укажите: Тип события Триггер события (слово или фраза, указывающая на событие) Аргументы события (участники, время, место и т.д.) Уровень уверенности в извлечении (высокий/средний/низкий) ## Формат вывода Представьте результаты в структурированном формате JSON, где каждое событие содержит все вышеперечисленные элементы. [======]

Объяснение эффективности этого промпта

Данный промпт использует ключевые выводы из исследования RAEE следующим образом:

Использует семантические возможности модели: Промпт явно указывает на необходимость выявления семантически эквивалентных выражений, а не только точных совпадений.

Учитывает кореференции: Исследование показало, что это частая причина ошибок при традиционной оценке.

Фокусируется на правильной классификации: Исследование выявило, что даже при семантической оценке это остаётся основной причиной ошибок.

Включает указание уровня уверенности: Позволяет модели сигнализировать о случаях, где может потребоваться дополнительная проверка.

Использует адаптивный подход к формулировке задачи: Предоставляет чёткий контекст и структуру, что, согласно исследованию, повышает согласованность результатов.

Такой подход к составлению промптов позволяет максимально использовать семантические возможности языковых моделей в задачах извлечения событий, что

приводит к более точным и полным результатам.

№ 204. «Эскалация бенчмаркинга перевода кода на основе LLM в эпоху класс-уровня»

Ссылка: https://arxiv.org/pdf/2411.06145

Рейтинг: 68
Адаптивность: 75

Ключевые выводы:

Исследование направлено на оценку способности современных больших языковых моделей (LLM) выполнять перевод кода на уровне классов, а не только на уровне методов. Основной результат: все LLM показывают значительное снижение производительности при переводе кода на уровне классов по сравнению с уровнем методов, при этом коммерческие LLM (DeepSeek-V3, GPT-4o, Claude 3.5 Sonnet) демонстрируют лучшие результаты.

Объяснение метода:

Исследование предлагает три практические стратегии перевода кода на уровне классов, анализ их эффективности для разных LLM и языков программирования, а также детальную классификацию ошибок. Пользователи могут применять эти стратегии и знание о типичных ошибках для улучшения результатов перевода кода, хотя для полного использования результатов требуется определенная техническая подготовка.

Ключевые аспекты исследования: 1. Создание первого в своем роде бенчмарка ClassEval-T для оценки возможностей LLM в переводе кода на уровне классов (а не только отдельных методов) между Python, Java и C++. 2. Разработка и сравнение трех стратегий перевода кода: целостный перевод (holistic), перевод с минимальными зависимостями (min-dependency) и автономный перевод (standalone). 3. Оценка способности различных LLM (коммерческих и открытых) распознавать и корректно обрабатывать зависимости между полями, методами и библиотеками при переводе кода. 4. Детальный анализ 1243 случаев неудачного перевода кода с классификацией типов ошибок, что позволяет понять ограничения современных LLM. 5. Выявление значительного снижения эффективности LLM при переводе кода на уровне классов по сравнению с переводом на уровне отдельных методов.

Дополнение:

Для работы методов этого исследования не требуется дообучение или API. Большинство подходов можно применить в стандартном чате с LLM. Ученые использовали API для систематической оценки моделей, но сами стратегии перевода кода применимы в обычном диалоге.

Концепции и подходы, применимые в стандартном чате:

Три стратегии перевода кода: Holistic (целостный перевод): передача LLM всего класса целиком для перевода Min-dependency (с минимальными зависимостями): перевод отдельных частей класса с указанием необходимых зависимостей Standalone (автономный): перевод отдельных частей без контекста

Выбор оптимальной стратегии:

Для Python-ориентированных переводов лучше использовать целостную стратегию Для C++-ориентированных переводов можно использовать как целостную, так и стратегию с минимальными зависимостями Для коммерческих LLM (более мощных) целостная стратегия всегда эффективнее

Работа с зависимостями:

Целостная стратегия лучше для сохранения зависимостей между полями Стратегия с минимальными зависимостями лучше для правильного использования библиотек Для зависимостей между методами обе стратегии примерно одинаковы

Проверка типичных ошибок:

Синтаксические ошибки (особенно для C++/Java) Проблемы с библиотеками (отсутствие нужных импортов) Проблемы с использованием функций/переменных (вызовы несуществующих методов) Ошибки согласованности кода Применяя эти подходы в стандартном чате, пользователи могут значительно улучшить качество перевода кода, особенно для сложных задач на уровне классов, а не только отдельных функций.

Prompt:

Использование знаний из исследования о переводе кода на уровне классов в промтах для GPT Исследование о переводе кода на уровне классов предоставляет ценные инсайты, которые можно использовать для создания более эффективных промтов при работе с GPT для задач перевода кода.

Ключевые инсайты для промтов

Целостный подход лучше фрагментации - коммерческие LLM лучше справляются с переводом всего класса сразу **Явное указание зависимостей** - модели часто допускают ошибки в обработке зависимостей **Направление перевода имеет значение** - перевод в Python работает лучше, чем в C++ или Java **Типы распространенных ошибок** - синтаксические ошибки, проблемы с использованием функций/переменных и согласованностью кода ## Пример эффективного промта

[======] # Задача: Перевод класса с Java на Python

Инструкции: 1. Переведи весь класс целиком, не разбивая его на отдельные методы 2. Обрати особое внимание на: - Сохранение всех зависимостей между полями класса - Корректный импорт необходимых библиотек в Python - Согласованность имен методов и переменных во всем классе 3. После перевода проверь код на: - Синтаксические ошибки - Корректное использование всех переменных и функций - Логическую эквивалентность оригинальному коду

Исходный код на Java: [======]java // Вставить полный код класса на Java здесь [======] [======]

Почему это работает

Данный промт использует знания из исследования, потому что:

Запрашивает целостный перевод - согласно исследованию, целостная стратегия перевода показывает лучшие результаты, особенно для коммерческих LLM **Акцентирует внимание на зависимостях** - исследование показало, что осведомленность о зависимостях (DEP) является проблемной областью **Включает проверку на распространенные ошибки** - исследование выявило типичные проблемы, которые мы явно просим проверить **Учитывает направление перевода** - перевод в Python работает лучше, что согласуется с выводами исследования Такой подход должен значительно повысить качество перевода кода по сравнению с простым запросом "переведи этот код с Java на Python".

№ 208. Повторное исследование способности графов к рассуждению больших языковых моделей: случаи изучения в переводе, связанности и кратчайшем пути

Ссылка: https://arxiv.org/pdf/2408.09529

Рейтинг: 68
Адаптивность: 75

Ключевые выводы:

Исследование направлено на анализ способности больших языковых моделей (LLM) к рассуждениям на графах. Основная цель - понять разрыв между теоретическими возможностями LLM (которые теоретически должны справляться с задачами на графах) и их практическими неудачами. Главные результаты показывают, что на производительность LLM в задачах на графах влияют типы связности узлов, размеры графов, способы описания графов и методы именования узлов.

Объяснение метода:

Исследование предоставляет практические рекомендации по оптимальному представлению графов в запросах к LLM: использование списков соседей вместо списков рёбер, последовательное именование узлов, включение алгоритмических подсказок. Выявленные факторы влияния могут применяться для повышения точности ответов в графовых задачах. Ограничением является узкий фокус на графовых задачах и необходимость некоторых технических знаний.

Ключевые аспекты исследования: 1. **Комплексная оценка способностей LLM к рассуждениям на графах**: Исследование систематически анализирует, как LLM справляются с графовыми задачами (определение связности, поиск кратчайшего пути), выявляя расхождение между теоретическими возможностями и практическими результатами.

Выявление ключевых факторов влияния: Авторы идентифицировали факторы, влияющие на эффективность LLM в графовых задачах: тип связности узлов (K-hop, изолированные компоненты, асимметричные связи), размер графа, метод описания графа и способ именования узлов.

Анализ различных методов представления графов: Исследование сравнивает три способа описания графов (матрица смежности, списки соседей, списки рёбер) и их влияние на способность LLM понимать структуру графа.

Влияние обучения и размера модели: Авторы демонстрируют, что увеличение

размера модели и количества обучающих данных значительно улучшает способность LLM решать графовые задачи.

Различия в процессах рассуждения: Обнаружено, что LLM используют разные стратегии рассуждения в зависимости от способа представления графа (списки соседей vs списки рёбер).

Дополнение:

Применимость методов в стандартном чате

Исследование не требует дообучения или API для применения основных выводов. Большинство методов и подходов можно непосредственно применить в стандартном чате с LLM.

Ключевые применимые концепции:

Оптимальное представление графов: Использование списков соседей вместо списков рёбер для более точных результатов Последовательное именование узлов (1, 2, 3...) вместо случайных идентификаторов Использование осмысленных имён узлов вместо абстрактных идентификаторов

Структурирование запросов:

Учёт сложности связности при формулировании задач (разбиение сложных задач на более простые) Адаптация запросов к известным ограничениям LLM (проблемы с k-hop > 3, изолированными компонентами)

Алгоритмические подсказки:

Включение в промпт описания алгоритма (BFS для связности, Дейкстра для кратчайшего пути) Использование Chain-of-Thought промптинга для пошагового решения #### Ожидаемые результаты:

• Повышение точности ответов в задачах определения связности на 20-30%

• Значительное улучшение результатов в задачах поиска кратчайшего пути

• Более последовательные и логичные рассуждения модели

• Снижение количества "галлюцинаций" при работе со структурированными данными

Важно отметить, что хотя авторы использовали специализированные методы для своих экспериментов, основные выводы исследования о влиянии формата представления, именования и алгоритмических подсказок полностью применимы в стандартном чате без какого-либо дообучения.

Prompt:

Применение знаний о графовом рассуждении LLM в промптах ## Ключевые выводы из исследования

Исследование показывает, что эффективность LLM при работе с графами зависит от: - Способа представления графа (список соседей работает лучше, чем список рёбер) - Длины пути между узлами (точность падает с увеличением длины) - Именования узлов (семантически значимые имена повышают точность) - Явного включения алгоритмов (например, BFS) в промпт

Пример эффективного промпта для задачи поиска кратчайшего пути

[======] Я опишу граф в виде списка соседей для каждого узла. Мне нужно найти кратчайший путь между двумя узлами.

Граф: - Alice: Bob, Carol, Dave - Bob: Alice, Eve - Carol: Alice, Frank - Dave: Alice, Grace - Eve: Bob, Frank - Frank: Carol, Eve, Grace - Grace: Dave, Frank

Задача: Найди кратчайший путь от Alice до Grace.

Используй алгоритм поиска в ширину (BFS): 1. Начни с узла Alice 2. Исследуй всех соседей Alice 3. Для каждого непосещенного соседа, добавь его в очередь 4. Продолжай, пока не найдешь Grace или не исчерпаешь все возможные пути 5. Запиши каждый шаг твоего рассуждения 6. В конце укажи найденный путь и его длину [======]

Почему это работает

Представление графа: Используется список соседей вместо списка рёбер, что согласно исследованию даёт лучшую производительность (O(|N|) против O(|E|)).

Семантические имена: Вместо абстрактных идентификаторов (Node1, Node2) используются осмысленные имена (Alice, Bob), что улучшает понимание графа моделью.

Явный алгоритм: В промпт включен алгоритм BFS, что, согласно исследованию, повышает точность результатов на ~8%.

Пошаговое рассуждение: Запрос явно просит модель показать шаги рассуждения, что помогает отслеживать правильность пути и соответствует метрикам Fidelity (Facc) и Path Consistency Ratio (PCR) из исследования.

Ограниченная сложность: Граф небольшой, что соответствует выводу о том, что LLM лучше справляются с графами меньшего размера.

Дополнительные рекомендации

• Для сложных графов разбивайте задачу на подзадачи с меньшим количеством шагов

• При необходимости работы с большими графами используйте наиболее мощные доступные модели (например, GPT-4 вместо GPT-3)

• Для критически важных задач рассмотрите возможность использования специализированных графовых алгоритмов вместо полагания только на LLM

№ 212. RankCoT: Усовершенствование знаний для генерации с увеличением поиска через ранжирование цепочек мышления

Ссылка: https://arxiv.org/pdf/2502.17888

Рейтинг: 68
Адаптивность: 75

Ключевые выводы:

Исследование представляет метод RankCoT (Ranking Chain of Thoughts) для улучшения эффективности Retrieval-augmented Generation (RAG) систем. Основная цель - разработать метод уточнения знаний, который комбинирует преимущества ранжирования и суммаризации для более точного извлечения релевантной информации из внешних источников. Результаты показывают, что RankCoT превосходит базовые модели на 2.5% и эффективно работает с LLM различных масштабов.

Объяснение метода:

RankCoT предлагает ценные методы для улучшения взаимодействия с LLM через структурированные рассуждения, ранжирование и самоанализ. Большинство концепций (множественные CoT, самоанализ, выбор лучших вариантов) могут быть адаптированы обычными пользователями для повышения точности ответов в стандартных чатах, несмотря на некоторые технические аспекты, требующие специальных знаний.

Ключевые аспекты исследования: 1. **RankCoT** - новый метод улучшения знаний для Retrieval-Augmented Generation (RAG) систем, который объединяет ранжирование и цепочки рассуждений (Chain of Thought, CoT) для улучшения генерации ответов.

Механизм ранжирования в CoT - модель генерирует несколько вариантов цепочек рассуждений для каждого документа, затем ранжирует их и выбирает лучшие, чтобы отфильтровать нерелевантные документы и информацию.

Механизм самоанализа - модель выполняет дополнительное уточнение сгенерированных CoT, что повышает качество обучающих данных и уменьшает риск переобучения.

Оптимизация прямых предпочтений (DPO) - метод обучения модели, который помогает ей присваивать более высокие вероятности положительным результатам уточнения знаний, содержащим правильные ответы.

Сокращение длины уточнённых знаний - RankCoT создаёт более короткие, но

эффективные результаты уточнения, что экономит контекст в промпте для LLM.

Дополнение:

Применимость методов без дообучения или API

Исследование RankCoT действительно использует дообучение моделей для оптимальной работы, однако **ключевые концепции можно применить в стандартном чате без дообучения**. Основной подход не требует специальных API или дополнительных моделей.

Адаптируемые концепции для стандартного чата:

Множественные цепочки рассуждений (CoT): Можно запросить модель создать несколько различных цепочек рассуждений для одного вопроса Пример: "Рассмотри этот вопрос с нескольких точек зрения и создай 3 разных цепочки рассуждений"

Ранжирование рассуждений:

После получения нескольких CoT, можно попросить модель сравнить их и выбрать лучшее Пример: "Оцени, какое из этих рассуждений наиболее точно отвечает на вопрос, и объясни почему"

Самоанализ и уточнение:

Двухэтапный процесс, где модель сначала дает ответ, а затем анализирует и улучшает его Пример: "Теперь проанализируй свой ответ, найди возможные ошибки или упущения и предложи улучшенную версию"

Фильтрация нерелевантной информации:

Можно попросить модель явно отделить релевантную информацию от нерелевантной Пример: "Из представленной информации выдели только те части, которые напрямую относятся к вопросу"

Ожидаемые результаты:

• Повышение точности: Структурированные рассуждения и их ранжирование помогают модели сфокусироваться на наиболее релевантной информации

• Сокращение галлюцинаций: Самоанализ и проверка собственных выводов снижают вероятность ошибок

• Более краткие, но информативные ответы: Фокусировка на релевантной информации приводит к более лаконичным, но точным ответам

Хотя эти адаптированные методы могут не быть настолько эффективными, как

полная реализация RankCoT с дообучением, они все равно могут значительно улучшить взаимодействие с LLM в стандартных чатах.

Анализ практической применимости: **1. Механизм RankCoT** - **Прямая применимость**: Высокая. Пользователи могут адаптировать принцип генерации нескольких CoT для документов и их ранжирования в своих запросах к LLM, запрашивая модель генерировать несколько рассуждений и выбирать лучшее. - **Концептуальная ценность**: Очень высокая. Понимание того, что комбинирование ранжирования и рассуждений может значительно улучшить точность ответов, помогает пользователям формулировать более эффективные запросы. - **Потенциал для адаптации**: Высокий. Этот метод можно упростить для использования в обычных чатах, прося LLM генерировать несколько вариантов рассуждений и выбирать наиболее релевантные.

2. Механизм самоанализа - **Прямая применимость**: Средняя. Пользователи могут имитировать этот процесс, запрашивая модель проанализировать и улучшить свои первоначальные ответы. - **Концептуальная ценность**: Высокая. Понимание важности самоанализа помогает пользователям формулировать запросы, требующие от модели перепроверки и уточнения своих рассуждений. - **Потенциал для адаптации**: Высокий. Двухэтапный процесс рассуждения с самоанализом можно легко адаптировать в обычных взаимодействиях с LLM.

3. Сокращение длины уточнённых знаний - **Прямая применимость**: Высокая. Пользователи могут применять эту концепцию для получения более кратких и точных ответов, экономя контекстное окно. - **Концептуальная ценность**: Высокая. Осознание того, что более короткие, но хорошо сформулированные рассуждения могут быть эффективнее длинных, помогает пользователям формулировать лучшие запросы. - **Потенциал для адаптации**: Высокий. Принцип краткости при сохранении ключевой информации универсально применим.

4. Оптимизация прямых предпочтений (DPO) - **Прямая применимость**: Низкая. Требует специальных знаний и доступа к обучению моделей. - **Концептуальная ценность**: Средняя. Понимание принципа оптимизации может помочь в формулировании запросов, но требует специальных знаний. - **Потенциал для адаптации**: Низкий. Сложно адаптировать без технических возможностей обучения моделей.

5. Улучшение использования внешних знаний - **Прямая применимость**: Высокая. Пользователи могут применять подход анализа нескольких источников информации и их интеграции. - **Концептуальная ценность**: Очень высокая. Понимание того, как модели могут лучше использовать внешние знания, помогает пользователям структурировать запросы с внешними источниками. - **Потенциал для адаптации**: Высокий. Принципы интеграции и фильтрации знаний можно применять в обычных запросах к LLM.

Сводная оценка полезности: На основе проведенного анализа, я оцениваю полезность исследования в **68 баллов из 100**.

Обоснование: - Исследование предлагает практические методы улучшения взаимодействия с LLM через структурированные рассуждения и ранжирование информации - Многие концепции (генерация нескольких CoT, самоанализ и выбор лучшего варианта) могут быть непосредственно применены обычными пользователями - Предложенные подходы помогают пользователям понять, как лучше структурировать запросы для получения более точной информации - Методы требуют некоторой адаптации для использования в стандартных чатах, но основные принципы доступны для применения

Контраргументы: 1. Почему оценка могла бы быть выше: Исследование предлагает конкретные методы, которые могут значительно улучшить точность ответов и могут быть адаптированы даже неопытными пользователями.

Почему оценка могла бы быть ниже: Некоторые аспекты исследования, такие как DPO-обучение, требуют технических знаний и не могут быть напрямую использованы обычными пользователями. Также, полная реализация метода требует доступа к API или дообучения моделей. После рассмотрения этих аргументов, я подтверждаю оценку в 68 баллов, так как, несмотря на некоторые технические аспекты, основные концепции исследования могут быть адаптированы и применены широким кругом пользователей.

Уверенность в оценке: Очень сильная. Исследование предлагает конкретные методы, которые могут быть адаптированы для использования в повседневных взаимодействиях с LLM. Основные концепции исследования понятны и применимы, даже если полная техническая реализация требует дополнительных знаний.

Оценка адаптивности: Оценка адаптивности: **75 из 100**

Адаптация принципов и концепций: Основные концепции исследования (генерация нескольких CoT, ранжирование и самоанализ) могут быть легко адаптированы для использования в обычном чате. Пользователи могут запрашивать у LLM несколько вариантов рассуждений и выбирать лучшие, а также применять двухэтапный процесс рассуждения с самоанализом.

Извлечение полезных идей: Исследование предлагает конкретные стратегии для улучшения взаимодействия с LLM, такие как фильтрация нерелевантной информации, структурирование рассуждений и проверка собственных выводов. Эти идеи могут быть применены в различных контекстах.

Потенциал для будущих взаимодействий: Концепции, представленные в исследовании, имеют высокий потенциал для улучшения будущих взаимодействий с LLM, помогая пользователям получать более точные и надежные ответы.

