

УДК 004.588

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ДЛЯ ГЕНЕРАЦИИ ТЕСТОВ НА ОСНОВЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Нормахмадов И.У., студент гр. 4416, IV курс

Научный руководитель: Сабитов Р.А., доцент кафедры динамики
процессов и управления

Казанский Национальный Исследовательский Технический
Университет им. А. Н. Туполева – КАИ

г. Казань

Аннотация: в статье рассматривается подход к разработке интеллектуальной веб-системы для генерации и оценки тестовых заданий по программированию с использованием технологий искусственного интеллекта. Рассматриваются преимущества применения больших языковых моделей в обучении, включая автоматическое формирование заданий, персонализацию обучения и адаптацию сложности под уровень пользователя. В системе реализованы методы генерации трёх типов заданий: тестовых с вариантами ответов, открытых вопросов и практических задач, требующих написания программного кода. Архитектура системы реализована по модульному принципу с использованием FastAPI и React и включает отдельные компоненты для генерации заданий через OpenAI, проверки решений, анализа кода и формирования рекомендаций на основе выявленных ошибок пользователя.

Ключевые слова: искусственный интеллект, генерация тестов, программирование, адаптивное обучение, персонализация, большие языковые модели, образовательные технологии, FastAPI, MySQL, React, OpenAI, анализ кода, обратная связь, веб-система.

Введение

Цифровая трансформация образования актуализирует задачу разработки инструментов для оперативной и объективной оценки компетенций учащихся [9]. Традиционные способы проверки знаний, включая письменные работы и устные ответы, требуют значительных временных затрат и во многом зависят от субъективной оценки преподавателя, что затрудняет обеспечение единых стандартов оценивания [1]. Особенно это актуально для обучения программированию, где необходимо оценивать не только теоретические знания, но и практические навыки написания кода [4].

В этих условиях автоматизация процессов генерации и проверки заданий обеспечивает масштабируемость системы оценки и предоставляет пользователям быструю обратную связь [6]. Применение искусственного интеллекта, в том числе больших языковых моделей, позволяет создавать системы, адаптирующие содержание заданий под уровень подготовки пользователя [5].

В статье рассматривается архитектура системы автоматической генерации и оценки заданий по программированию. Также представлены теоретические основы применения ИИ в образовании, архитектура на базе микросервисов, методы генерации заданий трёх типов и перспективы внедрения системы в учебный процесс.

Искусственный интеллект в образовательных системах

В образовательных платформах активно применяются методы машинного обучения и языковые модели, обеспечивающие автоматизацию ключевых процессов. Это позволяет формировать задания и изменять их сложность с учётом уровня подготовки пользователя. Подобные системы используют алгоритмы машинного обучения для подбора заданий и анализа ответов, что повышает вовлечённость обучающихся [1]. В разрабатываемой системе данный подход реализован за счёт адаптивной настройки сложности заданий на основе результатов пользователя.

Ряд платформ формирует персонализированные траектории обучения, однако такие решения требуют значительных вычислительных ресурсов и большого объёма данных [4].

В отечественных разработках чаще применяются более простые подходы. В частности, генерация заданий по программированию нередко основана на шаблонах, что упрощает реализацию, но ограничивает гибкость системы [5]. Также применяются инструменты автоматической проверки кода, эффективность которых зависит от качества используемых данных и алгоритмов [2].

В результате существующие системы характеризуются высокой сложностью, ограниченной адаптивностью и отсутствием универсальных решений. Это показывает необходимость разработки более гибких и эффективных систем тестирования.

Специфика обучения программированию

Процесс изучения программирования включает освоение синтаксиса языка, развитие алгоритмического мышления, а также формирование навыков анализа, отладки и оптимизации программных решений [13]. В отличие от дисциплин с фиксированными ответами, задачи по программированию могут иметь несколько корректных решений, что усложняет автоматическую

проверку [14]. В предлагаемой системе эта проблема решается за счёт комбинации статического анализа кода и семантической проверки через LLM.

