

Агентная репродукция ошибок для эффективного автоматизированного исправления программ в Google

Дата: 2025-02-03 00:00:00

Ссылка на исследование: <https://arxiv.org/pdf/2502.01821>

Рейтинг: 65

Адаптивность: 75

Ключевые выводы:

Основная цель исследования - разработка и оценка методов автоматической генерации тестов воспроизведения ошибок (BRT) в промышленной среде Google. Главные результаты: разработан подход BRT Agent на основе LLM, который значительно превосходит существующий метод LIBRO, достигая 28% успешной генерации BRT против 10% у LIBRO на 80 реальных ошибках из внутреннего трекера Google. Интеграция сгенерированных BRT с системой автоматического исправления программ (APR) увеличила количество исправленных ошибок на 30%.

Объяснение метода:

Исследование предлагает ценный агентный подход к использованию LLM для генерации тестов воспроизведения ошибок, который может быть адаптирован для эффективного взаимодействия с LLM в разных контекстах. Метрика EPR для отбора лучших вариантов универсально полезна. Однако, полное применение требует значительных технических знаний и адаптации для использования вне промышленной среды разработки.

Ключевые аспекты исследования: 1. **Агентный подход к воспроизведению ошибок:** Исследование представляет BRT Agent — агентную систему на базе LLM для автоматического создания тестов воспроизведения ошибок (Bug Reproduction Tests, BRTs) из описаний багов.

Улучшение автоматического исправления программ: Авторы показывают, что сгенерированные BRTs значительно повышают эффективность системы автоматического исправления программ (APR) Passerine, увеличивая количество успешно исправленных ошибок на 30%.

Метрика Ensemble Pass Rate (EPR): Предложена новая метрика для отбора наиболее перспективных исправлений из множества сгенерированных APR-системой, основанная на проценте прохождения тестов из набора

сгенерированных BRTs.

Промышленное применение: Исследование фокусируется на применении в реальной промышленной среде Google, работая с закрытым кодом и реальными багами из внутренней системы отслеживания проблем.

Сравнительная оценка: BRT Agent значительно превосходит базовый подход LIBRO, генерируя правдоподобные BRTs для 28% ошибок по сравнению с 10% у LIBRO, при этом работая с шестью различными языками программирования.

Дополнение: Для работы методов этого исследования в полном объеме действительно требуется специализированная инфраструктура и потенциально API, поскольку BRT Agent взаимодействует с окружением через набор команд (просмотр файлов, поиск кода, запуск тестов). Однако, многие ключевые концепции и подходы можно адаптировать для использования в стандартном чате с LLM.

Концепции, применимые в стандартном чате:

Структурированное рассуждение и планирование: Пользователь может попросить LLM сначала проанализировать описание ошибки, затем спланировать шаги для создания теста, имитируя процесс рассуждения агента.

Пошаговый подход к сложной задаче: Вместо попытки сразу сгенерировать тест, пользователь может разбить задачу на шаги: анализ ошибки → определение тестируемой функциональности → написание теста → рефакторинг.

Итеративное улучшение: Пользователь может имитировать цикл "редактирование → проверка" через последовательные запросы, где LLM генерирует тест, а пользователь предоставляет обратную связь о его работе.

Метрика Ensemble Pass Rate: Можно попросить LLM сгенерировать несколько вариантов решения проблемы, а затем использовать простые тесты для выбора лучшего варианта, следуя принципу EPR.

Использование контекста для улучшения генерации: Исследование показывает, что предоставление соответствующих файлов (тестовых файлов, кода с ошибкой) значительно улучшает качество генерации – этот принцип применим в стандартном чате.

Возможные результаты от применения этих концепций: - Более качественная генерация тестов и кода через структурированный пошаговый подход - Лучшее понимание ошибок и путей их решения через формализацию процесса анализа - Возможность выбора лучшего варианта из нескольких сгенерированных решений - Эффективное использование контекста для улучшения результатов генерации

Хотя полная реализация агентной системы требует специальной инфраструктуры, основные принципы рассуждения, планирования и итеративного улучшения можно успешно применять в стандартном чате с LLM.