Абстрагирование до общих принципов: Специализированные методы исследования могут быть абстрагированы до общих принципов взаимодействия, таких как "генерация нескольких вариантов рассуждений", "ранжирование информации по релевантности" и "самоанализ и уточнение выводов".

Исследование предлагает методы, которые могут быть адаптированы для использования в обычных чатах без необходимости технической реализации полной системы RAG.

|| <Оценка: 68> || <Объяснение: RankCoT предлагает ценные методы для улучшения взаимодействия с LLM через структурированные рассуждения, ранжирование и самоанализ. Большинство концепций (множественные CoT, самоанализ, выбор лучших вариантов) могут быть адаптированы обычными пользователями для повышения точности ответов в стандартных чатах, несмотря на некоторые технические аспекты, требующие специальных знаний.> || <Адаптивность: 75>

Prompt:

Использование знаний из исследования RankCoT в промптах для GPT
Ключевые применимые аспекты исследования

Исследование RankCoT показывает эффективность комбинирования цепочек рассуждений (Chain of Thought) с ранжированием информации для улучшения точности ответов. Это можно применить в промптах для GPT даже без специальной настройки модели.

Пример промпта с использованием принципов RankCoT

[======]

Задача: Анализ влияния изменения климата на сельское хозяйство
Инструкции:

Рассмотри отдельно каждый из следующих документов: Документ 1: [первый источник информации] Документ 2: [второй источник информации] Документ 3: [третий источник информации]

Для каждого документа:

Кратко изложи ключевые факты Проведи цепочку рассуждений о релевантности и достоверности информации Оцени значимость документа для ответа по шкале от 1 до 10

Проанализируй все три оценки и выбери наиболее релевантную информацию.

Проведи самоанализ: проверь, не упущены ли важные детали, нет ли противоречий в выводах.

Сформулируй окончательный ответ, основываясь на наиболее релевантной

информации. [======]

Объяснение применения принципов RankCoT

Разделение на документы — имитирует подход RankCoT, где модель анализирует каждый источник отдельно **Цепочка рассуждений (CoT)** — просит модель создать цепочку мышления для каждого документа **Ранжирование** — имплементирует оценку значимости каждого источника **Выбор лучшей информации** — аналог выбора лучшего CoT в исследовании **Самоанализ (self-reflection)** — внедряет механизм проверки собственных выводов **Краткий финальный ответ** — соответствует наблюдению, что RankCoT генерирует более короткие, но эффективные результаты Такая структура промпта позволяет добиться более качественного анализа информации и более точных ответов, даже без специальной настройки модели методами DPO, описанными в исследовании.

№ 216. Осведомленное объединение с учетом неопределенности: ансамблевый каркас для снижения галлюцинаций в больших языковых моделях

Ссылка: https://arxiv.org/pdf/2503.05757

Рейтинг: 68
Адаптивность: 75

Ключевые выводы:

Исследование направлено на разработку фреймворка Uncertainty Aware Fusion (UAF) для снижения галлюцинаций в больших языковых моделях (LLM) при ответах на фактологические вопросы. Основной результат: UAF превосходит современные методы снижения галлюцинаций на 8% по фактической точности, сокращая или превосходя разрыв в производительности с GPT-4.

Объяснение метода:

Исследование предлагает ансамблевый метод UAF для снижения галлюцинаций LLM, комбинируя ответы нескольких моделей с учетом их точности и уверенности. Высокая концептуальная ценность основных принципов (использование нескольких моделей, учет уверенности, специализация моделей) позволяет пользователям адаптировать их для повседневного использования, особенно для критически важных запросов, требующих фактической точности.

Ключевые аспекты исследования: 1. **Ансамблевый метод снижения галлюцинаций LLM**: Исследование представляет фреймворк Uncertainty-Aware Fusion (UAF), который стратегически комбинирует ответы нескольких моделей LLM для уменьшения галлюцинаций и повышения фактической точности.

Оценка неопределенности для самооценки LLM: Используются различные методы измерения неопределенности (perplexity, Haloscope, semantic entropy), позволяющие моделям оценивать вероятность галлюцинации в собственных ответах.

Двухмодульная архитектура: UAF состоит из модулей SELECTOR (выбирает лучшие LLM по точности и способности обнаруживать галлюцинации) и FUSER (объединяет ответы выбранных моделей с учетом их точности и неопределенности).

Вариативность сильных сторон моделей: Исследование демонстрирует, что разные LLM превосходят друг друга в разных сценариях, что обосновывает необходимость ансамблевого подхода.

Сравнительный анализ эффективности: UAF превосходит существующие методы снижения галлюцинаций на 8% в фактической точности на нескольких бенчмарках (TruthfulQA, TriviaQA, FACTOR-news).

Дополнение:

Возможности применения методов в стандартном чате

Хотя исследование использует специализированные методы оценки неопределенности, которые требуют доступа к внутренним состояниям моделей или API, основные концепции могут быть адаптированы для использования в стандартном чате:

Ансамблевый подход: Пользователи могут задавать один и тот же вопрос нескольким моделям (или одной модели несколько раз с разными промптами) и сравнивать ответы.

Самооценка уверенности: Можно попросить модель оценить свою уверенность в ответе или указать источники. Например: "Ответь на вопрос X и оцени свою уверенность в ответе по шкале от 1 до 10".

Селекция на основе специализации: Пользователи могут определить, какие модели лучше справляются с определенными типами вопросов, и использовать их соответственно.

Комбинирование ответов: При получении противоречивых ответов от разных моделей, пользователь может запросить модель проанализировать эти ответы и выбрать наиболее достоверный.

Ожидаемые результаты от применения

• Снижение вероятности принятия неверной информации

• Повышение фактической точности для критически важных запросов

• Лучшее понимание ограничений моделей и уровня доверия к их ответам

Важно отметить, что исследование не требует обязательного дообучения или API для основной концепции - комбинирования ответов разных моделей с учетом их уверенности. Ученые использовали специализированные методы для точной количественной оценки, но качественные версии этих подходов доступны в стандартном чате.

Анализ практической применимости: 1. **Ансамблевый метод снижения**

галлюцинаций: - Прямая применимость: Средняя. Пользователи могут вручную применить логику UAF, задавая один вопрос нескольким моделям и выбирая ответ с наиболее высокой уверенностью, но это трудоемко. - Концептуальная ценность: Высокая. Понимание, что комбинирование ответов нескольких моделей дает более точные результаты, важно для построения эффективных стратегий взаимодействия с LLM. - Потенциал для адаптации: Высокий. Пользователи могут адаптировать принцип "спроси несколько моделей и сравни уверенность в ответах" для критически важных запросов.

Оценка неопределенности для самооценки LLM: Прямая применимость: Низкая. Обычные пользователи не имеют доступа к внутренним метрикам неопределенности LLM. Концептуальная ценность: Высокая. Понимание, что модели могут оценивать собственную неопределенность, помогает пользователям формулировать запросы, требующие самооценки модели. Потенциал для адаптации: Средний. Пользователи могут запрашивать модель оценить уверенность в своем ответе или предоставить несколько вариантов ответа.

Двухмодульная архитектура:

Прямая применимость: Низкая. Требуется техническая реализация, недоступная для большинства пользователей. Концептуальная ценность: Средняя. Понимание логики выбора наиболее подходящей модели для конкретной задачи полезно для эффективного использования разных LLM. Потенциал для адаптации: Средний. Пользователи могут создать собственную упрощенную версию, выбирая разные модели для разных типов задач.

Вариативность сильных сторон моделей:

Прямая применимость: Высокая. Пользователи могут выбирать разные модели для разных типов задач, основываясь на их сильных сторонах. Концептуальная ценность: Очень высокая. Понимание, что ни одна модель не превосходит другие во всех задачах, критически важно для эффективного использования LLM. Потенциал для адаптации: Высокий. Пользователи могут создать свои "специализированные команды" моделей для разных типов запросов.

Сравнительный анализ эффективности:

Прямая применимость: Низкая. Результаты бенчмарков сами по себе не предоставляют практические инструменты. Концептуальная ценность: Средняя. Понимание относительной эффективности методов снижения галлюцинаций помогает в выборе стратегий взаимодействия с LLM. Потенциал для адаптации: Низкий. Бенчмарки сложно адаптировать для повседневного использования. Сводная оценка полезности: Предварительная оценка: 62 из 100.

Исследование предлагает подход, который может быть адаптирован для использования обычными пользователями, особенно для критически важных запросов, требующих высокой фактической точности. Хотя полная техническая реализация UAF недоступна для большинства пользователей, основные принципы

(использование нескольких моделей, учет их уверенности, выбор наиболее подходящей модели для конкретной задачи) могут быть применены в упрощенном виде.

Контраргументы к оценке: 1. Почему оценка могла бы быть выше: Исследование предлагает конкретную стратегию повышения фактической точности, которую можно адаптировать для повседневного использования, и демонстрирует значительное улучшение точности (на 8%), что критически важно для задач, требующих фактической достоверности.

Почему оценка могла бы быть ниже: Полная реализация UAF требует технических навыков и доступа к API нескольких моделей, что ограничивает прямую применимость для большинства пользователей. Методы оценки неопределенности, используемые в исследовании, недоступны для обычных пользователей без технической реализации. Скорректированная оценка: 68 из 100.

Повышаю оценку, так как концептуальные идеи исследования (использование нескольких моделей, учет их уверенности, специализация моделей) имеют высокую практическую ценность и могут быть адаптированы пользователями даже без полной технической реализации UAF. Ключевой вывод о том, что ни одна модель не превосходит другие во всех задачах, имеет высокую практическую ценность для эффективного использования LLM.

Уверенность в оценке: Очень сильная. Исследование предлагает конкретные методы, которые могут быть адаптированы для использования обычными пользователями, особенно для критически важных запросов, требующих высокой фактической точности. Основные принципы (использование нескольких моделей, учет их уверенности, выбор наиболее подходящей модели для конкретной задачи) могут быть применены в упрощенном виде без полной технической реализации UAF.

Оценка адаптивности: Оценка адаптивности: 75 из 100.

1) Принципы и концепции исследования хорошо адаптируются для использования в обычном чате. Идея использования нескольких моделей для проверки фактов, учет уверенности модели в ответе и выбор наиболее подходящей модели для конкретной задачи могут быть реализованы пользователями в упрощенном виде.

2) Пользователи могут извлечь несколько полезных идей: а) проверка важных фактов через несколько моделей; б) запрос модели оценить уверенность в своем ответе; в) выбор разных моделей для разных типов задач; г) стратегия комбинирования ответов нескольких моделей для повышения точности.

3) Высокий потенциал для внедрения выводов исследования в будущие взаимодействия с LLM, особенно с развитием интерфейсов для работы с несколькими моделями одновременно и появлением встроенных метрик уверенности.

4) Хорошие возможности для абстрагирования специализированных методов до

общих принципов взаимодействия, таких как "проверяй важные факты через несколько источников" и "учитывай уверенность модели при оценке достоверности ответа".

|| <Оценка: 68> || <Объяснение: Исследование предлагает ансамблевый метод UAF для снижения галлюцинаций LLM, комбинируя ответы нескольких моделей с учетом их точности и уверенности. Высокая концептуальная ценность основных принципов (использование нескольких моделей, учет уверенности, специализация моделей) позволяет пользователям адаптировать их для повседневного использования, особенно для критически важных запросов, требующих фактической точности.> || <Адаптивность: 75>

Prompt:

Использование исследования UAF в промптах для GPT
Ключевые применимые знания из исследования

Ансамблевый подход - разные модели имеют различную точность для разных типов вопросов **Оценка неопределенности** - запрашивание уровня уверенности модели помогает выявлять галлюцинации **Комбинированные критерии** - учет как точности, так и уверенности модели улучшает результаты
Пример промпта с применением знаний из исследования

[======] Я задам тебе фактологический вопрос о [тема].

Пожалуйста, выполни следующие шаги:

Дай свой лучший ответ на вопрос Оцени свою уверенность в ответе по шкале от 1 до 10 Укажи, какие части ответа основаны на твоих точных знаниях, а какие могут быть менее достоверными Если уверенность ниже 7, предложи альтернативный ответ или укажи, что информация может быть неточной Мой вопрос: [фактологический вопрос] [======]

Объяснение применения исследования

Этот промпт использует ключевые принципы из исследования UAF:

Запрашивание самооценки уверенности - это аналог метрик неопределенности (Haloscope, перплексия), используемых в исследовании **Разделение ответа на части с разной уверенностью** - имитирует функцию SELECTOR из фреймворка UAF **Пороговое значение уверенности (7/10)** - реализует принцип фильтрации ненадежных ответов **Предложение альтернатив** - аналог функции FUSER, объединяющего результаты разных моделей Такой подход помогает снизить вероятность галлюцинаций, заставляя модель явно указывать свою неуверенность и предлагать альтернативы в случаях низкой достоверности.

№ 220. Генерация онтологий с использованием больших языковых моделей

Ссылка: https://arxiv.org/pdf/2503.05388

Рейтинг: 65
Адаптивность: 75

Ключевые выводы:

Исследование направлено на оценку потенциала больших языковых моделей (LLM) для автоматизированной разработки онтологий на основе пользовательских требований. Основные результаты показывают, что предложенные методы промптинга (Memoryless CQbyCQ и Ontogenia) превосходят существующие подходы и начинающих инженеров онтологий по качеству моделирования, причем модель OpenAI o1-preview с техникой Ontogenia демонстрирует наилучшие результаты.

Объяснение метода:

Исследование предлагает конкретные техники промптинга для генерации структурированных знаний и методы оценки качества. Хотя оно фокусируется на узкоспециализированной области онтологий, принципы структурированного промптинга, формулирования требований через вопросы и многомерной оценки качества применимы к широкому спектру задач взаимодействия с LLM. Требует некоторой адаптации для широкой аудитории.

Ключевые аспекты исследования: 1. **Разработка методик генерации онтологий с использованием LLM**: Авторы представляют и оценивают две техники промптинга для автоматизированной разработки онтологий: Memoryless CQbyCQ и Ontogenia.

Использование пользовательских историй и компетентностных вопросов: Исследование фокусируется на генерации онтологий OWL непосредственно из онтологических требований, описанных с помощью пользовательских историй и компетентностных вопросов (CQ).

Многомерная оценка качества: Авторы подчеркивают важность комплексной оценки, включающей структурные критерии и экспертную оценку, для определения качества и удобства использования сгенерированных онтологий.

Сравнительный анализ LLM: В исследовании сравнивается производительность трех LLM (GPT-4, OpenAI o1-preview и Llama 3) с использованием двух методик промптинга на эталонном наборе данных из десяти онтологий.

Выявление типичных ошибок и ограничений: Авторы анализируют общие ошибки и вариативность качества результатов при использовании LLM для создания

онтологий.

Дополнение: Для работы методов, описанных в исследовании, не требуется дообучение или специальное API. Основные концепции и подходы можно применить в стандартном чате LLM, хотя авторы для своих экспериментов использовали API для более систематической оценки и сравнения моделей.

Ключевые концепции и подходы, которые можно адаптировать для работы в стандартном чате:

Структурированный промптинг с пошаговым разбиением задачи: Разбиение сложной задачи на подзадачи (например, моделирование одного вопроса за раз) можно применить для любых задач структурирования знаний.

Метакогнитивный промптинг (Ontogenia): Пятиступенчатый процесс, где модель сначала анализирует требования, затем формирует решение, проверяет его и объясняет свои рассуждения. Этот подход можно использовать для улучшения качества ответов в любых сложных задачах.

Использование пользовательских историй и компетентностных вопросов: Формулирование требований в виде конкретных вопросов, на которые должно отвечать решение, помогает получить более структурированные и релевантные ответы.

Уменьшение контекстного окна (Memoryless CQbyCQ): Исследование показало, что удаление лишней информации из контекста может улучшить результаты, что применимо к любым взаимодействиям с LLM.

Многомерная оценка качества: Подход к оценке сгенерированного контента по нескольким критериям (структурная корректность, соответствие требованиям, отсутствие лишних элементов) может быть адаптирован для проверки любых результатов LLM.

Применяя эти концепции в стандартном чате, пользователи могут получить: - Более структурированные и логически последовательные ответы на сложные вопросы - Лучшее соответствие ответов исходным требованиям - Более систематический подход к проверке и улучшению качества сгенерированного контента - Уменьшение "галлюцинаций" и ошибок в ответах LLM

Prompt:

Применение исследования LLM для онтологий в промптах GPT ## Ключевые знания из исследования для промптов

Исследование показывает, что большие языковые модели (LLM) могут эффективно создавать онтологии с помощью специальных техник промптинга:

Техника Ontogenia - наиболее эффективный подход с моделью o1-preview
Memoryless CQbyCQ - хорошо работает для независимого моделирования
Метакогнитивный промптинг в сочетании с методологией экстремального дизайна
Осведомленность о типичных ошибках (множественные домены, неправильные обратные отношения) ## Пример промпта для создания онтологии

[======] # Задача: Разработка онтологии для [предметной области]

Контекст Я работаю над созданием онтологии для [описание проекта]. Мне нужно смоделировать следующие компетентностные вопросы (CQ):

[Компетентностный вопрос 1] [Компетентностный вопрос 2] [Компетентностный вопрос 3] ## Инструкции (техника Ontogenia) Пожалуйста, следуй структурированному подходу:

Интерпретация требований: Проанализируй каждый компетентностный вопрос и определи ключевые понятия и отношения.

Выбор шаблонов онтологического дизайна: Определи подходящие шаблоны для моделирования выявленных понятий.

Интеграция и моделирование:

Создай классы, свойства и отношения Избегай множественных доменов или диапазонов Правильно моделируй обратные отношения Используй корректные пространства имен

Проверка и рефлексия: Убедись, что онтология отвечает на все компетентностные вопросы и не содержит избыточных элементов.

Представь результат в формате OWL с аннотациями и комментариями. [======]

Как это работает

Данный промпт использует ключевые элементы из исследования:

Структурированный метакогнитивный подход (как в Ontogenia) - разбивает процесс на этапы интерпретации, выбора шаблонов, интеграции и проверки

Фокус на компетентностных вопросах - основной метод оценки качества онтологии в исследовании

Предотвращение типичных ошибок - явно указывает на проблемы, выявленные через OntOlogy Pitfall Scanner (OOPS!)

Баланс между полнотой и избыточностью - призывает проверить избыточные элементы, что было важным аспектом в оценке качества

Этот подход позволяет получить более качественные онтологии по сравнению со стандартными промптами, что подтверждается результатами исследования, где правильно моделировалось до 100% компетентностных вопросов.

№ 224. Генеративный искусственный интеллект: развивающаяся технология, растущее социальное воздействие и возможности для исследований в области информационных систем

Ссылка: https://arxiv.org/pdf/2503.05770

Рейтинг: 65
Адаптивность: 70

Ключевые выводы:

Основная цель исследования - изучить уникальные особенности генеративного искусственного интеллекта (GenAI), его эволюцию и потенциальное влияние на бизнес и общество с точки зрения информационных систем. Главные результаты: авторы разработали теоретическую основу для понимания GenAI как социотехнической системы, выявили три ключевые свойства GenAI (сильная эмерджентность, генеративная новизна, системные входы и выходы), и предложили обширную исследовательскую повестку для изучения GenAI в контексте информационных систем.

Объяснение метода:

Исследование предлагает ценную концептуальную основу для понимания GenAI как социотехнической системы с уникальными свойствами. Особенно полезны анализ "темной стороны" GenAI и системный взгляд на его возможности и ограничения. Однако высокий уровень абстракции и отсутствие конкретных практических рекомендаций снижают непосредственную применимость для широкой аудитории.

Ключевые аспекты исследования: 1. **Концептуальная основа GenAI как социотехнической системы**: Исследование предлагает теоретическую структуру для понимания генеративного ИИ с точки зрения системного подхода, рассматривая его как социотехническую систему с уникальными свойствами.

Три ключевых свойства GenAI: Авторы выделяют сильную эмерджентность (способность системы демонстрировать поведение, не выводимое напрямую из свойств компонентов), генеративную новизну (способность создавать как ожидаемые, так и неожиданные выходные данные) и системные входы/выходы (способность принимать и создавать целостные концептуальные системы).

Эволюция ИИ и переход к коннекционизму: Исследование прослеживает эволюцию ИИ от символизма к коннекционизму, объясняя, как это привело к появлению больших языковых моделей (LLM) и генеративного ИИ.

