УДК 004.89

Исследование возможности применения LLM для генерации персонализированных подсказок на платформе Code&Test

Кривоносов Е.Д. (Университет ИТМО)
Научный руководитель – доцент, кандидат технических наук, Перл И.А. (Университет ИТМО)

Введение.

В современном образовательном процессе в области программирования особое значение приобретает автоматизированная проверка заданий и формирование конструктивной обратной связи [1]. Платформа Code&Test активно применяется для тестирования кода студентов во время обучения, однако традиционные системы зачастую не учитывают индивидуальные особенности учащихся. Современные большие языковые модели (LLM, Large Language Models) представляют собой нейросетевые модели, обученные на огромном объёме текстовых данных, способные генерировать, интерпретировать и анализировать текстовую информацию. Такие модели позволяют не только отвечать на вопросы, но и создавать подробные отчёты, давать рекомендации по исправлению ошибок и рефакторингу кода. В рамках исследования проведён анализ зарубежного опыта применения ИИ в образовательных системах, а также возможностей объединения статического и динамического анализа кода с генеративными моделями для формирования персонализированных подсказок [2].

Особое внимание уделяется соблюдению единообразного стиля кода. Применение рекомендаций из Google Style Guides [3] обеспечивает повышение читаемости, структурированности и поддерживаемости программного кода, что является критически важным как для образовательного процесса, так и для разработки качественных программных решений.

Основная часть.

Ключевым нововведением предлагаемой системы является персонализация подсказок, достигаемая за счёт анализа не только исходного кода, но и истории решений студента. Такой подход способствует формированию индивидуальной траектории обучения и помогает выявлять закономерности в допущенных ошибках.

Использование статического анализа позволяет:

– проводить подробное выявление синтаксических и семантических ошибок;

– осуществлять подсветку проблемных участков кода с детальной расшифровкой обнаруженных нарушений;

– в совокупности с большими языковыми моделями автоматически составлять отчёты, в которых перечисляются ошибки, выявленные при проверке, и формируются рекомендации по рефакторингу.

Динамический анализ, в свою очередь, обеспечивает сбор метрик выполнения, таких как время исполнения и использование памяти, что даёт объективную оценку производительности решений.

Кроме того, система собирает сводку типичных ошибок, обнаруженных во всех работах студентов, и предоставляет преподавателю полную картину для дальнейшего разбора на занятиях.

Особое внимание уделено выбору подходящей генеративной модели с учётом предъявляемых ограничений к проекту Code&Test. Возможные модели, рассмотренные в

рамках исследования, включают: GPT-4o, GPT-4o-mini, GPT-o1, GPT-o1-mini, Gemini 1.5, Gemini 2.0, Luma AI, Llama, Grok, Claude, YandexGPT 4, а также другие локальные модели для запуска через Ollama.

Основные ограничения заключаются в следующем:

– ограниченная доступность LLM на территории РФ (некоторые зарубежные модели могут быть заблокированы);

– необходимость использования моделей, способных обрабатывать большие объёмы токенов для формирования развернутых отчётов;

– обязательное составление подробных инструкций до начала генерации, что обеспечивает корректную интерпретацию входных данных;

– невозможность применения локальных решений ввиду ограниченных вычислительных ресурсов в учебной лаборатории;

– а также ограничение на бесплатное использование: для образовательного процесса критически важно использование моделей, доступных на бесплатном тарифе или с минимальными затратами, поскольку дополнительные расходы могут негативно сказаться на масштабировании проекта.

Эти требования диктуют выбор модели, способной обеспечить стабильную и качественную генерацию рекомендаций в условиях ограниченной доступности и высоких нагрузок на систему.

Выводы.

Проведённое исследование показало, что применение LLM для генерации персонализированных подсказок на платформе Code&Test имеет значительный потенциал для повышения качества образовательного процесса. Автоматизация формирования рекомендаций позволяет оперативно выявлять ошибки, предоставлять детальную обратную связь и предлагать конкретные пути для улучшения кода. Такой подход способствует систематизации знаний студентов и развитию индивидуальной траектории обучения.

Особое внимание уделено выбору оптимальной модели с учётом ограничений, связанных с доступностью сервисов и необходимостью обработки больших объёмов данных. В дальнейшем планируется провести масштабное тестирование разработанного решения на реальных студенческих работах, что позволит оценить его эффективность в условиях действующего образовательного процесса и выявить дополнительные возможности для оптимизации.

Список использованных источников:

1. Schiewe M. et al. Advancing static code analysis with language-agnostic component identification //IEEE Access. – 2022. – Т. 10. – С. 30743-30761.
2. Ludwig J., Cline D. Challenges in explaining source code quality assessment //2022 IEEE Aerospace Conference (AERO). – IEEE, 2022. – С. 1-7.
3. Google Style Guides. Google Python Style Guide, Google C++ Style Guide, Google Java Style Guide. – URL: https://google.github.io/styleguide/ (дата обращения: 27.12.2024).