Практические компетенции в контексте изучения программирования представляют собой способность учащихся применять теоретические знания для написания рабочего кода, анализа ошибок и интерпретации результатов [3]. Такие компетенции включают навыки проектирования, реализации, тестирования и рефакторинга кода, что делает их ключевым элементом современного IT-образования.

Методы искусственного интеллекта для генерации тестовых заданий

Применение методов искусственного интеллекта позволяет автоматизировать процесс создания тестовых заданий, что особенно важно в обучении программированию. В результате снижается нагрузка на преподавателя и обеспечивается генерация большого числа заданий различного уровня сложности. Основные методы включают шаблонные подходы, методы обработки текста и генеративные модели.

Шаблонные методы

Шаблонные методы основаны на использовании заранее заданных структур, в которые подставляются изменяемые параметры. Такой способ часто используют для задач по программированию - можно менять входные данные, но сама логика задания остаётся неизменной [6]. Данный подход даёт стабильный результат, но ограничивает разнообразие создаваемых заданий [6].

Методы обработки текста

Методы обработки текста позволяют формировать задания на основе текстовой информации, что особенно полезно для генерации теоретических вопросов и описаний задач [7]. Главным достоинством такого подхода является гибкость и возможность создавать разнообразные текстовые задания, однако реализация этих методов требует значительных вычислительных ресурсов и может приводить к неточным формулировкам [8]. Поэтому их чаще применяют для теоретических вопросов и текстовых заданий, где важна смысловая нагрузка, а не строгая алгоритмическая структура.

Генеративные модели

Генеративные модели формируют задания на основе заданного контекста и широко применяются при разработке упражнений по программированию, включая вопросы с развёрнутым ответом и практические задачи [9]. Согласно исследованиям, «локальные языковые модели могут использоваться для создания экзаменационных заданий и их автоматической проверки с высокой точностью» [9]. Ключевым преимуществом таких моделей является способность генерировать разнообразные и уникальные задания, однако их применение сопровождается риском непредсказуемых результатов, что требует

дополнительной верификации. Подобные решения особенно востребованы при работе со сложными задачами и в адаптивных системах тестирования.

Сравнительный анализ показывает, что шаблонные методы проще в реализации и обеспечивают более стабильный результат, однако приводят к снижению вариативности заданий [6]. Генеративные модели, напротив, обеспечивают высокую гибкость, но требуют значительных вычислительных ресурсов и экспертной проверки результатов [9]. В практических системах чаще применяется комбинированный подход, объединяющий надёжность шаблонных методов и гибкость генеративных моделей.

Подходы к автоматической оценке результатов

Автоматизированная проверка с использованием ИИ обеспечивает быструю и независимую оценку результатов обучения. В случае тестов с готовыми вариантами ответов обычно используют классификационные алгоритмы - они отличаются высокой точностью и быстрым выполнением [12]. Анализ публикаций свидетельствует о высокой востребованности подобных решений в массовых онлайн-курсах, где критически важны скорость и объективность оценки [12]. При этом такой подход имеет свои ограничения: он эффективен только для заданий с выбором из предложенных вариантов и не подходит для вопросов, требующих развёрнутого письменного ответа или практического решения.

Анализ кода и текстовых ответов

Для задач по программированию используют методы анализа кода и текстовых ответов, которые учитывают не только правильность, но и структуру решения [13]. Научные данные указывают на то, что «комбинация правил и моделей машинного обучения повышает точность оценки, но требует тщательной настройки под предметную область» [14]. Такие методы более гибкие и позволяют проверять сложные практические задания, но их реализация сложнее, а результаты могут быть менее стабильными.