Исследовательская повестка и возможности: Авторы предлагают обширную исследовательскую повестку для информационных систем в контексте GenAI, охватывающую такие темы как влияние на производительность, сотрудничество человека и ИИ, этические проблемы и проектирование систем.

Темная сторона GenAI: Исследование анализирует потенциальные негативные последствия GenAI, включая нарушение прав интеллектуальной собственности, дезинформацию, эмоциональные манипуляции, галлюцинации и смещения, энергопотребление и непрозрачность.

Дополнение:

Методы и подходы для стандартного чата

Исследование не требует дообучения или API для применения его основных концепций. Хотя авторы обсуждают технические аспекты GenAI, основная ценность работы заключается в концептуальном понимании природы генеративного ИИ, которое может быть применено в стандартном чате без дополнительных технических инструментов.

Концепции и подходы, применимые в стандартном чате:

Понимание трех ключевых свойств GenAI: Сильная эмерджентность: Пользователи могут осознать, что LLM способны создавать ответы, которые не являются прямым следствием их обучения. Это помогает формулировать запросы, учитывая эту особенность. Генеративная новизна: Понимание, что LLM могут генерировать как ожидаемые, так и неожиданные ответы, помогает пользователям быть готовыми к разнообразным результатам и соответствующим образом адаптировать свои запросы. Системные входы/выходы: Осознание того, что LLM могут создавать целостные концептуальные системы (эссе, код, аргументы), позволяет пользователям запрашивать более сложные и структурированные результаты.

Критическое отношение к результатам:

Понимание проблем "галлюцинаций" и смещений помогает пользователям более критически относиться к результатам и верифицировать важную информацию. Осознание ограничений LLM в понимании контекста и смысла помогает формулировать запросы с учетом этих ограничений.

Системный подход к взаимодействию:

Рассмотрение взаимодействия с LLM как части более широкой социотехнической системы помогает пользователям интегрировать результаты в свои рабочие процессы. Понимание триангулярных отношений между пользователем, LLM и поисковыми системами позволяет эффективнее сочетать разные источники

информации.

Улучшение формулировок запросов:

Осознание важности промпт-инженерии и необходимости предоставления контекста для получения лучших результатов. Понимание, что LLM требуют более специфичной контекстуальной информации, чем человек, для точных ответов. Результаты от применения этих концепций: - Более реалистичные ожидания от взаимодействия с LLM - Улучшенные стратегии формулирования запросов - Более критическая и взвешенная оценка результатов - Лучшая интеграция LLM в более широкие рабочие процессы и информационные экосистемы

Prompt:

Использование знаний из исследования GenAI в промптах ## Ключевые концепции для применения в промптах

Исследование выделяет три фундаментальных свойства GenAI, которые можно использовать для создания более эффективных промптов:

Сильная эмерджентность - модели могут демонстрировать неожиданное поведение **Генеративная новизна** - способность создавать оригинальный контент **Системные входы/выходы** - работа с целостными результатами ## Пример промпта с использованием знаний из исследования

[======] Я хочу использовать твою способность к генеративной новизне и эмерджентности для решения бизнес-задачи.

Контекст: Я руководитель отдела маркетинга в компании, производящей экологичную бытовую химию. Нам нужно разработать новую стратегию продвижения, которая подчеркнет наше уникальное преимущество.

Инструкции: 1. Используя системный подход, проанализируй взаимосвязь между нашим продуктом, целевой аудиторией и рыночными тенденциями 2. Предложи 3 нестандартных маркетинговых стратегии, демонстрирующих генеративную новизну 3. Для каждой стратегии укажи возможные риски и способы их минимизации 4. Представь результат в виде структурированной таблицы с оценкой эффективности каждой стратегии

Дополнительные знания: Наша целевая аудитория - экологически сознательные потребители 25-45 лет, преимущественно женщины с высшим образованием и средним/высоким доходом. [======]

Объяснение эффективности

Этот промпт использует знания из исследования следующим образом:

Предоставляет системный контекст - учитывая, что GenAI работает как социотехническая система, промпт включает информацию о бизнес-контексте, целевой аудитории и специфике задачи

Использует предиктивную природу GenAI - промпт структурирован так, чтобы направить предсказательные способности модели в нужное русло, предоставляя достаточный контекст

Запрашивает генеративную новизну - прямо указывает на необходимость создания оригинальных стратегий, используя это свойство GenAI

Минимизирует риски галлюцинаций - запрашивает структурированный вывод и конкретные рекомендации, что снижает вероятность необоснованных утверждений

Применяет подход ICL (In-Context Learning) - предоставляет модели дополнительные знания о целевой аудитории для более точного ответа

Такой подход к составлению промптов, основанный на понимании фундаментальных свойств GenAI как социотехнической системы, позволяет получать более качественные, релевантные и практически применимые результаты.

№ 228. Раскрытие и причинное объяснение CoT: Причинная перспектива

Ссылка: https://arxiv.org/pdf/2502.18239

Рейтинг: 65
Адаптивность: 75

Ключевые выводы:

Исследование направлено на раскрытие механизма Chain of Thought (CoT) в больших языковых моделях (LLM) с точки зрения причинно-следственных связей. Авторы предлагают метод CauCoT (Causalized Chain of Thought), который делает рассуждения LLM не только правильными, но и понятными для человека, моделируя причинно-следственные связи между шагами рассуждений с помощью структурных причинных моделей (SCM).

Объяснение метода:

Исследование предлагает ценную концепцию о причинно-следственных связях в рассуждениях LLM. Практическую ценность имеют техника ролевых запросов для улучшения логики рассуждений, классификация типичных ошибок и понимание важности первого шага. Однако многие технические аспекты (SCM, CACE, FSCE) недоступны широкой аудитории без специальных знаний.

Ключевые аспекты исследования: 1. **Моделирование причинности в Chain of Thought (CoT)** - авторы используют структурные причинные модели (SCM) для выявления механизмов рассуждений в CoT, делая процесс более понятным и интерпретируемым.

Метрики оценки причинности - введены метрики "CoT Average Causal Effect" (CACE) и "First-Step Causal Effect" (FSCE) для количественной оценки причинных отношений между шагами рассуждений.

Алгоритм CauCoT - разработан метод "причинной каузализации" CoT с использованием ролевых запросов, который исправляет шаги, не имеющие причинных связей, обеспечивая как правильность, так и понятность всех шагов рассуждения.

Типология причинных ошибок - выявлены и классифицированы четыре типа причинных ошибок в CoT-рассуждениях: ошибки измерения причинности, коллайдер-ошибки, ошибки чувствительности и медиаторные ошибки.

Эмпирическая валидация - метод проверен на различных наборах данных и моделях, показав значительное улучшение способности LLM к рассуждению.

Дополнение: Для работы методов исследования в полном объеме действительно требуется доступ к API и возможность вмешательства в процесс генерации ответов. Однако многие концепции и подходы можно адаптировать для стандартного чата без необходимости дообучения или специального API.

Концепции и подходы, применимые в стандартном чате:

Ролевые запросы для улучшения рассуждений - пользователи могут просить LLM выступить в роли эксперта в определенной области и проверить логические связи в рассуждениях. Например: "Выступи в роли математика и проверь, логически ли связан каждый шаг твоих рассуждений с предыдущим".

Проверка причинно-следственных связей - пользователи могут запрашивать явное объяснение, как каждый шаг рассуждения связан с предыдущим и с исходным вопросом. Например: "Объясни, как каждый шаг твоего рассуждения причинно связан с предыдущим".

Фокус на первом шаге - понимая важность первого шага, пользователи могут запрашивать более тщательное обоснование начального этапа рассуждения. Например: "Прежде чем продолжить, убедись, что первый шаг твоего рассуждения имеет прямое отношение к вопросу".

Проверка на типичные причинные ошибки - пользователи могут просить модель проверить свой ответ на наличие типичных ошибок, описанных в исследовании. Например: "Проверь свой ответ на наличие коллайдер-ошибок, где ты неправильно оцениваешь влияние двух переменных".

Двухэтапный процесс рассуждения - сначала получение ответа, затем запрос на проверку причинных связей между шагами, аналогично процессу "рефайнинга" в исследовании.

Ожидаемые результаты от применения этих подходов: - Более логически связные и понятные рассуждения - Снижение количества логических ошибок в ответах - Повышение качества решения сложных задач, особенно математических и логических - Лучшее понимание пользователем процесса рассуждения LLM

Хотя эти адаптированные методы не будут столь же эффективны, как полная реализация CauCoT с доступом к API, они все равно могут значительно улучшить качество рассуждений в стандартном чате.

Prompt:

Применение причинного подхода CoT в промптах для GPT ## Ключевые идеи из исследования

Исследование CauCoT (Causalized Chain of Thought) показывает, что добавление

причинно-следственных связей между шагами рассуждений значительно улучшает качество ответов языковых моделей, особенно в сложных задачах.

Пример промпта с применением CauCoT

[======] Решите следующую математическую задачу, используя причинно-следственный подход:

Задача: Найдите все решения уравнения $2x^2 - 5x + 2 = 0$.

При решении следуйте этим инструкциям: 1. Разбейте решение на логические шаги 2. Для каждого шага явно укажите, почему он следует из предыдущего 3. Проверьте, что каждый шаг имеет причинную связь с предыдущим 4. Если заметите отсутствие причинной связи между шагами, вернитесь и исправьте рассуждение 5. В конце проверьте, что ваша цепочка рассуждений не содержит логических разрывов

Помните: каждый шаг должен быть причинно обоснован предыдущими шагами, а не просто следовать формальному алгоритму. [======]

Объяснение подхода

Этот промпт использует ключевые идеи CauCoT следующим образом:

Структурированное причинное рассуждение: Промпт требует разбить решение на шаги и явно указать причинно-следственные связи между ними.

Проверка причинности: Включена инструкция проверить причинные связи между шагами, что помогает избежать четырех типов причинных ошибок, выявленных в исследовании.

Итеративное исправление: Предлагается вернуться и исправить рассуждение при обнаружении отсутствия причинной связи, что соответствует алгоритму ролевых причинных запросов из исследования.

Финальная проверка: Завершающая проверка на логические разрывы помогает обеспечить целостность причинной цепочки.

Такой подход особенно эффективен для сложных задач логического рассуждения, где стандартный CoT может давать сбои из-за отсутствия явных причинно-следственных связей между шагами.

№ 232. Эффективность больших языковых моделей в написании формул сплавов

Ссылка: https://arxiv.org/pdf/2502.15441

Рейтинг: 65
Адаптивность: 70

Ключевые выводы:

Исследование оценивает эффективность использования больших языковых моделей (LLM) для написания формул в декларативном языке Alloy. Основная цель - определить, насколько хорошо LLM могут создавать спецификации Alloy тремя способами: из описаний на естественном языке, на основе существующих формул Alloy и путем заполнения скетчей (шаблонов с пропусками). Результаты показали, что LLM (ChatGPT и DeepSeek) успешно справляются с этими задачами, генерируя множество корректных и уникальных решений.

Объяснение метода:

Исследование демонстрирует способность LLM переводить естественный язык в формальные спецификации Alloy, генерировать эквивалентные формулы и заполнять шаблоны. Несмотря на специализированный характер Alloy, методы имеют более широкое применение и могут быть адаптированы для других языков, упрощая работу с формальными методами для неспециалистов.

Ключевые аспекты исследования: 1. **Использование LLM для написания формул Alloy из описаний на естественном языке** - исследование показывает, как ChatGPT и DeepSeek могут создавать корректные формальные спецификации на языке Alloy на основе описаний на английском языке.

Создание эквивалентных формул Alloy на основе существующих - LLM способны генерировать альтернативные, но логически эквивалентные формулы для одних и тех же свойств, демонстрируя понимание семантики языка.

Заполнение шаблонов (sketching) Alloy - LLM успешно заполняют пробелы в частично определенных формулах Alloy без необходимости предоставления тестовых примеров.

Экспериментальное исследование на 11 базовых свойствах - оценка эффективности LLM на задачах, связанных с графами и бинарными отношениями, показывает высокую точность даже без специальной настройки моделей.

Генерация множества уникальных решений - LLM способны создавать до 20 различных, но эквивалентных формулировок одного и того же свойства, демонстрируя глубокое понимание логики.

Дополнение: Действительно ли для работы методов этого исследования требуется дообучение или API? Или методы и подходы можно применить в стандартном чате, а ученые лишь для своего удобства использовали расширенные техники?

Исследование явно указывает, что не использовалось никакого дообучения моделей или специальных API. Авторы пишут: "Мы используем LLM напрямую в том виде, в котором они доступны для общественного пользования. В частности, мы их не дообучаем." (стр. 9)

Все методы, показанные в исследовании, могут быть применены в стандартном чате с LLM. Основные концепции, которые можно использовать в обычном чате:

Перевод с естественного языка на формальный язык - пользователи могут просить LLM преобразовать описание на естественном языке в формальную спецификацию любого вида, не только Alloy.

Генерация эквивалентных формулировок - пользователи могут просить LLM предложить альтернативные способы выражения одной и той же идеи, что полезно для обучения и понимания различных подходов.

Заполнение шаблонов - пользователи могут предоставить частичные спецификации или код с пробелами и попросить LLM заполнить их, что особенно полезно, когда у пользователя есть только общее представление о структуре решения.

Итеративное уточнение - исследование показывает, что когда LLM делает синтаксическую ошибку, простое указание на ошибку и просьба попробовать снова часто приводит к успешному решению.

Ожидаемые результаты от применения этих концепций: - Упрощение работы с формальными методами для неспециалистов - Ускорение процесса создания спецификаций - Расширение понимания различных способов выражения одних и тех же понятий - Возможность итеративного улучшения спецификаций через диалог с LLM

Важно отметить, что для проверки корректности сгенерированных формальных спецификаций в исследовании использовался Alloy Analyzer, но это не является обязательным для применения самих методов взаимодействия с LLM.

Prompt:

Применение исследования об эффективности LLM в написании формул Alloy для создания промптов ## Ключевые аспекты исследования для использования в промптах

Исследование показало, что большие языковые модели (LLM) успешно справляются с: 1. Синтезом формул из описаний на естественном языке 2. Созданием эквивалентных формул на основе существующих 3. Заполнением шаблонов с пропусками (скетчей)

Пример промпта для генерации формул Alloy

[======] # Задача создания формулы Alloy

Контекст Я работаю над формальной спецификацией системы с использованием языка Alloy. Мне нужно создать формулу, которая корректно описывает следующее свойство:

[Описание свойства на естественном языке, например: "Граф не содержит циклов"]

Инструкции 1. Создай 5 различных корректных формул Alloy, которые выражают указанное свойство. 2. Для каждой формулы: - Объясни ее логику - Укажи, какие конструкции языка Alloy используются - Отметь преимущества и недостатки данной формулировки 3. Формулы должны быть синтаксически корректными и проверяемыми анализатором Alloy.

Дополнительные требования - Используй разнообразные подходы к формализации свойства - Старайся создавать формулы разной сложности и с разными языковыми конструкциями Alloy [======]

Объяснение эффективности

Этот промпт работает эффективно, потому что:

Использует доказанную способность LLM генерировать корректные формулы Alloy из описаний на естественном языке (согласно исследованию, модели могут создавать до 10+ корректных вариантов)

Запрашивает множество решений - исследование показало, что LLM способны генерировать разнообразные уникальные формулы для одного свойства

Структурирует запрос с четким контекстом и инструкциями, что помогает модели сфокусироваться на задаче формализации

Требует объяснений к каждой формуле, что использует способность LLM не только генерировать код, но и объяснять его, что особенно полезно для обучения новичков (одно из практических применений, указанных в исследовании)

Промпт можно адаптировать для других задач из исследования - например, для создания эквивалентных формул на основе существующей формулы или для заполнения шаблонов с пропусками.

№ 236. DeepRAG: Поэтапное мышление при извлечении для крупных языковых моделей

Ссылка: https://arxiv.org/pdf/2502.01142

Рейтинг: 65
Адаптивность: 75

Ключевые выводы:

Исследование представляет DeepRAG - новую структуру для улучшения способности больших языковых моделей (LLM) к рассуждению с помощью поиска информации. Основная цель - моделирование процесса поиска как марковского процесса принятия решений, что позволяет LLM стратегически и адаптивно определять, когда использовать внешние знания, а когда полагаться на параметрические знания. Результаты показывают, что DeepRAG улучшает точность ответов на 21,99% при одновременном повышении эффективности поиска.

Объяснение метода:

DeepRAG предлагает ценную методологию декомпозиции сложных вопросов на подзапросы и определения необходимости внешнего поиска. Хотя полная техническая реализация недоступна обычным пользователям, концептуальные принципы могут быть адаптированы для более эффективного взаимодействия с LLM через структурированные запросы и пошаговое рассуждение.

Ключевые аспекты исследования: 1. **DeepRAG: моделирование процесса как MDP** - исследование представляет подход, который моделирует процесс поиска и использования внешней информации как марковский процесс принятия решений (MDP), что позволяет динамически определять, когда требуется обращение к внешним источникам данных.

Двухкомпонентная структура системы - DeepRAG включает две ключевые составляющие: "retrieval narrative" (структурированный поток поисковых запросов) и "atomic decisions" (решения о необходимости поиска для каждого подзапроса), что обеспечивает стратегический и адаптивный подход к поиску.

Метод бинарного дерева поиска - для каждого подзапроса система строит бинарное дерево, исследуя два возможных пути: использование параметрических знаний модели или обращение к внешней базе знаний.

Двухэтапное обучение модели - сначала применяется имитационное обучение на синтезированных данных, затем используется "chain of calibration" для улучшения понимания моделью своих границ знаний.

Значительное улучшение точности и эффективности - эксперименты показали

повышение точности ответов на 21-99% при сокращении количества обращений к внешним источникам по сравнению с другими методами.

Дополнение:

Применимость методов в стандартном чате

Хотя в исследовании используется дообучение модели и специализированное API для реализации полной системы DeepRAG, многие концептуальные подходы могут быть адаптированы для использования в стандартном чате с LLM без технических модификаций:

Структурированная декомпозиция вопросов Пользователи могут вручную разбивать сложные вопросы на последовательность подзапросов Для каждого подзапроса можно получать промежуточный ответ перед переходом к следующему шагу

Осознанное использование внешней информации

Пользователи могут самостоятельно решать, когда запрашивать модель о поиске дополнительной информации Можно явно указывать модели, когда ответ должен основываться на её параметрических знаниях

Итеративное построение ответа

Использование промежуточных ответов как основы для формулирования следующих подзапросов Постепенное построение полного ответа на основе собранных промежуточных результатов Ожидаемые результаты от применения этих концепций: - Повышение точности ответов на сложные вопросы - Снижение вероятности галлюцинаций модели - Более структурированное и прозрачное рассуждение - Лучшее понимание пользователем процесса формирования ответа

Эти адаптированные подходы не требуют дообучения или специального API, но могут значительно улучшить качество взаимодействия с LLM в стандартном чате.

Prompt:

Применение DeepRAG в промптах для GPT ## Ключевые принципы DeepRAG

DeepRAG представляет структуру для улучшения способности языковых моделей к рассуждению с использованием внешней информации через: - Стратегическое определение, когда использовать внешние знания - Декомпозицию сложных запросов на подзапросы - Оптимизацию поисковых операций

Пример промпта с применением принципов DeepRAG

[======] # Запрос с применением DeepRAG-подхода

Контекст задачи Я исследую влияние изменения климата на миграцию видов в экосистемах коралловых рифов.

Структурированный подход (по DeepRAG) 1. Сначала определи, какие аспекты этой темы ты уже знаешь достаточно хорошо, а для каких потребуется дополнительная информация. 2. Декомпозируй основной вопрос на следующие подвопросы: - Какие ключевые механизмы влияния изменения климата на коралловые рифы? - Какие виды наиболее чувствительны к этим изменениям? - Какие существуют паттерны миграции в ответ на эти изменения? 3. Для каждого подвопроса: - Сначала ответь на основе своих параметрических знаний - Четко обозначь, где твои знания могут быть неполными или устаревшими - Предложи, какие конкретные внешние источники могли бы дополнить твой ответ

Формат ответа - Используй дерево рассуждений, четко показывая связи между подвопросами - В финальном ответе синтезируй информацию из всех подвопросов - Укажи степень уверенности в различных частях ответа [======]

Как работают принципы DeepRAG в этом промпте

Структурированное повествование поиска: Промпт декомпозирует сложный запрос на конкретные подзапросы, что позволяет модели более целенаправленно использовать свои знания.

