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

Дата: 2025-02-10 00:00:00

Ссылка на исследование: 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

Анализ практической применимости: **Методология оценки LLM-as-a-judge** - Прямая применимость: Высокая для пользователей, которым нужно оценить качество генерируемого кода или текста. Особенно полезно понимание, что output-based методы (прямой запрос LLM оценить ответ) наиболее эффективны и просты в применении. - Концептуальная ценность: Значительная - пользователи узнают о различных подходах к автоматической оценке и их ограничениях, что помогает выбирать правильный подход. - Потенциал для адаптации: Высокий - пользователи могут адаптировать эти методы для оценки собственного кода или для сравнения различных решений.

Эмпирические результаты по задачам - Прямая применимость: Высокая - исследование четко показывает, для каких задач какие методы оценки работают лучше всего, что позволяет пользователям выбрать оптимальный подход. - Концептуальная ценность: Средняя - понимание того, что эффективность методов зависит от типа задачи, помогает формировать реалистичные ожидания. - Потенциал для адаптации: Средний - пользователи могут экстраполировать результаты на схожие задачи.

Различия в эффективности методов - Прямая применимость: Высокая - результаты показывают, что output-based методы с большими моделями (GPT-4o, Deep Seek v2.5) обеспечивают наилучшую согласованность с человеческими оценками. - Концептуальная ценность: Высокая - понимание того, что размер модели имеет значение, а сложные стратегии вывода дают лишь незначительное улучшение. - Потенциал для адаптации: Средний - пользователи могут предпочесть использовать более крупные модели для оценки вместо более сложных методов.

Попарное сравнение - Прямая применимость: Средняя - исследование показывает, что индивидуальная оценка более надежна, чем попарное сравнение, что может повлиять на выбор метода оценки. - Концептуальная ценность: Высокая - понимание ограничений попарного сравнения важно для пользователей. - Потенциал для адаптации: Низкий - результаты скорее предостерегают от использования попарного сравнения.

Анализ распределения оценок - Прямая применимость: Низкая - это скорее академический аспект исследования. - Концептуальная ценность: Средняя - понимание того, как распределяются оценки, может помочь интерпретировать результаты. - Потенциал для адаптации: Низкий - сложно адаптировать этот аспект для практического использования.

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.