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

Дата: 2025-01-09 00:00:00

Ссылка на исследование: 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.

Анализ практической применимости: 1. **Бенчмарк функциональных вызовов API** - Прямая применимость: Средняя. Сам бенчмарк не может быть напрямую использован обычными пользователями, но демонстрирует, какие модели лучше справляются с вызовами API. - Концептуальная ценность: Высокая. Помогает пользователям понять, что современные LLM могут эффективно взаимодействовать с API и какие ограничения существуют при усложнении задач. - Потенциал для

адаптации: Высокий. Структура тестов может быть адаптирована для оценки моделей в конкретных пользовательских сценариях.

Градация сложности задач Прямая применимость: Средняя. Пользователи могут учитывать сложность своих задач при выборе модели. Концептуальная ценность: Высокая. Понимание того, что сложность задачи существенно влияет на производительность модели, помогает формировать более реалистичные ожидания. Потенциал для адаптации: Высокий. Пользователи могут разбивать сложные задачи на более простые шаги для повышения эффективности.

Метрики оценки и стабильности

Прямая применимость: Низкая. Обычные пользователи редко будут вычислять метрики стабильности. Концептуальная ценность: Высокая. Понимание, что стабильность вывода модели важна для практических приложений. Потенциал для адаптации: Средний. Пользователи могут проверять стабильность ответов путем повторных запросов.

Методы оптимизации

Прямая применимость: Высокая. Разделение сложной задачи на этапы (сначала выбор API, затем параметры) может быть напрямую использовано пользователями. Концептуальная ценность: Высокая. Демонстрирует эффективные стратегии для улучшения результатов при работе с LLM. Потенциал для адаптации: Очень высокий. Подходы могут быть применены к различным задачам, требующим структурированного вывода.

Сравнительный анализ моделей

Прямая применимость: Высокая. Пользователи могут выбирать модели, лучше подходящие для работы с API. Концептуальная ценность: Средняя. Показывает разрыв в возможностях между коммерческими и открытыми моделями. Потенциал для адаптации: Средний. Результаты сравнения помогают в выборе модели, но требуют учета конкретных задач.

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.