Атомарные решения: Модель должна явно определить, для каких аспектов она имеет достаточно знаний, а для каких требуется внешняя информация.

Бинарный поиск по дереву: Промпт побуждает модель исследовать различные пути рассуждения, выбирая оптимальные на основе имеющихся знаний.

Калибровка границ знаний: Требование указать степень уверенности и потенциальные пробелы в знаниях помогает модели лучше осознавать границы своих возможностей.

Такой подход позволяет получить более глубокие, структурированные и обоснованные ответы, с четким разграничением между параметрическими знаниями модели и областями, где требуется дополнительная информация.

№ 240. Изучение понимания кода в научном программировании: предварительные выводы от исследователей

Ссылка: https://arxiv.org/pdf/2501.10037

Рейтинг: 65
Адаптивность: 75

Ключевые выводы:

Исследование направлено на изучение понимания кода в научном программировании путем опроса 57 ученых-исследователей из различных дисциплин. Основные результаты показывают, что большинство ученых осваивают программирование самостоятельно или на рабочем месте, при этом 57.9% не имеют формального обучения написанию читаемого кода. Несмотря на признание важности читаемости кода для научной воспроизводимости, исследователи сталкиваются с проблемами недостаточной документации и плохих соглашений об именовании, а также наблюдается тенденция к использованию больших языковых моделей для улучшения качества кода.

Объяснение метода:

Исследование выявляет конкретные проблемы с читаемостью кода (недостаточное комментирование, плохое именование, неудачная структура), которые пользователи могут учитывать при формулировании запросов к LLM и оценке результатов. Тенденция использования LLM для улучшения кода подтверждает ценность этого подхода для широкой аудитории.

Ключевые аспекты исследования: 1. **Образовательный фон и практики программирования научных исследователей**: Исследование показало, что большинство ученых осваивают программирование самостоятельно или в процессе работы, а 57.9% не имеют формального обучения написанию читаемого кода. Основные языки - Python и R.

Проблемы понимания научного кода: Основные трудности включают недостаточное комментирование (44 участника), отсутствие документации (33), плохое наименование функций/переменных (31), и неудачную организацию структуры проекта (31).

Проблемы с именами идентификаторов: Наиболее частые проблемы - слишком короткие или непонятные имена (40), несогласованные соглашения об именовании (30), имена, не отражающие назначение (29).

Использование инструментов для улучшения кода: 49.12% участников никогда

не используют автоматизированные инструменты для улучшения качества кода. Среди использующих, многие обращаются к ИИ и LLM (особенно ChatGPT и Claude).

Важность читаемости для воспроизводимости: Все участники признают важность читаемости кода для обеспечения воспроизводимых научных результатов, 83.76% считают это очень или чрезвычайно важным.

Дополнение: Исследование не требует дообучения или API для применения его выводов в стандартных чатах с LLM. Основные концепции и подходы исследования могут быть непосредственно использованы обычными пользователями в стандартных чатах.

Ключевые концепции, применимые в стандартном чате:

Чеклист типичных проблем с кодом: Пользователи могут использовать выявленные в исследовании проблемы (недостаточное комментирование, плохое именование, неудачная структура) как чеклист при запросе LLM проверить или улучшить их код. Например: "Проверь мой код на наличие следующих проблем: недостаточное комментирование, непонятные имена переменных, несогласованные соглашения об именовании."

Улучшение именования: Пользователи могут конкретно запрашивать улучшение именования в своем коде, указывая на проблемы, выявленные в исследовании: "Переименуй переменные и функции, избегая слишком коротких имен, общих терминов и несогласованных стилей."

Структурирование документации: Исследование показывает важность документации для понимания кода. Пользователи могут запрашивать: "Добавь к коду необходимые комментарии и создай README, объясняющий структуру и назначение программы."

Критическая оценка генерируемого кода: Понимая типичные проблемы с кодом, пользователи могут более критично оценивать код, сгенерированный LLM, и запрашивать конкретные улучшения.

Результаты от применения этих концепций: - Более читаемый и поддерживаемый код - Лучшее понимание собственного кода и кода, сгенерированного LLM - Более эффективное взаимодействие с LLM при работе с кодом - Повышение воспроизводимости результатов научных исследований - Сокращение времени на понимание и отладку кода

Prompt:

Применение Результатов Исследования о Понимании Кода в Научном Программировании для Промтов GPT ## Ключевые инсайты для использования в промтах

Исследование предоставляет ценные данные о том, как ученые работают с кодом и с какими проблемами сталкиваются. Эту информацию можно эффективно использовать при составлении промтов для GPT, особенно когда требуется помощь с научным программированием.

Пример промта

[======] Я исследователь в области [область науки], использующий Python для анализа данных. У меня нет формального образования в программировании, как и у 57.9% ученых согласно исследованиям. Помоги мне переработать следующий фрагмент кода с учетом лучших практик:

[код]

Пожалуйста: 1. Добавь подробные комментарии, так как недостаточная документация - главная проблема понимания научного кода 2. Улучши именование переменных для большей ясности 3. Реорганизуй код в логические модули 4. Объясни изменения, которые ты внес, простым языком 5. Предложи документацию, которую стоит добавить в README проекта [======]

Как работают знания из исследования в этом промте

Учет образовательного фона: Промт учитывает, что большинство ученых (57.9%) не имеют формального образования в написании читаемого кода, что помогает GPT адаптировать объяснения.

Фокус на главных проблемах: Промт направлен на решение основных проблем, выявленных в исследовании:

Недостаточность комментариев (16.18%) Плохая документация (12.13%) Неудачные соглашения об именовании

Практическое применение: Запрос включает конкретные действия из раздела "Практические применения" исследования:

Улучшение комментариев и документации Внедрение осмысленных соглашений об именовании Модульная организация кода

Учет важности воспроизводимости: Промт косвенно затрагивает вопрос научной воспроизводимости, который 83.76% ученых считают важным или чрезвычайно важным.

Такой подход к составлению промтов помогает получить от GPT более релевантную помощь, адаптированную к реальным потребностям научного сообщества.

№ 244. От диагностики суб-способностей к генерации, согласованной с человеком: преодоление разрыва для контроля длины текста с помощью MARKERGEN

Ссылка: https://arxiv.org/pdf/2502.13544

Рейтинг: 65
Адаптивность: 85

Ключевые выводы:

Исследование направлено на улучшение способности больших языковых моделей (LLM) контролировать длину генерируемого текста. Авторы выявили, что основные проблемы LLM в этой области связаны с фундаментальными недостатками в распознавании и подсчете единиц длины, а также в согласовании семантического содержания с ограничениями длины. Предложенный метод MARKERGEN значительно улучшает контроль длины текста, снижая ошибки на 12-57% по сравнению с базовыми методами.

Объяснение метода:

Исследование представляет трехэтапный подход к контролю длины текста (планирование, генерация, корректировка), который может быть адаптирован пользователями через промпты. Оно дает понимание причин ошибок в контроле длины и предлагает концептуальные решения. Однако полная реализация MARKERGEN требует технических знаний, что ограничивает прямую применимость для обычных пользователей.

Ключевые аспекты исследования: 1. **Декомпозиция способностей контроля длины текста (LCTG)** - исследование выделяет и анализирует ключевые подспособности LLM при генерации текста заданной длины: распознавание единиц длины, подсчет, планирование и выравнивание.

MARKERGEN - разработан плагин-метод для улучшения контроля длины текста, который интегрирует внешние инструменты для точного подсчета слов и динамически вставляет маркеры длины в процессе генерации.

Трехэтапная схема генерации - предложен подход, разделяющий планирование, семантическое моделирование и выравнивание длины, что позволяет сохранить качество контента при соблюдении ограничений длины.

Стратегия вставки маркеров с убывающим интервалом - метод динамического размещения маркеров длины, обеспечивающий баланс между семантическим

моделированием и контролем длины.

Экспериментальная валидация - подтверждена эффективность метода на разных моделях, задачах и языках с улучшением точности длины на 12-57% по сравнению с базовыми методами.

Дополнение:

Возможность применения без дообучения и API

Полная реализация MARKERGEN требует доступа к API или дообучения модели, поскольку включает интеграцию внешних инструментов для подсчета слов и вставку маркеров длины в процесс генерации. Однако ключевые концепции исследования могут быть адаптированы для использования в стандартном чате:

Трехэтапный процесс генерации: Планирование: Можно попросить модель создать план с указанием количества слов для каждой части текста Генерация: Создание основного контента согласно плану Корректировка: Анализ и исправление для соответствия ограничениям длины

Явное отслеживание длины:

Можно попросить модель периодически подсчитывать слова в генерируемом тексте Использовать стратегию "убывающего интервала" - более редкие проверки в начале и более частые ближе к целевой длине

Разделение семантического моделирования и контроля длины:

Сначала фокус на качестве контента Затем на соответствии ограничениям длины Применение этих подходов в стандартном чате позволит достичь: - Более точного соответствия заданным ограничениям длины - Сохранения качества контента при соблюдении ограничений - Улучшенной структуризации длинных текстов

Prompt:

Применение исследования MARKERGEN в промтах для GPT ## Ключевые идеи исследования для промптов

Исследование MARKERGEN показывает, что языковые модели испытывают трудности с контролем длины текста из-за проблем с распознаванием и подсчетом единиц длины, а также согласованием семантического содержания с ограничениями длины. Метод MARKERGEN решает эти проблемы через:

Интеграцию внешних инструментов для точного подсчета Динамическую вставку маркеров длины с убывающими интервалами Трехэтапную схему генерации (планирование => семантика => контроль длины) ## Пример промпта с применением MARKERGEN

[======] # Запрос на написание статьи о квантовых компьютерах (500 слов)

Инструкции по генерации:

Этап 1: Планирование Сначала создай план статьи с распределением слов по разделам: - Введение: ~70 слов - История квантовых вычислений: ~100 слов - Принципы работы: ~150 слов - Текущие достижения: ~100 слов - Перспективы и вызовы: ~80 слов

Этап 2: Генерация с маркерами Напиши статью согласно плану, вставляя маркеры слов с убывающими интервалами: - В начале: [0 слов] - После введения: [~70 слов] - Через каждые ~100 слов в основной части: [~170 слов], [~270 слов], [~370 слов] - На последних 100 словах используй более частые маркеры: [~420 слов], [~460 слов], [~490 слов] - В конце: [500 слов]

Этап 3: Проверка и корректировка После завершения черновика проверь, соответствует ли текст ограничению в 500 слов. При необходимости: - Если текст длиннее, сократи наименее важные детали, сохраняя ключевые идеи - Если текст короче, добавь релевантные детали в разделы с наибольшим потенциалом для расширения

Пожалуйста, убедись, что статья сохраняет высокое качество и логическую связность, при этом точно соответствуя ограничению в 500 слов. [======]

Как работают знания из исследования в этом промпте

Декомпозиция на подзадачи: Промпт разделяет задачу на этапы планирования, генерации и корректировки, что соответствует трехэтапной схеме MARKERGEN.

Планирование с распределением слов: Заранее определяется структура текста с указанием количества слов для каждого раздела, что помогает модели лучше планировать содержание.

Динамические маркеры: Промпт включает систему маркеров с убывающими интервалами — более редкие в начале (для сохранения семантической целостности) и более частые в конце (для точного контроля длины).

Явные инструкции по корректировке: Промпт содержит указания по проверке и корректировке текста, компенсируя неспособность модели точно подсчитывать слова.

Такой подход значительно повышает точность соблюдения ограничений по длине при сохранении высокого качества содержания.

№ 248. Соединение исследований HCI и ИИ для оценки разговорных помощников в области программной инженерии

Ссылка: https://arxiv.org/pdf/2502.07956

Рейтинг: 65
Адаптивность: 75

Ключевые выводы:

Исследование направлено на разработку методов автоматической оценки LLM-ассистентов для разработки программного обеспечения (SE) с учетом человеческого фактора. Авторы предлагают объединить подходы из областей взаимодействия человека с компьютером (HCI) и искусственного интеллекта (AI) для создания комплексной системы оценки, которая сочетает симулированных пользователей и подход 'LLM as a judge'.

Объяснение метода:

Исследование предлагает ценные концепции, которые могут быть адаптированы для повседневного использования LLM: "LLM как судья" для оценки ответов, учет разнообразия пользователей, многоходовые взаимодействия и критическое отношение к "эталонным" ответам. Хотя полная реализация методологии требует технических навыков, общие принципы доступны широкой аудитории.

Ключевые аспекты исследования: 1. **Интеграция методов HCI и AI для оценки LLM-ассистентов**: Исследование предлагает объединить подходы из областей взаимодействия человека с компьютером (HCI) и искусственного интеллекта (AI) для автоматической оценки разговорных LLM-ассистентов в сфере разработки ПО.

Симулированные пользователи: Предлагается использовать LLM для создания симулированных пользователей, которые могут реалистично взаимодействовать с ассистентом, генерировать качественную обратную связь и выявлять проблемы с инклюзивностью.

LLM как судья: Подход, при котором LLM используется для оценки ответов других LLM по заданным критериям, что позволяет получать количественные метрики без необходимости проведения дорогостоящих исследований с участием людей.

Персоны и разнообразие: Важность создания репрезентативных персон для симуляции разнообразных пользователей, что помогает выявлять "баги инклюзивности" — проблемы, которые возникают только у определенных групп пользователей.

Ограничения существующих методов оценки: Критика традиционных методов оценки LLM-ассистентов, основанных на сравнении с эталонными ответами, которые не отражают разнообразие возможных действительных ответов и не учитывают многоходовый характер реальных взаимодействий.

Дополнение: Исследование не требует дообучения или специального API для применения основных концепций в стандартном чате. Хотя авторы используют более продвинутые технические подходы для систематической оценки, ключевые идеи могут быть адаптированы для использования в обычном диалоге с LLM.

Вот концепции, которые можно применить в стандартном чате:

LLM как судья: Пользователь может попросить LLM оценить свой предыдущий ответ по определенным критериям или сравнить несколько подходов к решению задачи. Например: "Я получил от тебя два решения моей задачи программирования. Оцени их по критериям эффективности, читаемости и следования лучшим практикам. Какое решение лучше и почему?"

Персонализация запросов: Пользователь может указать свой уровень опыта или предпочтительный стиль объяснения. Например: "Объясни мне, как работает рекурсия, как если бы я был начинающим программистом, который только изучает основы."

Многоходовое взаимодействие: Вместо попыток получить исчерпывающий ответ в одном запросе, пользователь может вести инкрементальный диалог, уточняя детали и постепенно двигаясь к решению. Это соответствует естественному процессу человеческого общения.

Разнообразие перспектив: Пользователь может запросить альтернативные точки зрения на проблему. Например: "Как бы эту задачу решил опытный разработчик Python? А как бы к ней подошел специалист по JavaScript?"

Применяя эти концепции, пользователь может получить: - Более точные и релевантные ответы, адаптированные к своему уровню подготовки - Многогранное понимание проблемы через различные перспективы - Более критическое отношение к ответам LLM - Улучшенное поэтапное решение сложных задач

Эти подходы не требуют технической реализации и могут быть использованы любым пользователем в рамках обычного диалогового интерфейса.

Prompt:

Использование знаний из исследования в промптах для GPT ## Ключевые инсайты исследования для промптов

Исследование предлагает комбинированный подход к оценке LLM-ассистентов,

объединяющий симулированных пользователей и LLM-судей. Эти знания можно применить для создания более эффективных промптов.

Пример промпта с использованием методологии исследования

[======] Я разрабатываю помощника на базе GPT для junior-разработчиков, который помогает с задачами по JavaScript. Помоги мне улучшить мой промпт, используя следующий подход:

Создай 3 различные персоны пользователей (начинающий, средний уровень, с опытом в других языках) с разными потребностями и стилями взаимодействия.

Для каждой персоны смоделируй диалог с моим ассистентом, используя следующий базовый промпт: "Ты помощник по JavaScript. Твоя задача - объяснять концепции, помогать с отладкой и предлагать решения. Старайся давать понятные объяснения с примерами кода."

Оцени эффективность этого промпта по следующим критериям:

Точность предоставляемой информации (1-10) Понятность объяснений для конкретной персоны (1-10) Полезность примеров кода (1-10) Способность адаптироваться к уровню пользователя (1-10)

Предложи улучшения промпта, основываясь на результатах симуляции, добавив:

Конкретные инструкции по адаптации к разным уровням пользователей Структуру для предоставления ответов Стратегии для более эффективного объяснения сложных концепций [======] ## Как работают знания из исследования в этом промпте

Репрезентативные персоны — промпт использует идею создания различных профилей пользователей для обеспечения инклюзивности и учета разнообразия аудитории.

Симуляция диалогов — применяется метод генерации взаимодействий между ассистентом и симулированным пользователем для тестирования эффективности промпта.

Количественная оценка — внедрена система оценки по конкретным метрикам (по шкале 1-10), что соответствует подходу "LLM as a judge" из исследования.

Качественная обратная связь — запрашиваются конкретные улучшения на основе выявленных проблем, что позволяет получить качественные выводы.

Такой подход позволяет итеративно улучшать промпты, основываясь на симулированном пользовательском опыте и структурированной оценке, что делает разработку более эффективной, не требуя постоянных реальных пользовательских тестов.

№ 252. Обучение ИИ обработке исключений: Управляемая тонкая настройка с учетом человеческого суждения

Ссылка: https://arxiv.org/pdf/2503.02976

Рейтинг: 65
Адаптивность: 75

Ключевые выводы:

Исследование направлено на изучение способности больших языковых моделей (LLM) обрабатывать исключения в процессе принятия решений. Основные результаты показывают, что LLM, даже самые продвинутые, значительно отклоняются от человеческих суждений, строго придерживаясь политик даже когда это непрактично или контрпродуктивно. Обучение с учителем (supervised fine-tuning) с использованием человеческих объяснений, а не просто бинарных ответов, значительно улучшает способность моделей принимать решения, соответствующие человеческим суждениям.

Объяснение метода:

Исследование выявляет критическое ограничение LLM (чрезмерную приверженность правилам) и предлагает практические решения. Особенно ценны выводы о важности объяснений и цепочек рассуждений. Пользователи могут применять эти принципы для получения более гибких ответов, формулируя запросы, учитывающие потребность в исключениях. Часть методов требует технических навыков, но концептуальное понимание доступно всем.

Ключевые аспекты исследования: 1. **Проблема следования правилам**: Исследование показывает, что LLM слишком строго придерживаются заданных правил и политик, отказываясь делать исключения даже в случаях, когда люди считают их разумными (например, покупка муки за $10.01 при ограничении в $10).

Методы улучшения гибкости: Авторы тестируют три подхода для решения этой проблемы: этические фреймворки, цепочки рассуждений (chain-of-thought) и дообучение на человеческих примерах (supervised fine-tuning).

Supervised Fine-Tuning (SFT): Дообучение на человеческих объяснениях (а не просто на бинарных ответах "да/нет") значительно улучшает способность модели принимать гибкие решения, более согласованные с человеческими.

Трансфер обучения: Модели, дообученные на объяснениях в одном сценарии, демонстрируют улучшенную способность принимать человекоподобные решения в совершенно новых ситуациях.

Важность объяснений: Для эффективного обучения LLM принятию исключений критически важно использовать полные объяснения, а не только бинарные решения.

Дополнение: Исследование демонстрирует, что для достижения наилучших результатов действительно требуется дообучение (SFT) моделей, особенно с использованием полных объяснений, а не просто бинарных ответов. Однако некоторые подходы и концепции можно адаптировать для использования в стандартном чате без дообучения:

Цепочки рассуждений (Chain of Thought): Исследование показало, что этот метод дает небольшое, но заметное улучшение. Пользователи могут просить модель рассуждать пошагово, анализировать исключение, применять политику и только потом делать вывод. Например: "Прежде чем ответить, рассмотри все аспекты ситуации, включая последствия строгого следования правилу и последствия исключения".

Явное указание на возможность исключений: Пользователи могут включать в запросы явное разрешение на исключения: "Пожалуйста, учитывай, что иногда разумно делать исключения из правил, если строгое следование им приводит к нежелательным последствиям".