Анализ практической применимости: 1. **Агентный подход к воспроизведению ошибок:** - **Прямая применимость:** Высокая для разработчиков и тестировщиков, которые могут использовать этот подход для автоматизации создания тестов на основе описаний багов, что экономит время на ручное воспроизведение ошибок. - **Концептуальная ценность:** Значительная — демонстрирует возможности агентной архитектуры LLM для решения сложных задач разработки ПО через последовательность действий (поиск кода, редактирование, тестирование). - **Потенциал для адаптации:** Высокий — принцип агентной системы с набором действий и рассуждений можно адаптировать для взаимодействия с LLM в других контекстах разработки.

Улучшение автоматического исправления программ: **Прямая применимость:** Средняя — требует интеграции с APR-системой, но сам принцип использования тестов для улучшения исправлений применим широко. **Концептуальная ценность:** Высокая — показывает, как дополнительный контекст (в виде тестов) может значительно улучшить работу LLM при генерации кода. **Потенциал для адаптации:** Значительный — идея предоставления дополнительного контекста в виде тестов может быть применена к любым задачам генерации кода.

Метрика Ensemble Pass Rate (EPR):

Прямая применимость: Высокая — можно использовать для оценки и ранжирования любых сгенерированных исправлений или решений. **Концептуальная ценность:** Значительная — предлагает эффективный способ отбора наиболее перспективных вариантов из множества сгенерированных LLM. **Потенциал для адаптации:** Очень высокий — принцип оценки через прохождение набора тестов может быть адаптирован для многих задач выбора лучшего варианта из нескольких.

Промышленное применение:

Прямая применимость: Ограниченная для обычных пользователей — исследование ориентировано на промышленную среду разработки. **Концептуальная ценность:** Высокая — демонстрирует, что LLM могут эффективно работать с реальными, сложными проектами в промышленной среде. **Потенциал для адаптации:** Средний — некоторые методы могут быть сложны для адаптации вне промышленной среды, но общие принципы применимы.

Сравнительная оценка:

Прямая применимость: Низкая — результаты сравнения сами по себе не предоставляют практических инструментов. **Концептуальная ценность:** Средняя — показывает преимущества агентного подхода над прямым использованием LLM. **Потенциал для адаптации:** Средний — понимание сильных сторон агентного подхода может быть перенесено на другие задачи.

Prompt:

Использование знаний из исследования BRT Agent в промптах для GPT ##
Ключевые концепции для промптов

Исследование о BRT Agent предоставляет ценные знания, которые можно применить при составлении промптов для работы с кодом:

Агентный подход - структурирование взаимодействия с LLM через специфические команды **Контекстуализация ошибок** - предоставление богатого контекста для понимания проблемы **Генерация тестов** - создание воспроизводимых тестовых случаев **Итеративное улучшение** - пошаговое уточнение решений ## Пример промпта для GPT

[=====] # Задача: Генерация теста для воспроизведения ошибки

Контекст ошибки [Здесь вставьте описание ошибки, включая сообщение об ошибке, стек вызовов если доступен]

Код с ошибкой [=====]java [Вставьте проблемный код] [=====]

Инструкции Действуй как агент для создания теста воспроизведения ошибки (BRT), следуя этим шагам:

Анализ: Изучи код и описание ошибки. Определи потенциальный источник проблемы. **Исследование:** Укажи, какие части кодовой базы тебе нужно дополнительно изучить (классы, методы, зависимости). **Планирование:** Опиши стратегию создания минимального теста для воспроизведения ошибки. **Генерация:** Создай модульный тест, который: Минимален и фокусируется только на воспроизведении ошибки Включает необходимые импорты и настройку окружения Содержит четкие комментарии о том, как тест демонстрирует ошибку **Проверка:** Объясни, как твой тест воспроизводит исходную ошибку и почему он должен работать. В своем ответе используй структурированный подход, подобный агентному методу BRT Agent из исследования Google. [=====]

Почему это работает

Данный промпт использует ключевые принципы из исследования BRT Agent:

- Структурированный агентный подход: Разбивает процесс на четкие этапы, аналогично тому, как BRT Agent использует команды для взаимодействия с кодовой базой
- Богатый контекст: Запрашивает полное описание ошибки и код, что помогает LLM лучше понять проблему
- Целенаправленность: Фокусируется на создании минимального теста для воспроизведения ошибки

- Пошаговое планирование: Включает этап планирования перед генерацией кода, что соответствует методологии BRT Agent

Такой подход повышает вероятность получения качественного результата, как показало исследование, где агентный метод превзошел традиционные подходы в 2.8 раза (28% против 10%).