Формирование рекомендаций

Современные образовательные платформы всё чаще внедряют ИИ для создания индивидуальных рекомендаций. Анализируя ответы пользователя, платформа способна находить повторяющиеся ошибки и подсказывать, над чем стоит поработать [15]. Согласно научным данным, «эффективность таких систем подтверждена в исследованиях адаптивного обучения, однако их внедрение требует накопления репрезентативных данных» [15]. Благодаря этому обучение становится более персонализированным и результативным, но для этого нужно собирать и обрабатывать значительные массивы информации.

В итоге получается, что проверка тестов с готовыми вариантами ответов выполняется быстро и без ошибок, но такой способ работает не для всех типов заданий [12]. Методы, которые анализируют код и развёрнутые текстовые

ответы, справляются со сложными задачами, но их настройка требует больше усилий, а результаты могут быть менее предсказуемыми [13]. По этой причине на практике чаще всего комбинируют несколько подходов, подбирая подходящий вариант под конкретный тип вопроса и уровень подготовки студента.

Пробелы и недостатки существующих решений

Анализ научных работ показывает, что исследователи по-разному оценивают существующие проблемы: одни выделяют высокую техническую сложность и значительные вычислительные затраты моделей [16], другие - недостаточную гибкость систем, их неспособность адаптироваться под конкретного пользователя и разобщённость функциональных компонентов [17]. Применение продвинутых алгоритмов требует существенных вычислительных ресурсов, что ограничивает их внедрение в условиях недостаточной инфраструктуры [16]. Во многих случаях платформы используют упрощённые шаблонные решения, не обеспечивающие полноценной персонализации обучения [17]. Кроме того, значительная часть сервисов не предоставляет детализированного анализа ошибок, ограничиваясь итоговой оценкой [18]. В результате недостаток качественной обратной связи и развитой аналитики снижает эффективность обучения [18].

Перспективы и предложения для разработки интеллектуальной системы

Современные исследования предлагают несколько направлений для решения этих проблем. Одни специалисты советуют совмещать разные способы создания заданий, чтобы сохранить стабильность работы системы и при этом получить разнообразный контент [19]. Другие фокусируются на развитии адаптивных платформ, которые подстраиваются под успехи конкретного ученика и меняют программу обучения на основе его прогресса [20]. В итоге такой модульный подход позволяет создавать гибкие веб-платформы, которые можно адаптировать под задачи конкретного учебного заведения.

Заключение

Разработанная платформа обеспечивает автоматизацию генерации и проверки заданий по программированию с использованием искусственного интеллекта и может применяться для развития образовательных подходов [5]. Снижается нагрузка на преподавателей и ускоряется получение обратной связи, что положительно влияет на усвоение материала [4]. Используемый технологический стек обеспечивает стабильную работу и высокую производительность системы. Генерация заданий выполняется через API OpenAI с последующей обработкой на стороне FastAPI, что позволяет контролировать качество результатов при сохранении вариативности.

Практическая значимость решения связана с уменьшением времени проверки и предоставлением индивидуализированной обратной связи. В режиме «укрепление слабых тем» система обрабатывает статистику ошибок пользователя, выделяет проблемные области и формирует запрос к OpenAI, на основе которого создаётся персонализированный набор заданий. Перспективы развития включают применение более сложных моделей для анализа структуры программного кода и архитектурных решений, а также оценку влияния системы на результаты обучения. Предложенный подход может использоваться для повышения доступности, объективности и практической направленности образовательного процесса.