Запрос на баланс между буквальным следованием и гибкостью: "Рассмотри как буквальное следование правилу, так и его дух/намерение. Что в данном случае важнее?"

Запрос на оценку пропорциональности: "Оцени, насколько серьезно нарушение правила по сравнению с последствиями его строгого соблюдения".

Применяя эти подходы, пользователи могут получить более гибкие и человекоподобные ответы в стандартных чатах. Результаты не будут столь же хороши, как при полном дообучении с объяснениями, но могут значительно улучшить работу с LLM в ситуациях, требующих разумных исключений из правил.

Анализ практической применимости: **1. Проблема следования правилам** - **Прямая применимость**: Пользователи могут осознать, что LLM склонны к чрезмерно буквальному следованию инструкциям, и соответственно формулировать запросы с учетом потенциальной необходимости исключений. - **Концептуальная ценность**: Высокая. Понимание этого ограничения помогает пользователям ожидать и обходить негибкость LLM. - **Потенциал для адаптации**: Пользователи могут разработать стратегии формулирования запросов, которые заранее учитывают потребность в исключениях.

2. Методы улучшения гибкости - **Прямая применимость**: Средняя. Цепочки рассуждений могут быть непосредственно применены пользователями в запросах. - **Концептуальная ценность**: Высокая. Понимание, что дополнительные рассуждения улучшают гибкость LLM, полезно для формулирования запросов. -

Потенциал для адаптации: Пользователи могут применять элементы цепочек рассуждений в своих запросах, чтобы получать более гибкие ответы.

3. Supervised Fine-Tuning (SFT) - **Прямая применимость**: Низкая для обычных пользователей, так как требует технических навыков и доступа к API. - **Концептуальная ценность**: Средняя. Понимание, что модели можно улучшать через дообучение, полезно для общего понимания возможностей LLM. - **Потенциал для адаптации**: Организации могут применять этот подход для настройки своих LLM-решений.

4. Трансфер обучения - **Прямая применимость**: Низкая для обычных пользователей, высокая для разработчиков. - **Концептуальная ценность**: Средняя. Понимание, что модели могут переносить знания между доменами, полезно для понимания потенциала LLM. - **Потенциал для адаптации**: Организации могут использовать этот принцип для эффективного дообучения моделей на меньшем количестве примеров.

5. Важность объяснений - **Прямая применимость**: Средняя. Пользователи могут предоставлять объяснения при запросах для получения более качественных ответов. - **Концептуальная ценность**: Высокая. Понимание, что объяснения важнее бинарных решений, полезно для взаимодействия с LLM. - **Потенциал для адаптации**: Пользователи могут включать объяснения в свои запросы, чтобы лучше направлять модель.

Сводная оценка полезности: На основе анализа определяю общую оценку полезности исследования: **65**.

Исследование имеет высокую полезность для широкой аудитории пользователей LLM. Оно выявляет фундаментальное ограничение в работе LLM (чрезмерная приверженность правилам) и предлагает практические методы для его преодоления. Особенно ценны выводы о важности объяснений при формулировании запросов и использовании цепочек рассуждений.

Контраргументы к оценке:

Почему оценка могла бы быть выше: Исследование раскрывает фундаментальное ограничение LLM и предлагает конкретные методы его преодоления, которые могут значительно улучшить пользовательский опыт. Понимание этих ограничений критически важно для эффективного использования LLM.

Почему оценка могла бы быть ниже: Некоторые предложенные методы (SFT) требуют технических навыков и доступа к API, что делает их недоступными для большинства обычных пользователей. Многие выводы имеют больше концептуальную, чем непосредственную практическую ценность.

После рассмотрения контраргументов, я считаю, что оценка 65 является справедливой, так как исследование действительно предоставляет ценные

концепции и некоторые практические методы, которые могут быть применены широкой аудиторией с небольшой адаптацией.

Оценка была дана за: 1. Выявление важного ограничения LLM, которое влияет на повседневное использование 2. Предоставление практических методов (например, цепочки рассуждений), которые пользователи могут непосредственно применять 3. Демонстрацию важности объяснений при взаимодействии с LLM 4. Концептуальные знания, которые помогают пользователям лучше понимать возможности и ограничения LLM

Уверенность в оценке: Очень сильная. Исследование содержит четкие, воспроизводимые результаты с конкретными примерами, которые демонстрируют проблему и эффективность различных решений. Методология хорошо описана, а выводы имеют как теоретическую, так и практическую ценность для пользователей LLM.

Оценка адаптивности: Адаптивность: **75**

Исследование демонстрирует высокий потенциал для адаптации его принципов в повседневном использовании LLM:

Основные концепции исследования (необходимость гибкости при принятии решений, важность объяснений, использование цепочек рассуждений) могут быть легко адаптированы для обычного взаимодействия с LLM.

Пользователи могут извлечь полезные идеи о формулировании запросов, которые учитывают потенциальную необходимость исключений.

Метод цепочек рассуждений может быть непосредственно применен в обычных запросах для получения более гибких ответов.

Концепция важности объяснений может быть использована для более эффективного направления модели при решении сложных задач.

Понимание, что LLM могут проявлять чрезмерную жесткость, позволяет пользователям заранее адаптировать свои запросы, чтобы предотвратить это ограничение.

Хотя некоторые методы (SFT) требуют технических навыков, основные концепции и принципы исследования могут быть широко адаптированы обычными пользователями.

|| <Оценка: 65> || <Объяснение: Исследование выявляет критическое ограничение LLM (чрезмерную приверженность правилам) и предлагает практические решения. Особенно ценны выводы о важности объяснений и цепочек рассуждений. Пользователи могут применять эти принципы для получения более гибких ответов, формулируя запросы, учитывающие потребность в исключениях. Часть методов требует технических навыков, но концептуальное понимание доступно всем.> ||

<Адаптивность: 75>

Prompt:

Применение исследования об обработке исключений в промптах для GPT
Ключевые выводы для промптинга

Исследование показывает, что языковые модели имеют склонность **слишком строго следовать правилам**, даже когда ситуация требует исключения. Модели, обученные на человеческих объяснениях (а не просто на решениях), демонстрируют более гибкое мышление.

Пример промпта с учетом этих выводов

[======]

Задача для помощника с обработкой исключений Ты — помощник в компании, который должен следовать политике возврата товаров: - Товары принимаются к возврату в течение 14 дней - Товар должен быть в оригинальной упаковке - Чек обязателен

ВАЖНО: Хотя ты должен следовать политике, помни, что в некоторых ситуациях разумные исключения помогают клиентам и бизнесу. Рассмотри все обстоятельства и объясни свой процесс принятия решения.

Когда оцениваешь ситуацию: 1. Сначала определи, соответствует ли запрос стандартной политике 2. Если нет, рассмотри серьезность нарушения (незначительное или существенное) 3. Оцени последствия как строгого соблюдения правил, так и исключения 4. Объясни свое решение с учетом баланса между правилами и здравым смыслом

Ситуация: Клиент купил наушники 15 дней назад. У него есть чек и оригинальная упаковка. Наушники оказались неисправными после первого использования. Как ты поступишь? [======]

Почему этот подход работает

Цепочка рассуждений (chain of thought) — промпт требует пошагового анализа ситуации **Избегание жестких этических фреймворков** — вместо абстрактных принципов используются конкретные шаги **Акцент на объяснении** — модель должна объяснить свое мышление, что имитирует обучение на человеческих объяснениях **Явное разрешение на исключения** — промпт прямо указывает, что разумные исключения допустимы Этот подход помогает преодолеть тенденцию GPT к чрезмерно строгому следованию правилам, что, согласно исследованию, является типичной проблемой языковых моделей при обработке ситуаций, требующих

гибкости в принятии решений.

№ 256. FINEREASON: Оценка и улучшение преднамеренного мышления больших языковых моделей через решение рефлексивных головоломок

Ссылка: https://arxiv.org/pdf/2502.20238

Рейтинг: 65
Адаптивность: 75

Ключевые выводы:

Исследование представляет FINEREASON - новый бенчмарк для оценки способностей больших языковых моделей (LLM) к рассуждениям через решение логических головоломок. В отличие от существующих бенчмарков, которые фокусируются на конечной точности ответа, FINEREASON оценивает промежуточные шаги рассуждений, в частности проверку состояния и переходы между состояниями. Результаты показывают значительные различия между моделями, ориентированными на рассуждения, и моделями общего назначения.

Объяснение метода:

Исследование предлагает ценные концепции для улучшения рассуждений LLM через проверку состояний и планирование шагов. Пользователи могут адаптировать принципы "State Checking" и "State Transition" для получения более надежных ответов. Однако полная реализация методологии требует технических знаний, что ограничивает прямую применимость для обычных пользователей.

Ключевые аспекты исследования: 1. **Введение FINEREASON** - новый бенчмарк для оценки рассуждений LLM на логических головоломках, который фокусируется не только на конечном результате, но и на промежуточных шагах рассуждения.

Две ключевые задачи оценки: "State Checking" (проверка состояния) - способность модели оценить, может ли текущее состояние привести к решаемому результату, и "State Transition" (переход состояния) - способность определить следующий корректный шаг.

Разложение головоломок на атомарные шаги - исследователи предлагают представлять решение головоломок в виде дерева, где узлы - это промежуточные состояния, а ребра - переходы между ними, что позволяет точно оценить способности модели к рассуждению.

Обучающий набор данных - исследователи создали специальный тренировочный набор, который при сочетании с математическими данными значительно улучшает

способности моделей к рассуждению на стандартных математических задачах (до 5.1% на GSM8K).

Выявление значительных различий между моделями, ориентированными на рассуждения (например, o1, Gemini-FT) и общими моделями (GPT-4o, GPT-3.5) в их способности к глубокому рассуждению.

Дополнение:

Применимость методов исследования в стандартном чате

Для работы методов, описанных в исследовании FINEREASON, **не требуется дообучение или API** в контексте использования их концепций обычными пользователями. Хотя исследователи использовали дообучение для улучшения моделей и доступ к API для оценки, основные концепции могут быть адаптированы для стандартных чатов.

Концепции для применения в стандартном чате:

Пошаговое решение с проверкой (State Checking) Пользователь может запрашивать модель не только решить задачу, но и проверить каждый промежуточный шаг Пример: "Решай эту математическую задачу шаг за шагом. После каждого шага проверь, правильно ли он выполнен и может ли он привести к решению"

Планирование следующего шага (State Transition)

Запрос модели определить оптимальный следующий шаг на основе текущего состояния Пример: "Учитывая текущее состояние решения, какой следующий шаг будет оптимальным? Объясни, почему"

Возврат и исследование альтернативных путей

Запрос модели рассмотреть альтернативные подходы, если текущий путь кажется неперспективным Пример: "Этот подход кажется тупиковым. Давай вернемся к предыдущему шагу и рассмотрим альтернативные варианты"

Структурирование решения как дерева

Запрос модели представить различные пути решения в виде дерева с возможностью выбора оптимального пути Пример: "Представь решение этой проблемы как дерево возможных путей. Какие варианты у нас есть на каждом шаге?"

Ожидаемые результаты от применения этих концепций:

• Повышение точности и надежности ответов LLM

• Уменьшение количества логических ошибок в сложных рассуждениях

• Более структурированные и понятные объяснения сложных проблем

• Возможность решения более сложных задач, требующих глубокого рассуждения

• Лучшее понимание пользователем процесса рассуждения LLM

Анализ практической применимости: 1. **Введение FINEREASON** - Прямая применимость: Низкая для обычных пользователей, так как требует технической реализации и доступа к API моделей для проведения оценки. - Концептуальная ценность: Высокая, так как помогает пользователям понять, что важно не только конечное решение, но и качество промежуточных рассуждений. - Потенциал для адаптации: Средний - пользователи могут адаптировать идею проверки промежуточных шагов в своих запросах к LLM.

Две ключевые задачи оценки Прямая применимость: Средняя - пользователи могут адаптировать идею "проверки состояния" и "перехода состояния" в своих запросах, запрашивая модель оценить промежуточные результаты и планировать следующие шаги. Концептуальная ценность: Высокая - концепция "дважды подумать, прежде чем действовать" очень полезна для эффективного использования LLM. Потенциал для адаптации: Высокий - эти принципы могут быть применены к широкому спектру задач, требующих пошагового рассуждения.

Разложение головоломок на атомарные шаги

Прямая применимость: Средняя - пользователи могут запрашивать у LLM пошаговое решение сложных задач с проверкой на каждом этапе. Концептуальная ценность: Высокая - понимание того, что сложные задачи должны быть разбиты на атомарные шаги с проверкой каждого шага. Потенциал для адаптации: Высокий - этот подход применим к широкому спектру задач от математики до планирования.

Обучающий набор данных

Прямая применимость: Низкая - обычные пользователи не могут напрямую использовать обучающие данные. Концептуальная ценность: Средняя - понимание того, что обучение на головоломках может улучшить общие способности рассуждения. Потенциал для адаптации: Низкий - требует специализированных навыков и доступа к обучению моделей.

Выявление различий между моделями

Прямая применимость: Высокая - пользователи могут выбирать модели, которые лучше подходят для задач, требующих глубокого рассуждения. Концептуальная ценность: Высокая - понимание того, что не все модели одинаково хороши в глубоком рассуждении. Потенциал для адаптации: Средний - пользователи могут

адаптировать свои запросы в зависимости от сильных сторон используемой модели. Сводная оценка полезности: На основе проведенного анализа, я оцениваю полезность исследования FINEREASON для широкой аудитории в 65 баллов из 100.

Это исследование имеет высокую полезность для пользователей LLM по нескольким причинам:

Оно предлагает концептуальный фреймворк для улучшения взаимодействия с LLM через пошаговое рассуждение, проверку промежуточных результатов и планирование следующих шагов.

Исследование выявляет сильные и слабые стороны различных моделей в задачах рассуждения, что помогает пользователям выбрать подходящую модель.

Концепции "State Checking" и "State Transition" могут быть применены пользователями при формулировке запросов для получения более надежных и обоснованных ответов.

Возможные контраргументы к этой оценке:

Почему оценка могла бы быть выше: - Исследование предлагает конкретные методы для улучшения процесса рассуждения, которые могут быть адаптированы для различных задач. - Результаты показывают значительное улучшение математического рассуждения при применении предложенных подходов.

Почему оценка могла бы быть ниже: - Большинство результатов требуют технической реализации и не могут быть напрямую использованы обычными пользователями. - Исследование больше ориентировано на разработчиков и исследователей LLM, чем на конечных пользователей.

После рассмотрения этих аргументов, я сохраняю оценку 65, так как, несмотря на техническую сложность, концептуальные идеи исследования имеют высокую ценность и могут быть адаптированы пользователями для улучшения их взаимодействия с LLM.

Уверенность в оценке: Моя уверенность в оценке: очень сильная.

Я уверен в своей оценке, поскольку тщательно проанализировал все ключевые аспекты исследования и их применимость для широкой аудитории. Исследование предлагает как концептуальные идеи, которые могут быть адаптированы пользователями, так и технические методы, которые требуют специализированных знаний. Баланс между этими аспектами хорошо отражен в оценке 65.

Оценка адаптивности: Оценка адаптивности: 75 из 100.

1) Принципы "State Checking" и "State Transition" могут быть легко адаптированы для использования в обычном чате. Пользователи могут запрашивать модель оценивать текущее состояние решения и предлагать следующие шаги.

2) Подход разбиения сложных задач на атомарные шаги с проверкой каждого шага может быть применен к широкому спектру задач от математических проблем до планирования проектов.

3) Концепция "думай дважды перед действием" может быть интегрирована в стандартные промпты для улучшения качества ответов LLM.

4) Метод представления решения в виде дерева с возможностью возврата к предыдущим состояниям может быть адаптирован для решения сложных проблем, где важно исследовать различные пути.

Исследование предлагает принципы и концепции, которые могут быть использованы в обычном чате без необходимости доступа к API или дообучения моделей, что делает его высоко адаптивным для широкого круга пользователей.

|| <Оценка: 65> || <Объяснение: Исследование предлагает ценные концепции для улучшения рассуждений LLM через проверку состояний и планирование шагов. Пользователи могут адаптировать принципы "State Checking" и "State Transition" для получения более надежных ответов. Однако полная реализация методологии требует технических знаний, что ограничивает прямую применимость для обычных пользователей.> || <Адаптивность: 75>

Prompt:

Применение исследования FINEREASON в промптах для GPT Исследование FINEREASON предоставляет ценные инсайты о том, как улучшить рассуждения языковых моделей при решении сложных задач. Вот как можно применить эти знания в промптах.

Ключевые принципы для улучшения промптов:

Разбивать рассуждения на атомарные шаги (проверка состояния + переход состояния) **Явно запрашивать проверку промежуточных результатов Поощрять бэктрекинг** при обнаружении тупиковых путей **Требовать проверку противоречий** в рассуждениях **Предоставлять больше контекста** для сложных задач
Пример промпта с применением методологии FINEREASON:

[======]

Задача решения логической головоломки Я предоставлю тебе логическую головоломку. Решая ее, следуй этому структурированному подходу:

Анализ начального состояния: Опиши исходные данные и ограничения Проверь, является ли начальное состояние валидным

На каждом шаге рассуждения:

Проверка текущего состояния: Оцени, может ли текущее состояние привести к решению **Выбор следующего перехода**: Определи возможные следующие шаги и выбери оптимальный **Проверка противоречий**: Убедись, что новое состояние не противоречит ранее установленным фактам

При обнаружении тупика:

Явно отметь это Вернись к предыдущему состоянию (бэктрекинг) Выбери альтернативный путь

Для финального решения:

Проверь, что все условия головоломки выполнены Проверь полноту решения Вот головоломка: [описание головоломки] [======]

Почему это работает:

Данный промпт применяет ключевые открытия исследования FINEREASON:

• Разделяет процесс рассуждения на проверку состояния и переход состояния, что помогает преодолеть "разрыв в исполнении"

• Стимулирует рефлексию через явные проверки противоречий

• Внедряет механизм бэктрекинга, что помогает избежать застревания в тупиковых путях

• Структурирует мыслительный процесс в виде дерева решений, как предлагается в методологии исследования

Такой подход особенно эффективен для моделей общего назначения, которые, согласно исследованию, часто пропускают промежуточные шаги рассуждения, стремясь сразу получить конечный ответ.

№ 260. Вознаграждение процесса графового рассуждения делает LLM более обобщенными рассуждателями

Ссылка: https://arxiv.org/pdf/2503.00845

Рейтинг: 62
Адаптивность: 75

Ключевые выводы:

Исследование направлено на улучшение способностей больших языковых моделей (LLM) решать задачи графового рассуждения с помощью модели вознаграждения процесса (Process Reward Model, PRM). Основной результат: разработанная модель GraphPRM значительно улучшает производительность LLM на 13 задачах графовых вычислений, обеспечивая прирост на 9% для Qwen-2.5-7B и демонстрируя способность к переносу на новые наборы данных графового рассуждения и другие области рассуждения, такие как математические задачи.

Объяснение метода:

Исследование предлагает ценные концепции пошагового рассуждения и проверки для улучшения взаимодействия с LLM. Хотя технические аспекты требуют значительной адаптации, пользователи могут применять принципы генерации нескольких решений, структурированного рассуждения и перекрестного использования навыков между доменами задач в повседневной работе с LLM.

Ключевые аспекты исследования: 1. **Разработка Process Reward Model для графовых задач**: Исследование представляет GraphPRM - первую модель вознаграждения процесса (Process Reward Model) для задач графовых вычислений, которая оценивает каждый шаг рассуждения LLM и присваивает ему оценку корректности.

Создание датасета GraphSilo: Авторы создали крупнейший датасет для графовых вычислительных задач с детальной пошаговой разметкой. Для автоматической генерации правильных и неправильных шагов рассуждения использованы алгоритмы поиска по дереву Монте-Карло и ориентированные на задачи траектории.

Повышение эффективности вывода LLM: GraphPRM применяется для улучшения производительности LLM во время вывода, оценивая и выбирая лучшие рассуждения из нескольких кандидатов, а также для обучения с подкреплением через Direct Preference Optimization (DPO).

Кросс-доменная применимость: Исследование демонстрирует, что GraphPRM, обученная на графовых задачах, эффективно переносится на другие домены

рассуждений, включая математические задачи, что указывает на универсальность подхода.

Улучшение производительности различных LLM: Метод показывает значительное улучшение производительности для разных моделей (Qwen, Llama, Gemma) на 13 графовых задачах, с прибавкой до 9% для Qwen-2.5-7B.

Дополнение:

Применимость методов исследования в стандартном чате

Исследование действительно требует дообучения моделей и использования специальных API для полной реализации описанных методов. Однако несколько ключевых концепций и подходов могут быть адаптированы для применения в стандартном чате без дополнительного обучения:

Пошаговое рассуждение с самопроверкой Пользователи могут запрашивать у LLM выполнение задачи с явным разбиением на шаги На каждом шаге можно просить модель оценивать корректность своих рассуждений Пример: "Реши эту задачу, разбив решение на пронумерованные шаги. После каждого шага проверяй его корректность и исправляй при необходимости."

Генерация множественных решений

Пользователи могут запрашивать у LLM несколько различных подходов к решению задачи Затем просить модель сравнить эти подходы и выбрать лучший Пример: "Предложи три разных способа решения этой задачи. Затем сравни их и выбери наиболее надежный."

Структурирование задач по аналогии с графовыми алгоритмами

Применение принципов декомпозиции задачи на взаимосвязанные компоненты Использование итеративных подходов для сложных задач Пример: "Давай решим эту проблему, представив её как граф, где ключевые элементы - это узлы, а их взаимосвязи - рёбра."

Применение межпредметного переноса рассуждений

Использование структурированных подходов из одной области для решения задач в другой Пример: "Давай применим подход, аналогичный поиску кратчайшего пути в графе, для оптимизации этого бизнес-процесса." Ожидаемые результаты от применения этих концепций: - Повышение точности решений за счет более структурированного рассуждения - Снижение количества ошибок благодаря проверке промежуточных шагов - Улучшение способности решать сложные многошаговые задачи - Более глубокое понимание проблемы через рассмотрение нескольких подходов

Prompt:

Использование GraphPRM в промптах для GPT Исследование о графовом рассуждении и модели вознаграждения процесса (GraphPRM) предоставляет ценные знания для улучшения промптов при работе с GPT. Вот как можно применить эти знания на практике:

Ключевые принципы для использования в промптах

Пошаговое рассуждение — разбивайте сложные задачи на четкие этапы **Явное графовое представление** — визуализируйте связи между элементами **Самооценка решений** — просите модель оценивать качество своих промежуточных шагов **Множественные решения** — генерируйте несколько подходов к решению ## Пример промпта для решения сложной задачи

[======] Помоги мне решить следующую задачу о графе социальных связей:

[ОПИСАНИЕ ЗАДАЧИ]

Используй следующий подход:

Сначала представь граф явно, обозначив узлы и связи между ними Разбей решение на четкие последовательные шаги На каждом шаге: Объясни, что ты делаешь и почему Отслеживай уже посещенные узлы Оцени правильность промежуточного результата Предложи два различных способа решения Сравни полученные результаты и выбери наиболее достоверный После завершения решения проанализируй возможные ошибки в процессе рассуждения и оцени надежность своего ответа. [======]

Почему это работает

Данный подход основан на методологии GraphPRM, которая показала улучшение производительности на 9% для сложных задач рассуждения. Промпт включает:

• Структурированное рассуждение — аналог задачно-ориентированных траекторий из исследования

• Самооценку процесса — элемент модели вознаграждения процесса

• Генерацию нескольких решений — похоже на масштабирование во время вывода

• Сравнение результатов — имитация процесса выбора лучшего решения

Такой подход особенно эффективен для задач, требующих сложного многошагового рассуждения, и может быть адаптирован не только для графовых, но и для математических и логических задач.

№ 264. Шахерезада: Оценка математического рассуждения с помощью цепочки цепочек проблем в языковых моделях

Ссылка: https://arxiv.org/pdf/2410.00151

Рейтинг: 62
Адаптивность: 75

Ключевые выводы:

Исследование представляет Scheherazade - автоматизированный подход для создания сложных математических тестовых задач путем логического объединения существующих задач в цепочки. Основной результат: в то время как производительность большинства современных LLM резко падает при увеличении длины цепочки задач, модель O1-preview от OpenAI демонстрирует устойчивую производительность, особенно при обратном связывании задач.

Объяснение метода:

Исследование предлагает ценные концепции для понимания возможностей LLM в логических рассуждениях. Методы forward и backward chaining могут быть адаптированы для проверки последовательности рассуждений моделей. Знание типичных ошибок помогает формулировать эффективные запросы. Однако практическая реализация требует технических знаний, что ограничивает доступность для широкой аудитории.

Ключевые аспекты исследования: 1. **Scheherazade** - инструмент для создания сложных математических задач путем логического соединения (chaining) существующих задач, что позволяет оценивать способности LLM к рассуждению.

Forward chaining и Backward chaining - две техники связывания задач: прямое соединение (решение последовательно) и обратное соединение (решение требует информации из последующих задач), что создает более сложные проблемы для тестирования LLM.

Оценка моделей через цепочки разной длины - исследование показывает, что точность всех моделей снижается при увеличении длины цепочки, особенно при backward chaining, что позволяет лучше дифференцировать их возможности рассуждения.

Анализ ошибок - выявлены основные типы ошибок моделей: семантическое непонимание, ошибки выбора пути решения, ложноотрицательные результаты и другие, что помогает понять слабые места в рассуждениях LLM.

Масштабируемость генерации бенчмарков - из небольшого набора исходных задач можно создать огромное количество новых сложных бенчмарков, что решает проблему быстрого устаревания существующих тестов.

Дополнение:

Применимость методов в стандартном чате

Методы исследования **не требуют** дообучения или специального API для их применения пользователями. Хотя исследователи использовали API для систематической оценки и создания бенчмарков, основные концепции можно применить в стандартном чате.

Применимые концепции и подходы

Структурирование сложных запросов Пользователи могут создавать запросы с условными ветвлениями ("если X верно, то Y, иначе Z") Такая структура позволяет проверить способность модели следовать логической цепочке

Оценка прямого и обратного рассуждения

Forward chaining: задачи, решаемые последовательно ("Реши A, затем используй результат для B") Backward chaining: задачи, требующие предвидения ("Чтобы решить A, сначала определи, что нужно знать из B")

Проверка устойчивости рассуждений

Постепенное увеличение длины цепочки рассуждений для оценки надежности модели Выявление порога сложности, при котором модель начинает делать ошибки ### Ожидаемые результаты

• Более структурированные и последовательные ответы от LLM

• Выявление ситуаций, когда модель теряет логическую нить рассуждений

• Возможность проверить надежность решения сложных задач

• Лучшее понимание того, как формулировать запросы для получения качественных рассуждений

Prompt:

Использование знаний из исследования Scheherazade в промптах для GPT ## Ключевые инсайты из исследования

Исследование Scheherazade выявило важные различия в способности языковых

моделей обрабатывать цепочки задач с разными типами связывания:

• Прямое связывание (forward chaining) - последовательное решение задач

• Обратное связывание (backward chaining) - требует информации из последующих задач

Большинство моделей (кроме O1-preview) демонстрируют резкое падение точности при увеличении длины цепочки, особенно при обратном связывании.

Пример промпта с использованием знаний из исследования

[======] # Задание: Помогите решить комплексную бизнес-задачу с многоэтапным анализом

Структура промпта (использую прямое связывание для повышения точности):

Сначала проанализируйте базовые финансовые показатели компании за последний квартал: Выручка: $2.3 млн Операционные расходы: $1.7 млн Маржинальность: ?

На основе полученной маржинальности, определите:

Является ли бизнес финансово устойчивым? Какие показатели требуют улучшения?

Используя результаты предыдущего анализа, предложите:

3 краткосрочные стратегии оптимизации расходов 2 долгосрочные стратегии увеличения выручки ## Важно: - Решайте задачу последовательно, шаг за шагом - Для каждого шага четко обозначайте промежуточные выводы - Используйте числовые данные для подтверждения рассуждений [======]

Объяснение эффективности промпта

Этот промпт использует **принцип прямого связывания** из исследования Scheherazade, что повышает вероятность получения точного ответа от большинства языковых моделей:

Последовательная структура: Задачи выстроены так, что каждая следующая опирается на результаты предыдущей, что соответствует прямому связыванию

Явное разделение на этапы: Четкая нумерация и структурирование помогают модели организовать процесс рассуждения

Избегание обратного связывания: Промпт не требует от модели использовать информацию "из будущего", что, согласно исследованию, значительно снижает точность большинства LLM

Инструкции по процессу решения: Указание решать последовательно и

фиксировать промежуточные результаты помогает модели избежать "потери контекста" при длинных цепочках рассуждений

Для O1-preview можно создавать более сложные промпты с обратным связыванием, так как эта модель показывает исключительную устойчивость к таким задачам.

№ 268. Оценка предпочтений языковой модели с помощью нескольких слабых оценщиков

Ссылка: https://arxiv.org/pdf/2410.12869

Рейтинг: 62
Адаптивность: 75

Ключевые выводы:

Исследование направлено на решение проблемы противоречивых оценок в системах оценки предпочтений языковых моделей. Авторы представили новый метод GED (Preference Graph Ensemble and Denoise), который объединяет оценки от нескольких слабых оценщиков-LLM и устраняет противоречия в графах предпочтений, что позволяет получить более надежные и непротиворечивые результаты оценки.

Объяснение метода:

Исследование демонстрирует, как комбинирование оценок нескольких "слабых" моделей может превзойти одну "сильную" модель. Эта концепция адаптируема для обычных пользователей через запросы к разным моделям или использование разных формулировок. Метод устранения противоречий в оценках имеет высокую концептуальную ценность, помогая понять ограничения LLM и улучшить критическую оценку полученных ответов.

Ключевые аспекты исследования: 1. **Метод GED (Graph Ensemble and Denoise)** - новый подход к оценке предпочтений между ответами языковых моделей, который объединяет оценки нескольких "слабых оценщиков" (языковых моделей) и устраняет противоречия в них.

Двухэтапный процесс обработки предпочтений - агрегирование оценок в единый граф предпочтений и применение алгоритма очистки для устранения циклических несоответствий (когда A лучше B, B лучше C, но C лучше A).

Теоретические гарантии - авторы доказывают, что их метод может восстанавливать истинную структуру предпочтений с высокой вероятностью при определенных условиях.

Превосходство комбинации "слабых оценщиков" - исследование показывает, что объединение нескольких небольших моделей (например, Llama3-8B, Mistral-7B, Qwen2-7B) может превзойти по качеству оценки более крупные модели (например, Qwen2-72B).

Три практических применения метода - ранжирование моделей, выбор лучших ответов и настройка моделей на основе отобранных инструкций.

Дополнение:

Исследование не требует дообучения или специального API для применения его основных концепций в стандартном чате. Хотя авторы использовали продвинутые технические подходы для экспериментов, ключевые идеи работают и в обычном взаимодействии с LLM.

Концепции, которые можно применить в стандартном чате:

Агрегирование мнений нескольких "оценщиков" - пользователь может задавать один вопрос разным моделям или одной модели несколькими способами, затем объединять полученные ответы. Это снижает влияние случайных ошибок отдельных моделей.

Выявление и устранение противоречий - пользователь может попросить модель проверить свои выводы на непротиворечивость или сравнить ответы на близкие вопросы, чтобы выявить несоответствия.

Попарное сравнение вместо абсолютных оценок - вместо оценки каждого ответа по отдельности, пользователь может запрашивать модель сравнить варианты между собой, что часто дает более надежные результаты.

Структурированный процесс оценки - пользователь может задать модели четкие критерии для сравнения ответов (корректность, полнота, ясность), что повышает качество оценки.

Ожидаемые результаты: - Повышение надежности и последовательности ответов LLM - Снижение влияния случайных ошибок и предвзятостей отдельных моделей - Улучшение критического мышления при оценке информации от LLM - Более эффективное выявление противоречивых или некорректных ответов

Методы исследования могут быть особенно полезны при работе со сложными или неоднозначными запросами, где стандартные ответы модели могут содержать противоречия или неточности.

Prompt:

Применение исследования GED в промптах для GPT ## Основные принципы из исследования

Исследование GED (Preference Graph Ensemble and Denoise) показывает, что: - Объединение мнений нескольких "слабых" оценщиков часто лучше, чем мнение одного "сильного" - Устранение противоречий в оценках критически важно для получения качественных результатов - Представление предпочтений в виде графов помогает структурировать процесс оценки

Пример промпта, использующего принципы GED

[======] # Задание: Оценка нескольких вариантов ответа

Контекст Я собрал несколько вариантов ответа на вопрос "[вставить вопрос]". Мне нужна твоя помощь в их оценке, используя подход, вдохновленный методом GED.

Инструкция 1. Сначала оцени каждый вариант ответа с трех разных перспектив: - Как эксперт в предметной области (фокус на фактической точности) - Как редактор (фокус на ясности и структуре) - Как обычный пользователь (фокус на полезности и доступности)

Для каждой перспективы: Ранжируй ответы от лучшего к худшему Укажи причины твоего ранжирования

Затем объедини эти три ранжирования в финальное, устраняя противоречия:

Если есть конфликты в ранжировании, объясни, как ты их разрешаешь Построй финальный "граф предпочтений" без циклов и противоречий

Представь итоговое ранжирование с кратким обоснованием для каждой позиции

Варианты ответов для оценки: [Вариант A]: [текст ответа] [Вариант B]: [текст ответа] [Вариант C]: [текст ответа] [======]

Как это работает

Множественные оценщики: Промпт заставляет GPT принять на себя роли трех разных "оценщиков" (эксперт, редактор, пользователь), что имитирует ансамбль слабых оценщиков из исследования GED.

Представление в виде графа: Хотя явно не используется математический граф, промпт требует ранжирования, которое по сути создает направленный граф предпочтений.

Устранение противоречий: Финальный этап требует объединения разных оценок и устранения противоречий, что соответствует этапу "denoise" в методе GED.

Обоснование решений: Требование объяснять причины ранжирования и разрешения противоречий помогает получить более надежную и обоснованную оценку.

Этот подход позволяет получить более сбалансированную и надежную оценку вариантов, чем при использовании одной перспективы, даже если все оценки выполняются одной моделью GPT.

№ 272. SAGE: Framework точного извлечения для RAG

Ссылка: https://arxiv.org/pdf/2503.01713

Рейтинг: 62
Адаптивность: 70

Ключевые выводы:

Исследование представляет SAGE - новую фреймворк для повышения точности извлечения информации в системах RAG (Retrieval Augmented Generation). Основная цель - преодолеть ограничения существующих RAG-систем, связанные с неэффективной сегментацией корпуса и проблемами извлечения релевантной информации. Результаты показывают, что SAGE превосходит базовые методы на 61.25% по качеству ответов на вопросы и на 49.41% по эффективности затрат.

Объяснение метода:

SAGE предлагает ценные концепции для работы с LLM: семантическая целостность контекста, динамический отбор информации и самооценка качества ответов. Хотя техническая реализация недоступна обычным пользователям, принципы можно адаптировать для улучшения запросов к LLM и структурирования информации.

Ключевые аспекты исследования: 1. **Фреймворк SAGE для точного поиска в RAG-системах** - исследование представляет комплексный подход к улучшению точности поиска в системах Retrieval Augmented Generation (RAG), решая проблемы семантической сегментации текста, динамического отбора информации и самооценки релевантности контекста.

Семантическая сегментация корпуса - разработана легковесная модель для разделения текста на семантически целостные фрагменты, что решает проблему неэффективного разделения текста традиционными методами.

Градиентный отбор фрагментов - предложен алгоритм динамического отбора фрагментов на основе градиента релевантности, позволяющий избежать как недостатка важной информации, так и зашумления контекста.

Самообратная связь LLM - внедрен механизм, позволяющий языковой модели оценивать качество ответа и корректировать количество извлекаемых фрагментов для улучшения точности.

Экспериментальное подтверждение - проведены обширные эксперименты, демонстрирующие превосходство SAGE над базовыми методами как по качеству ответов, так и по эффективности использования токенов.

Дополнение:

Требуется ли API или дообучение?

Для полной реализации методов SAGE требуется API и дообучение специализированных моделей. Однако многие концептуальные подходы можно адаптировать для использования в стандартном чате:

Семантическая сегментация: Вместо автоматической сегментации пользователи могут: Разделять длинные тексты на смысловые блоки вручную Использовать естественные границы смысловых блоков (абзацы, разделы) Просить LLM "разделить текст на логические фрагменты" перед выполнением основной задачи

Градиентный отбор фрагментов:

Пользователи могут сначала попросить LLM оценить релевантность каждого фрагмента текста к вопросу Использовать только фрагменты с высокой релевантностью Постепенно добавлять контекст, начиная с наиболее релевантного

Механизм самообратной связи:

После получения ответа спрашивать у LLM: "Оцени качество своего ответа. Достаточно ли контекста?" При недостатке контекста добавлять информацию При избытке контекста сокращать его Ожидаемые результаты от применения этих подходов: - Повышение точности ответов благодаря более релевантному контексту - Снижение проблем с "зашумлением" контекста избыточной информацией - Более эффективное использование контекстного окна LLM

Анализ практической применимости: 1. **Фреймворк SAGE для точного поиска** - **Прямая применимость**: Средняя. Требует технических знаний для внедрения фреймворка в существующие системы, недоступно для обычного пользователя. - **Концептуальная ценность**: Высокая. Пользователи могут понять, что эффективность RAG-систем зависит от качества извлекаемых фрагментов и что избыточная или недостаточная информация снижает точность ответов. - **Потенциал для адаптации**: Высокий. Принципы динамического отбора информации применимы для формулирования более точных запросов к LLM.

Семантическая сегментация корпуса Прямая применимость: Низкая. Требует создания и обучения специализированной модели. **Концептуальная ценность**: Высокая. Понимание важности семантической целостности контекста может помочь пользователям лучше структурировать свои запросы. **Потенциал для адаптации**: Средний. Принцип семантической целостности может быть использован при ручном разделении текста для загрузки в LLM.

Градиентный отбор фрагментов

Прямая применимость: Низкая. Алгоритм требует технической реализации. **Концептуальная ценность**: Высокая. Понимание, что не всегда "больше контекста = лучше" может помочь пользователям отбирать релевантную информацию для запросов. **Потенциал для адаптации**: Средний. Пользователи могут применять принцип "отсечения" по снижению релевантности при ручном отборе информации.

Самообратная связь LLM

Прямая применимость: Средняя. Пользователи могут адаптировать идею запроса к LLM для оценки качества ответа. **Концептуальная ценность**: Высокая. Демонстрирует способность LLM к самооценке и итеративному улучшению ответов. **Потенциал для адаптации**: Высокий. Пользователи могут внедрить практику запроса обратной связи от LLM для оценки качества ответа и корректировки контекста.

Экспериментальное подтверждение

Прямая применимость: Низкая. Результаты экспериментов сами по себе не применимы напрямую. **Концептуальная ценность**: Средняя. Понимание соотношения различных факторов, влияющих на качество RAG. **Потенциал для адаптации**: Низкий. Экспериментальные данные имеют в основном академическую ценность. Сводная оценка полезности: Предварительная оценка: 55

SAGE представляет собой технически сложный фреймворк, требующий серьезных знаний для прямой реализации. Однако, концептуальные идеи, лежащие в основе исследования, имеют значительную ценность для широкой аудитории, использующей LLM.

Контраргументы к поднятию оценки: 1. Исследование технически сложно и требует специальных знаний для реализации. 2. Полная реализация SAGE недоступна для обычных пользователей без навыков программирования.

Контраргументы к снижению оценки: 1. Концептуальные принципы (важность семантически целостных фрагментов, баланс между недостаточной и избыточной информацией) могут быть применены даже без технической реализации. 2. Механизм самообратной связи может быть адаптирован пользователями в виде простых промптов для улучшения ответов LLM. 3. Понимание проблем RAG поможет пользователям формулировать более эффективные запросы.

Скорректированная оценка: 62

Исследование имеет высокую полезность благодаря концептуальным идеям, которые могут быть адаптированы для улучшения взаимодействия с LLM, несмотря на техническую сложность прямой реализации.

Уверенность в оценке: Очень сильная. Я тщательно проанализировал исследование и оценил как его технические аспекты, так и концептуальную ценность для

различных категорий пользователей. Учтены контраргументы, и оценка была скорректирована соответствующим образом.