Список литературы

1. Коваленко Н.В. ИИ в образовательном контенте: тренды и практика // Образование и наука. 2021. №2(4). С. 28–35. URL: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-obrazovatelnom-kontente-aktualnyy-trend-i-prakticheskie-aspekty-evolyutsii-uchebnogo-protsessa>
2. Борисов А.В. Языковые модели в образовании РФ: проблемы и перспективы // Вестник науки и образования. 2023. №2(8). С. 45–52. URL: <https://cyberleninka.ru/article/n/problemny-i-perspektivy-rossiyskogo-obrazovaniya-v-novyh-usloviyah>
3. Иванов А.В. Алгоритмы искусственного интеллекта в образовательных системах // Современные технологии в образовании. 2021. №3(5). С. 34–41. URL: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-obrazovanii-obzor-vozmozhnostey-i-ogranicheniy>
4. Яковлев Д.С. Применение ИИ для адаптивного обучения // Образование и наука. 2022. №3(5). С. 34–41. URL: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-obrazovanii-primeneniye-iskusstvennogo-intellekta-dlya-obespecheniya-adaptivnosti-obrazovaniya>
5. Самигулина Г.А. Ограничения ИИ в образовательных системах // Вестник КазНУ. 2022. №1(87). С. 56–63. URL: <https://cyberleninka.ru/article/n/potentsial-iskusstvennogo-intellekta-v-obuchenii>
6. Голиков А.В. Автоматизация генерации тестовых заданий // Наука и образование. 2021. №3(5). С. 34–41. URL: <https://cyberleninka.ru/article/n/avtomatizatsii-postroeniya-testovyh-zadaniy>
7. Самигулина Г.А. Методы генерации заданий с использованием NLP // Вестник КазНУ. 2022. №1(87). С. 56–63. URL: <https://cyberleninka.ru/article/n/obzor-suschestvuyuschih-metodov-avtomaticheskoy-generatsii-zadach>
8. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers // arXiv. 2018. URL: <https://arxiv.org/abs/1810.04805>
9. Яковлев Д.С. Генерация заданий с использованием ChatGPT // Современные технологии в образовании. 2023. №2(8). С. 67–74. URL: <https://openedu.rea.ru/jour/article/view/939>
10. Борисов А.В. Генеративные модели в образовании // Вестник науки и образования. 2023. №3(9). С. 34–41. URL:

<https://cyberleninka.ru/article/n/porozhdayuschie-modeli-v-mashinnom-obuchenii>

11. Коваленко Н.В. Автоматизация оценки знаний // Образование и наука. 2021. №4(6). С. 28–35. URL: <https://cyberleninka.ru/article/n/problemny-avtomatizatsii-testirovaniya>

12. Яковлев Д.С. Адаптивное тестирование в обучении // Современные технологии. 2023. №2(8). URL: <https://cyberleninka.ru/article/n/tehnologii-deeptech-v-sisteme-adaptivnogo-upravleniya-proizvodstvom>

13. Коваленко Н.В. Барьеры внедрения ИИ в образование // Образование и наука. 2021. №4(6). URL: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-obrazovanii-evolyutsiya-i-bariery>

14. Голиков А.В. Проблемы генерации задач // Наука и образование. 2021. №3(5). URL: <https://cyberleninka.ru/article/n/avtomaticheskaya-generatsiya-zadach>

15. Борисов А.В. Перспективы ИИ в обучении // Вестник науки и образования. 2023. №4(10). URL: <https://cyberleninka.ru/article/n/rol-iskusstvennogo-intellekta-i-tsifrovyyh-tehnologiy-v-vysshem-obrazovanii>

16. Коваленко Н.В. Будущее ИИ в образовании // Образование и наука. 2021. №1(3). URL: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-i-budushee-obrazovaniya>

17. Самигулина Г.А. Интеграция ИИ в образовательные системы // Вестник КазНУ. 2022. №2(88). URL: <https://cyberleninka.ru/article/n/integratsiya-iskusstvennogo-intellekta-v-obrazovanie>

18. Голиков А.В. Архитектура систем тестирования // Наука и образование. 2021. №2(4). URL: <https://cyberleninka.ru/article/n/arhitektura-intellektualnoy-sistemy-testirovaniya-1>

19. Иванов А.В. Мобильные образовательные технологии // Современные технологии. 2023. №3(9). URL: <https://cyberleninka.ru/article/n/mobilnoe-obrazovanie>

20. Петров Д.С. Текстовые нейросети в обучении // Информатика и образование. 2023. №2(8). URL: <https://info.infojournal.ru/jour/article/view/287>