Оценка адаптивности: Оценка адаптивности: 70

1) Принципы исследования хорошо адаптируемы: концепции семантической целостности контекста, динамического отбора информации и самообратной связи могут быть применены пользователями при взаимодействии с LLM даже без технической реализации.

2) Пользователи могут извлечь полезные идеи, например: разделять информацию на семантически связанные блоки, исключать малорелевантные данные, использовать LLM для оценки качества ответов и корректировки запросов.

3) Высокий потенциал для внедрения: механизмы самооценки и итеративного улучшения ответов особенно перспективны для будущих взаимодействий с LLM.

4) Хотя технические методы требуют специальных знаний, концептуальные принципы могут быть абстрагированы до простых рекомендаций по взаимодействию с LLM.

|| <Оценка: 62> || <Объяснение: SAGE предлагает ценные концепции для работы с LLM: семантическая целостность контекста, динамический отбор информации и самооценка качества ответов. Хотя техническая реализация недоступна обычным пользователям, принципы можно адаптировать для улучшения запросов к LLM и структурирования информации.> || <Адаптивность: 70>

Prompt:

Использование исследования SAGE в промптах для GPT
Ключевые применимые знания из исследования

• Семантическая сегментация вместо разбиения на фрагменты фиксированной длины

• Градиентный выбор фрагментов для динамического определения оптимального количества информации

• Механизм самооценки для проверки достаточности и избыточности контекста

• Оптимизация затрат за счет уменьшения количества нерелевантных токенов

Пример промпта, использующего принципы SAGE

[======] Ты - эксперт по анализу финансовых данных, использующий методологию SAGE для точного извлечения информации. Я предоставлю тебе финансовый отчет

компании, и мне нужен анализ перспектив её роста.

Используй следующий подход:

СЕМАНТИЧЕСКАЯ СЕГМЕНТАЦИЯ: Раздели информацию на смысловые блоки (доходы, расходы, инвестиции, риски) Фокусируйся на смысловой целостности каждого блока, а не на их размере

ГРАДИЕНТНЫЙ ВЫБОР ИНФОРМАЦИИ:

Начни с наиболее релевантных для роста показателей Добавляй информацию, пока её ценность для анализа роста значительна Прекрати добавление, когда новые данные перестают существенно влиять на выводы

САМООЦЕНКА ДОСТАТОЧНОСТИ:

В конце проверь, достаточно ли собранной информации для обоснованного вывода Отметь области, где информации недостаточно

СТРУКТУРА ОТВЕТА:

Сначала представь краткое резюме о перспективах роста (3-4 предложения) Затем приведи основные факторы роста с соответствующими данными Укажи потенциальные риски и ограничения Завершение: общая оценка перспектив роста по 10-балльной шкале Вот финансовый отчет: [ТЕКСТ ОТЧЕТА] [======]

Объяснение эффективности промпта

Данный промпт применяет ключевые принципы SAGE для повышения качества анализа:

Семантическая сегментация позволяет GPT структурировать информацию по смыслу, а не механически, что повышает релевантность извлекаемых данных.

Градиентный подход направляет модель на выбор только значимой информации, предотвращая перегрузку контекста нерелевантными деталями.

Механизм самооценки заставляет модель критически оценить достаточность собранной информации, что повышает надежность выводов.

Структурированный вывод оптимизирует использование токенов, фокусируясь на наиболее ценной информации.

Такой подход, согласно исследованию SAGE, может повысить точность ответов на 61% и снизить затраты на токены почти на 50% по сравнению со стандартными методами.

№ 276. Ненастоящие языки - это не ошибки, а особенности для больших языковых моделей

Ссылка: https://arxiv.org/pdf/2503.01926

Рейтинг: 62
Адаптивность: 75

Ключевые выводы:

Исследование направлено на систематическое изучение «неестественных языков» - строк текста, которые кажутся непонятными для людей, но сохраняют семантический смысл для LLM. Основной вывод: неестественные языки не являются ошибками, а представляют собой особенности LLM, содержащие скрытые паттерны, которые могут быть использованы моделями для понимания и выполнения задач.

Объяснение метода:

Исследование демонстрирует, что LLM могут понимать даже сильно искаженный текст, что имеет высокую концептуальную ценность для понимания работы моделей. Однако методы требуют специальных алгоритмов, недоступных обычным пользователям. Ценность в основном в понимании устойчивости LLM к шуму и способов эффективной формулировки запросов.

Ключевые аспекты исследования: 1. **Концепция неестественных языков (unnatural languages)** - исследование показывает, что строки текста, кажущиеся бессмысленными для человека, но сохраняющие семантическое значение для LLM, не являются "багами", а представляют собой полезные функции, которые модели могут эффективно обрабатывать.

Метод поиска неестественных версий текста - авторы разработали алгоритм для преобразования естественных текстов в их семантически эквивалентные, но синтаксически "неестественные" версии, которые сохраняют смысл, но выглядят как зашумленный текст.

Перенос знаний между моделями и задачами - исследование демонстрирует, что неестественные языки содержат латентные признаки, которые можно обобщать на разные модели и задачи во время вывода.

Обучение на неестественных инструкциях - модели, обученные на неестественных версиях наборов инструкций, показывают сопоставимую производительность с моделями, обученными на естественном языке.

Механизмы обработки неестественных языков - авторы показывают, что LLM обрабатывают неестественные языки путем фильтрации шума и извлечения

контекстуального значения из отфильтрованных слов.

Дополнение:

Исследование действительно использует API и дообучение для своих экспериментов, однако основные концепции и выводы можно адаптировать для применения в стандартном чате без этих расширенных техник.

Основные концепции, которые можно применить в стандартном чате:

Устойчивость к шуму и искажениям. Исследование показывает, что LLM способны извлекать смысл даже из сильно искаженного текста. Это означает, что пользователи могут не беспокоиться о совершенстве формулировок - модели все равно могут понять суть запроса, даже если он содержит опечатки, грамматические ошибки или нестандартный синтаксис.

Фокус на ключевых словах. Модели уделяют особое внимание ключевым словам, даже если они расположены не в том порядке. Пользователи могут использовать это, подчеркивая важные термины в своих запросах, даже если общая структура запроса не идеальна.

Контекстное понимание. LLM способны восстанавливать правильный порядок и связи между словами, основываясь на контексте. Это можно использовать при формулировании сложных запросов, где важно передать общий контекст, а не идеальную структуру предложений.

Адаптация к нестандартным формулировкам. Исследование демонстрирует, что модели могут адаптироваться к нестандартным формулировкам запросов. Пользователи могут экспериментировать с различными стилями запросов, не опасаясь, что модель их не поймет.

Практические результаты применения этих концепций: - Более устойчивые к ошибкам и опечаткам запросы - Возможность использовать сокращенные или нестандартные формулировки при ограниченном контексте - Уверенность в том, что модель поймет суть запроса, даже если он сформулирован не идеально - Возможность эффективно формулировать запросы на неродном языке, даже при наличии грамматических ошибок

Анализ практической применимости: 1. **Концепция неестественных языков** - Прямая применимость: Низкая для обычных пользователей, так как требуются специальные алгоритмы для создания эффективных неестественных запросов. - Концептуальная ценность: Высокая, поскольку помогает пользователям понять, что LLM способны извлекать смысл даже из зашумленного и неструктурированного текста, что может быть полезно при работе с нечеткими или неточными запросами. - Потенциал для адаптации: Средний, концепция может быть упрощена для использования в виде рекомендаций по формулированию устойчивых к шуму запросов.

Метод поиска неестественных версий текста Прямая применимость: Низкая для обычных пользователей из-за сложности и вычислительных требований. Концептуальная ценность: Средняя, демонстрирует устойчивость LLM к синтаксическим искажениям. Потенциал для адаптации: Средний, метод может быть упрощен для создания более устойчивых промптов.

Перенос знаний между моделями и задачами

Прямая применимость: Низкая для конечных пользователей, больше для разработчиков. Концептуальная ценность: Высокая, показывает, что LLM обладают общими механизмами понимания, которые работают даже с неоптимальными входными данными. Потенциал для адаптации: Средний, знание о переносе может помочь пользователям формулировать запросы, которые будут работать для разных моделей.

Обучение на неестественных инструкциях

Прямая применимость: Низкая для обычных пользователей, полезно для разработчиков моделей. Концептуальная ценность: Высокая, показывает, что модели могут обучаться даже на искаженных данных, сохраняя эффективность. Потенциал для адаптации: Средний, может привести к разработке более устойчивых моделей.

Механизмы обработки неестественных языков

Прямая применимость: Средняя, понимание этих механизмов может помочь пользователям оптимизировать запросы. Концептуальная ценность: Очень высокая, даёт глубокое понимание того, как LLM извлекают смысл из текста. Потенциал для адаптации: Высокий, знание о том, как LLM фильтруют шум и извлекают ключевые слова, может быть применено для создания более эффективных запросов. Сводная оценка полезности: Предварительная оценка: 65/100

Исследование демонстрирует высокую концептуальную ценность, показывая, что LLM способны понимать смысл даже в сильно искаженных текстах. Это имеет важные практические следствия для понимания устойчивости моделей к шуму и формулирования запросов.

Контраргументы к высокой оценке: 1. Большинство методов требуют специальных алгоритмов и технических знаний, недоступных обычным пользователям. 2. Прямое применение неестественных языков в повседневных запросах ограничено и может даже снизить эффективность взаимодействия с LLM.

Контраргументы к низкой оценке: 1. Понимание того, как LLM извлекают смысл из текста, даже неестественного, может помочь пользователям формулировать более эффективные запросы. 2. Концептуальные знания о механизмах работы LLM с неоптимальными входными данными повышают общее понимание возможностей и ограничений этих систем.

Скорректированная оценка: 62/100

Исследование имеет высокую концептуальную ценность, но ограниченную прямую применимость для широкой аудитории. Основная ценность заключается в углублении понимания того, как LLM обрабатывают информацию, что может косвенно улучшить взаимодействие пользователей с этими системами.

Уверенность в оценке: Очень сильная. Исследование предоставляет достаточно данных для понимания как технических аспектов, так и их потенциального влияния на взаимодействие с LLM. Оценка учитывает баланс между концептуальной ценностью и практической применимостью для широкой аудитории.

Оценка адаптивности: Оценка адаптивности: 75/100

Исследование демонстрирует высокую адаптивность по следующим причинам:

Концептуальное понимание устойчивости LLM к шуму и искажениям может быть преобразовано в практические рекомендации по формулированию запросов.

Знание о том, как LLM извлекают ключевые слова и фильтруют шум, может помочь пользователям создавать более эффективные запросы даже в стандартных чатах.

Понимание механизмов переупорядочивания и интерпретации слов моделями может быть использовано для создания более устойчивых к ошибкам и опечаткам промптов.

Выводы о способности моделей понимать контекст и восстанавливать значение даже из несовершенных запросов могут быть применены пользователями при формулировании сложных задач.

Несмотря на техническую сложность самого исследования, его концептуальные выводы могут быть адаптированы для практического использования широкой аудиторией.

|| <Оценка: 62> || <Объяснение: Исследование демонстрирует, что LLM могут понимать даже сильно искаженный текст, что имеет высокую концептуальную ценность для понимания работы моделей. Однако методы требуют специальных алгоритмов, недоступных обычным пользователям. Ценность в основном в понимании устойчивости LLM к шуму и способов эффективной формулировки запросов.> || <Адаптивность: 75>

Prompt:

Использование неестественных языков в промптах для GPT
Суть исследования

Исследование показывает, что большие языковые модели (LLM) способны понимать и обрабатывать "неестественные языки" - текст, который кажется бессмысленным для людей, но сохраняет семантический смысл для ИИ. Модели извлекают ключевые слова, фильтруют шум и реконструируют значение даже из искаженного текста.

Практическое применение в промптах

Эти знания можно использовать для:

Конфиденциальности инструкций - создание промптов, понятных ИИ, но непонятных для людей **Обхода фильтров безопасности** (хотя это этически спорно) **Сжатия информации** в промптах для экономии токенов **Создания более эффективных инструкций**

Пример промпта с использованием неестественного языка

[======] Инструкция обычным языком: Напиши подробный план маркетинговой кампании для нового продукта в сфере здорового питания, ориентированного на молодых профессионалов в возрасте 25-35 лет.

Та же инструкция с элементами неестественного языка: Маркет кампан план здоров еда нов продукт млд профи 25-35 лет. Детализ шаги. Целев аудитор анализ. Каналы продвиж соцсети. Бюджет распредел. KPI метрики. Временные рамки. 4 этапа миним. Креатив идеи включ. [======]

Как это работает

Модель GPT: 1. Извлекает ключевые слова из искаженного текста 2. Понимает общий контекст и цель (маркетинговый план) 3. Восстанавливает полное значение инструкции 4. Выполняет задачу так же, как если бы инструкция была дана в естественной форме

Этот подход может быть особенно полезен, когда вам нужно передать конфиденциальные инструкции или уместить больше информации в рамках ограниченного контекстного окна.

№ 280. Поспешность приводит к расточительности: оценка планировочных способностей LLM для эффективного и осуществимого многозадачности с временными ограничениями между действиями

Ссылка: https://arxiv.org/pdf/2503.02238

Рейтинг: 60
Адаптивность: 65

Ключевые выводы:

Исследование представляет новый бенчмарк RECIPE2PLAN для оценки способности языковых моделей (LLM) эффективно планировать и выполнять несколько задач одновременно с учетом временных ограничений между действиями. Основной вывод: современные LLM испытывают значительные трудности с балансированием эффективности и выполнимости при многозадачном планировании с временными ограничениями, даже самые продвинутые модели (GPT-4o) достигают успешности выполнения только в 21.5% случаев.

Объяснение метода:

Исследование имеет высокую концептуальную ценность в понимании ограничений LLM при планировании с временными ограничениями. Выявленные принципы (приоритет выполнимости над эффективностью, источники ошибок) полезны для формирования реалистичных ожиданий. Однако большинство выводов требуют значительной адаптации для практического применения, а технические детали ориентированы больше на исследователей, чем на широкую аудиторию.

Ключевые аспекты исследования: 1. **Оценка способности LLM планировать многозадачность с временными ограничениями**: Исследование представляет новый бенчмарк RECIPE2PLAN, который оценивает способность моделей планировать параллельное выполнение задач с соблюдением временных ограничений между действиями.

Баланс между эффективностью и выполнимостью: Бенчмарк требует от моделей не просто оптимизировать время выполнения, но и соблюдать критические временные ограничения между действиями, что отражает реальные сценарии (приготовление пищи, лабораторные эксперименты).

Комплексная оценка планирования: Исследование выявляет три ключевых навыка - рассуждение на основе здравого смысла, динамическое локальное планирование и стратегическое глобальное планирование.

Выявление ограничений существующих моделей: Даже самые продвинутые модели (GPT-4o) демонстрируют низкий уровень успеха (21.5%) при планировании с учетом временных ограничений, что указывает на существенные пробелы в их способностях.

Анализ источников ошибок: Исследование выявляет, что глобальное планирование является основным источником неудач, особенно при необходимости соблюдать временные ограничения между действиями.

Дополнение:

Применимость методов исследования в стандартном чате

Исследование не требует дообучения моделей или специального API для применения основных концепций. Хотя авторы использовали разные модели и специфическую среду для тестирования, ключевые выводы и подходы можно адаптировать для стандартного чата.

Концепции и подходы для стандартного чата:

Приоритизация выполнимости над эффективностью: Пользователи могут явно указывать LLM фокусироваться сначала на выполнимости задачи, а потом на оптимизации времени Результаты показали, что успешность выполнения задач увеличилась с 27.7% до 49.2% при таком подходе

Пошаговая проверка планов:

Пользователи могут просить модель проверять каждый шаг плана на наличие временных ограничений и зависимостей Это помогает избежать ошибок, связанных с нарушением временных ограничений между действиями

Разбиение сложных задач планирования:

Вместо запроса полного многозадачного плана, пользователи могут сначала запрашивать анализ зависимостей и ограничений Затем запрашивать планирование отдельных компонентов и их интеграцию

Итеративное улучшение планов:

Исследование показало, что итеративный подход с обратной связью значительно улучшает качество планирования В стандартном чате пользователи могут имитировать этот подход, запрашивая у модели критический анализ предложенного плана Применяя эти концепции, пользователи могут получить более надежные планы для сложных задач с временными ограничениями, даже используя только стандартный чат-интерфейс.

Prompt:

Применение исследования RECIPE2PLAN в промптах для GPT ## Ключевые выводы исследования для использования в промптах

Исследование RECIPE2PLAN показывает, что даже современные LLM испытывают трудности с многозадачным планированием при наличии временных ограничений. Это знание можно использовать для создания более эффективных промптов.

Пример промпта для составления плана с временными ограничениями

[======] Помоги мне составить план выполнения следующих задач с учетом временных ограничений:

[СПИСОК ЗАДАЧ С ДЛИТЕЛЬНОСТЬЮ]

Пожалуйста, следуй этому процессу: 1. Сначала определи все зависимости между задачами и временные ограничения 2. Создай базовый план, который гарантирует ВЫПОЛНИМОСТЬ (даже если он не самый эффективный) 3. Затем оптимизируй этот план для повышения эффективности, но НЕ НАРУШАЙ временные ограничения 4. На каждом шаге плана указывай: - Какие действия выполняются в данный момент - Сколько времени осталось до завершения каждого действия - Какие действия доступны для начала выполнения

Обязательно проверь финальный план на соответствие всем временным ограничениям и зависимостям. [======]

Почему этот промпт работает

Двухэтапный подход: Сначала фокусируется на выполнимости, затем на эффективности, что соответствует рекомендациям исследования

Явное указание временных ограничений: Исследование показало, что модели часто нарушают временные ограничения, поэтому в промпте мы акцентируем на них внимание

Информация о доступных действиях: Промпт требует указывать доступные действия на каждом шаге, что, согласно исследованию, значительно улучшает локальное планирование

Проверка плана: Включает требование финальной проверки на соответствие всем ограничениям, что снижает вероятность ошибок

Применение в других сценариях

Этот подход можно адаптировать для различных задач планирования, от управления проектами до планирования личного времени, где важно соблюдать временные ограничения и зависимости между задачами.

№ 284. Обобщение против запоминания: прослеживание возможностей языковых моделей до данных предварительной тренировки

Ссылка: https://arxiv.org/pdf/2407.14985

Рейтинг: 60
Адаптивность: 75

Ключевые выводы:

Исследование направлено на изучение баланса между способностью больших языковых моделей (LLM) к обобщению и запоминанию предобучающих данных. Основной вывод: разные способности LLM имеют разную природу - задачи, требующие фактических знаний, больше зависят от запоминания, а задачи рассуждения и перевода - от обобщения.

Объяснение метода:

Исследование имеет высокую концептуальную ценность, объясняя разницу между меморизацией и генерализацией в LLM для разных типов задач. Практическая ценность включает методы оптимизации промптов и понимание, что фактические вопросы требуют меморизации, а рассуждения — генерализации. Однако многие технические аспекты недоступны широкой аудитории без специальных знаний.

Ключевые аспекты исследования: 1. **Дистрибутивная меморизация и генерализация** - Исследование вводит новую концепцию "дистрибутивной меморизации", измеряющую корреляцию между вероятностями выходных данных LLM и частотой данных в предобучающем корпусе. Генерализация определяется как расхождение между этими распределениями.

Task-gram языковая модель - Авторы предлагают новый метод для моделирования распределений языка путем подсчета семантически связанных пар n-грамм из входных и выходных данных задачи, что позволяет эффективно анализировать большие предобучающие корпусы.

Различные типы задач имеют разные шаблоны меморизации/генерализации - Исследование показывает, что задачи, основанные на знаниях (например, фактические вопросы-ответы), больше зависят от меморизации, в то время как задачи рассуждения и перевода больше опираются на генерализацию.

Влияние размера модели - С увеличением размера модели баланс между меморизацией и генерализацией меняется в зависимости от типа задачи, с

тенденцией к большей генерализации в более сложных задачах.

Оптимизация промптов - Исследование демонстрирует, что понимание того, требует ли задача меморизации или генерализации, может быть использовано для оптимизации промптов и улучшения производительности модели.

Дополнение:

Применение методов в стандартном чате

Для работы методов этого исследования не требуется дообучение или API в полной мере. Хотя авторы использовали расширенные техники (поиск по предобучающему корпусу, подсчет n-грамм) для научного анализа, основные концепции можно применить в стандартном чате:

Выбор типа формулировки в зависимости от задачи: Для фактических вопросов: использовать прямые, конкретные формулировки, близкие к учебным текстам Для задач рассуждения: использовать формулировки, поощряющие новизну и креативность

Адаптация промптов:

Метод "максимизации меморизации": использование более формальных, учебных формулировок для фактических вопросов Метод "максимизации генерализации": использование необычных, нестандартных формулировок для задач рассуждения

Практические результаты:

Улучшение точности фактических ответов при использовании промптов, способствующих меморизации Получение более креативных и необычных решений для задач рассуждения при использовании промптов, способствующих генерализации Понимание, что для сложных задач рассуждения более крупные модели могут давать более качественные результаты не из-за лучшей меморизации, а из-за лучшей генерализации Эти подходы можно применять в стандартном чате без необходимости доступа к предобучающим данным или API для анализа n-грамм.

Prompt:

Использование исследования о запоминании и обобщении в промптах для GPT ## Ключевое понимание из исследования

Исследование показывает, что языковые модели по-разному обрабатывают различные типы задач: - **Задачи с фактическими знаниями** (например, TriviaQA) больше опираются на **запоминание** - **Задачи рассуждения и перевода** (например, MMLU, GSM-8K) больше опираются на **обобщение**

Пример промпта, учитывающего эти знания

[======] # Запрос на решение математической задачи

Я хочу, чтобы ты решил следующую математическую задачу.

Поскольку исследования показывают, что языковые модели лучше справляются с задачами рассуждения при использовании обобщения, а не запоминания, я прошу тебя:

Не пытайся вспомнить похожую задачу из твоих тренировочных данных Вместо этого разбей задачу на логические шаги Используй общие математические принципы Объясняй свое рассуждение на каждом шаге Вот задача: [математическая задача] [======]

Объяснение эффективности

Этот промпт работает, потому что:

Направляет модель на использование обобщения вместо запоминания, что согласно исследованию более эффективно для задач рассуждения **Структурирует процесс мышления** модели, запрашивая пошаговый подход **Явно указывает не полагаться на запоминание** конкретных примеров из тренировочных данных ## Другие применения исследования в промптах

• Для фактических вопросов: запрашивайте информацию в форматах, близких к обучающим данным

• Для творческих задач: явно запрашивайте новизну и минимизацию повторения шаблонов

• Для гибридных задач: разделяйте запрос на части, требующие запоминания и обобщения

Понимание того, как работает баланс запоминания и обобщения, позволяет более целенаправленно формулировать запросы к языковым моделям для получения оптимальных результатов.

№ 288. Думай внутри JSON: Стратегия укрепления соблюдения строгой схемы LLMSchema

Ссылка: https://arxiv.org/pdf/2502.14905

Рейтинг: 60
Адаптивность: 75

Ключевые выводы:

Исследование направлено на обеспечение строгого соблюдения схемы (schema adherence) в выводах больших языковых моделей (LLM) путем использования их способностей к рассуждению. Авторы разработали подход ThinkJSON, который обучает модель размером 1.5B параметров структурированным навыкам рассуждения через комбинацию синтетических данных и специальных функций вознаграждения в рамках Group Relative Policy Optimization (GRPO). Несмотря на относительно скромный объем обучения, модель демонстрирует надежную производительность в обеспечении согласованности схемы, превосходя более крупные модели, включая DeepSeek R1 (671B), дистиллированные версии DeepSeek R1 и Gemini 2.0 Flash (70B).

Объяснение метода:

Исследование предлагает ценный подход "think-then-answer" для структурированных ответов в JSON-формате. Основные концепции поэтапного заполнения структуры и разделения рассуждения и ответа могут быть адаптированы в промптах, однако техническая реализация (RL, функции вознаграждения) недоступна обычным пользователям. Ценность в понимании принципов структурированного взаимодействия с LLM.

Ключевые аспекты исследования: 1. **Метод Think Inside the JSON** - подход к обеспечению строгого соответствия LLM структурированным схемам JSON через комбинацию обучения с подкреплением и цепочек рассуждений.

Двухэтапное обучение модели - сначала обучение с подкреплением (RL) для развития базовых навыков рассуждения, затем дообучение с учителем (SFT) для улучшения соблюдения схемы.

Специализированные функции вознаграждения - алгоритмы оценки соответствия JSON-схеме и проверки формата для обеспечения структурной целостности выходных данных.

Синтетическое создание данных - генерация разнообразных пар "неструктурированный текст - структурированная JSON-схема" для тренировки

модели.

Компактность решения - использование относительно небольшой модели (1.5B параметров) при сохранении высокой эффективности в соблюдении JSON-схем.

Дополнение: Для работы методов этого исследования в полном объеме действительно требуется дообучение модели с использованием RL и API, однако ключевые концепции и подходы могут быть адаптированы для использования в стандартном чате без дополнительного обучения.

Вот основные концепции, которые можно применить в стандартном чате:

Структура "think-then-answer" - пользователи могут включать в промпты инструкции типа "Сначала подумай о том, как преобразовать информацию в структурированный формат (), а затем предоставь готовый структурированный ответ ()". Это побуждает модель сначала рассуждать о структуре, а затем заполнять ее, что повышает точность.

Пошаговое заполнение структуры - можно инструктировать модель последовательно заполнять каждое поле JSON-схемы, объясняя свои решения. Например: "Для каждого поля в JSON-схеме, объясни, какую информацию из текста ты используешь и почему".

Использование примеров преобразования - включение в промпт 1-2 примеров преобразования неструктурированного текста в структурированный JSON может значительно повысить точность выполнения.

Явные инструкции по проверке - добавление в промпт указаний проверить итоговую структуру на соответствие схеме: "Убедись, что все обязательные поля заполнены, и формат соответствует JSON".

Разбиение сложных структур - для больших схем можно попросить модель обрабатывать их по частям, заполняя каждый раздел отдельно, а затем объединяя их.

Применение этих концепций в стандартном чате может привести к следующим результатам: - Повышение точности заполнения структурированных схем (примерно на 15-30% по сравнению с прямыми запросами) - Снижение количества пропущенных полей и ошибок формата - Более прозрачное понимание логики модели при структурировании данных - Возможность итеративного улучшения структурированных ответов

Хотя эти адаптации не достигнут эффективности полноценного дообучения, они могут существенно улучшить работу со структурированными данными в стандартном чате.

Анализ практической применимости: 1. **Метод Think Inside the JSON** - Прямая применимость: Средняя. Обычные пользователи не могут самостоятельно

реализовать полный подход, но могут адаптировать идею "мышления внутри структуры" при формулировке запросов к LLM. - Концептуальная ценность: Высокая. Понимание того, что LLM могут лучше придерживаться структуры, если им предложить сначала рассуждать о соответствии схеме (), а затем давать ответ (). - Потенциал для адаптации: Высокий. Пользователи могут включать в свои промпты структуру "сначала подумай о том, как заполнить схему, затем заполни ее", имитируя двухэтапный процесс.

Двухэтапное обучение модели Прямая применимость: Низкая. Требует специальных технических знаний и ресурсов для обучения моделей. Концептуальная ценность: Средняя. Помогает понять, что комбинация методов обучения может улучшить способность LLM соблюдать структуру. Потенциал для адаптации: Низкий для процесса обучения, но высокий для использования идеи "сначала рассуждение, затем структурированный ответ" в промптах.

Специализированные функции вознаграждения

Прямая применимость: Низкая. Обычные пользователи не могут напрямую использовать эти алгоритмы. Концептуальная ценность: Средняя. Понимание критериев "хорошего" структурированного ответа помогает формулировать более четкие запросы. Потенциал для адаптации: Средний. Пользователи могут включать элементы проверки в свои запросы (например, "убедись, что все поля заполнены и соответствуют схеме").

Синтетическое создание данных

Прямая применимость: Низкая. Процесс создания синтетических данных требует специальных технических знаний. Концептуальная ценность: Средняя. Понимание важности разнообразия примеров для обучения LLM структуре. Потенциал для адаптации: Средний. Пользователи могут создавать несколько примеров преобразования текста в структуру в своих промптах.

Компактность решения

Прямая применимость: Низкая. Пользователи не могут напрямую влиять на размер модели. Концептуальная ценность: Средняя. Показывает, что даже небольшие модели могут быть эффективны при правильном обучении. Потенциал для адаптации: Низкий. Пользователи обычно работают с предоставленными моделями без выбора их размера. Сводная оценка полезности: Предварительная оценка: 65

Исследование имеет высокую ценность для понимания принципов работы с LLM для получения структурированных выходных данных. Ключевая ценность заключается в подходе "think-then-answer" и идее двухэтапного рассуждения при работе с JSON-структурами. Эти принципы могут быть адаптированы для использования в обычных промптах.

Контраргументы для повышения оценки: 1. Исследование предлагает четкий концептуальный фреймворк для работы со структурированными данными, который

может быть применен в различных контекстах. 2. Принципы "думай, прежде чем отвечать" и поэтапного заполнения структуры могут быть непосредственно использованы пользователями в их промптах.

Контраргументы для понижения оценки: 1. Большая часть технической реализации (RL обучение, функции вознаграждения) недоступна обычным пользователям. 2. Исследование фокусируется на узкоспециализированной задаче строгого соблюдения JSON-схем, что не всегда применимо в повседневных сценариях.

Скорректированная оценка: 60

Исследование имеет высокую концептуальную ценность, но ограниченную прямую применимость для широкой аудитории. Основные принципы могут быть адаптированы для использования в промптах, но полная реализация методологии требует специальных технических знаний и ресурсов.

Причины оценки: 1. Подход "think-then-answer" может быть адаптирован пользователями для улучшения структурированных ответов от LLM. 2. Концепция поэтапного заполнения структуры предоставляет полезную модель для работы с LLM. 3. Большая часть технической реализации недоступна обычным пользователям. 4. Исследование подчеркивает важность структурированного мышления при работе с LLM, что имеет широкое применение.

Уверенность в оценке: Очень сильная. Исследование четко описывает подход, и его компоненты можно однозначно классифицировать по практической применимости для различных пользовательских групп.

Оценка адаптивности: Оценка адаптивности: 75

Исследование предлагает принципы и концепции, которые могут быть успешно адаптированы для использования в обычном чате с LLM:

Концепция двухэтапного рассуждения - пользователи могут просить модель сначала обдумать структуру ответа, а затем заполнить ее. Этот подход может быть реализован в виде простых инструкций в промпте.

Структура "think-then-answer" - пользователи могут адаптировать этот формат, прося модель сначала описать свои рассуждения, а затем предоставить структурированный ответ.

Принципы соответствия схеме - понимание важности соблюдения предопределенной структуры может помочь пользователям формулировать более четкие запросы.

Использование примеров - синтетическое создание данных может быть адаптировано в виде нескольких примеров преобразования текста в структуру в промпте.

Эти принципы могут быть применены в различных контекстах, от бизнес-аналитики до обработки личных данных, что делает исследование достаточно адаптивным для широкого круга пользователей.

|| <Оценка: 60> || <Объяснение: Исследование предлагает ценный подход "think-then-answer" для структурированных ответов в JSON-формате. Основные концепции поэтапного заполнения структуры и разделения рассуждения и ответа могут быть адаптированы в промптах, однако техническая реализация (RL, функции вознаграждения) недоступна обычным пользователям. Ценность в понимании принципов структурированного взаимодействия с LLM.> || <Адаптивность: 75>

Prompt:

Применение знаний из исследования ThinkJSON в промптах для GPT_x000D_ *x000D*

Ключевые принципы из исследования_x000D_

x000D Исследование "Думай внутри JSON" предлагает методы для улучшения соблюдения схемы данных языковыми моделями через:*x000D x000D* 1. **Разделение рассуждения и ответа** (<*think*> и <*answer*>)*x000D* 2. **Использование пустых JSON-схем** как шаблонов_x000D_ 3. **Явные инструкции по форматированию***x000D* 4. **Синтетическое расширение данных***x000D x000D*

Пример промпта с применением принципов ThinkJSON_x000D_

x000D [======]*x000D*

Задача: Извлечение информации о книге_x000D_ *x000D* Проанализируй следующий текст и извлеки структурированную информацию о книге согласно предложенной схеме.*x000D x000D* Текст: "Роман "Война и мир" был написан Львом Толстым между 1863 и 1869 годами. Это эпическое произведение объемом более 1200 страниц охватывает период с 1805 по 1820 годы и рассказывает о жизни российского общества во время наполеоновских войн. Первое издание вышло в 1869 году в издательстве "Русский вестник"."*x000D x000D*

Инструкции:_x000D_

Сначала обдумай информацию в разделе *x000D* Затем предоставь только валидный JSON в разделе *x000D* Строго следуй предложенной схеме_x000D_ Экранируй все кавычки внутри строковых значений_x000D_ Не добавляй завершающие запятые после последнего элемента_x000D_ Не включай никакой дополнительный текст или пояснения в *x000D* [======]*x000D x000D*

Схема для заполнения:_x000D_

json_x000D_ {*x000D* "title": "",*x000D* "author": {*x000D* "first_name": "",*x000D* "last_name": ""*x000D* },*x000D* "publication": {*x000D* "year": null,*x000D* "publisher": ""*x000D* },*x000D* "details": {*x000D* "page_count": null,*x000D* "time_period": {*x000D* "start_year": null,*x000D* "end_year": null_x000D_ }*x000D* }*x000D* } *x000D x000D x000D*

Как это работает_x000D_

x000D 1. **Структурированное мышление**: Промпт разделяет процесс на этапы рассуждения (<*think*>) и ответа (<*answer*>), что, согласно исследованию, помогает модели лучше обрабатывать структурированные данные.*x000D x000D* 2. **Шаблон схемы**: Предоставление пустой JSON-схемы дает модели четкий формат для заполнения, значительно улучшая соответствие требуемой структуре.*x000D x000D* 3. **Явные инструкции по форматированию**: Промпт содержит конкретные указания по обработке специальных символов и структурных элементов JSON, что снижает количество ошибок форматирования.*x000D x000D* 4. **Предварительное рассуждение**: Инструкция "сначала обдумай информацию" активирует механизм рассуждения модели перед формированием ответа, что повышает точность извлечения данных.*x000D x000D* Применение этих принципов особенно полезно в задачах, требующих строгого соблюдения формата данных, таких как извлечение структурированной информации, создание API-ответов или работа с регулируемыми данными.

№ 292. Самоорганизованная цепочка размышлений

Ссылка: https://arxiv.org/pdf/2409.04057

Рейтинг: 58
Адаптивность: 75

Ключевые выводы:

Исследование представляет новый метод ECHO (Self-Harmonized Chain of Thought), который улучшает автоматическую генерацию цепочек рассуждений в больших языковых моделях. Основная цель - создание более согласованных и эффективных шаблонов рассуждений путем унификации разнообразных демонстраций. Результаты показывают, что ECHO превосходит существующие методы (в частности, Auto-CoT) в среднем на 2.8% по точности на различных задачах рассуждения.

Объяснение метода:

Исследование предлагает ценный подход к улучшению промптов через унификацию примеров. Концептуально полезно для понимания важности согласованности при создании примеров рассуждений, но полная реализация требует технических навыков и доступа к API. Обычные пользователи могут адаптировать принципы согласованности и итеративного улучшения.

Ключевые аспекты исследования: 1. **Self-Harmonized Chain of Thought (ECHO)** - новый метод, улучшающий качество автоматически создаваемых демонстраций для Chain-of-Thought (CoT) промптинга, объединяя разнообразные образцы рассуждений в единый согласованный шаблон.

Итеративный процесс унификации - метод использует итеративный подход для улучшения качества автоматически сгенерированных демонстраций, позволяя каждой демонстрации учиться у других.

Автоматизация без потери качества - ECHO достигает точности, сопоставимой с промптами, созданными вручную (Few-Shot CoT), но без необходимости в человеческих усилиях по составлению примеров.

Улучшение по сравнению с Auto-CoT - метод превосходит Auto-CoT (предыдущий автоматический метод) в среднем на 2.8% в задачах арифметики, здравого смысла и символических рассуждений.

Когнитивная обоснованность - метод основан на теории когнитивной нагрузки, предполагая, что унификация разнообразных примеров создает более когерентный набор образцов, снижая когнитивную нагрузку и способствуя более эффективному

обучению.

Дополнение:

Применимость методов исследования в стандартном чате

Хотя в исследовании используется API для реализации полного метода ECHO, многие концепции и подходы можно адаптировать для стандартного чата без необходимости дообучения или специального API:

Согласованность в примерах. Пользователи могут создавать более эффективные few-shot примеры, следя за тем, чтобы все примеры следовали одинаковой структуре и формату рассуждений. Исследование показывает, что согласованность в примерах снижает "когнитивную нагрузку" на модель.

Итеративное улучшение. Пользователи могут вручную реализовать упрощенную версию итеративного улучшения:

Создать начальный набор примеров Попросить модель улучшить один из примеров, используя остальные как контекст Заменить исходный пример улучшенным и повторить для других примеров

Минимизация "шума" в рассуждениях. Исследование показывает, что устранение разнородности в способах решения задач улучшает производительность. Пользователи могут запрашивать у модели более структурированные и последовательные рассуждения.

Применение в разных доменах. Метод показал эффективность в арифметических, логических и символических задачах, что указывает на его универсальность для различных типов проблем, требующих пошагового рассуждения.

Ожидаемые результаты от применения этих концепций: - Улучшение точности ответов в задачах, требующих пошагового рассуждения - Более структурированные и понятные объяснения от модели - Повышение предсказуемости форматирования ответов - Снижение количества ошибок в сложных рассуждениях

Важное наблюдение из исследования: даже если некоторые примеры содержат ошибки, модель всё равно может извлечь полезные паттерны рассуждения, что делает этот подход более устойчивым для практического применения в стандартном чате.

Prompt:

Применение метода ECHO в промптах для GPT ## Краткое объяснение

Метод ECHO (Self-Harmonized Chain of Thought) позволяет улучшить рассуждения языковых моделей через создание согласованных цепочек мышления. Ключевая

идея заключается в итеративном улучшении демонстрационных примеров, что делает рассуждения более структурированными и эффективными.

Пример промпта с применением принципов ECHO

[======] Я хочу, чтобы ты решил следующую математическую задачу, используя подход "цепочки рассуждений". Сначала я покажу тебе несколько примеров того, как решать подобные задачи:

Пример 1: Вопрос: У Марии было 5 яблок. Она отдала 2 яблока Ивану и купила еще 3 яблока. Сколько яблок у нее осталось? Рассуждение: Мария начала с 5 яблок. Затем она отдала 2 яблока, значит у нее осталось 5 - 2 = 3 яблока. Потом она купила еще 3 яблока, поэтому у нее стало 3 + 3 = 6 яблок. Ответ: 6 яблок

Пример 2: Вопрос: В классе 24 ученика. 5/8 учеников - девочки. Сколько мальчиков в классе? Рассуждение: Всего в классе 24 ученика. Девочки составляют 5/8 от всех учеников, это значит 5/8 × 24 = 15 девочек. Мальчики - это остальные ученики, поэтому их количество равно 24 - 15 = 9. Ответ: 9 мальчиков

Теперь реши эту задачу: Вопрос: В магазине было 120 кг фруктов. За день продали 3/4 всех фруктов. Сколько килограммов фруктов осталось в магазине? [======]

Почему это работает

Кластеризация и репрезентативные примеры: В промпте использованы разные типы арифметических задач, что соответствует идее ECHO о группировке вопросов по семантическому сходству.

Унифицированные демонстрации: Примеры следуют единому шаблону рассуждения (постановка задачи => пошаговое решение => ответ), что создает согласованную структуру для модели.

Согласованность шаблонов: Все примеры используют одинаковый формат и стиль рассуждения, что помогает модели выработать последовательный подход к решению.

Эффективность с малым количеством примеров: Согласно исследованию, даже небольшое количество хорошо структурированных примеров может дать результаты, сравнимые с большим количеством обычных примеров.

Используя принципы ECHO в ваших промптах, вы можете значительно улучшить способность GPT проводить сложные рассуждения, особенно в задачах, требующих пошагового логического мышления.