

Несмотря на преимущества, широкое внедрение 3D-печати сдерживают высокая стоимость оборудования, длительность изготовления крупных деталей и необходимость постобработки. Крупные элементы печатаются долго, а качество изделий зависит от параметров печати и состояния материалов. Поэтому оптимальным становится комбинированный подход: применение 3D-печати для сложных и уникальных деталей, а традиционных методов для массового производства.

Дальнейшее развитие связано с повышением скорости печати, увеличением рабочего объема принтеров и расширением ассортимента материалов. Все это позволит применять 3D-печать не только для опытных образцов, но и в серийном производстве узлов, работающих под нагрузкой, делая аграрную технику более устойчивой и технологичной [3].

Результаты исследования показали, что 3D-печать становится важным инструментом для сельхозмашиностроения. Она ускоряет проектирование, улучшает конструктивные решения и повышает ремонтпригодность техники. По мере удешевления оборудования и стандартизации процессов аддитивные технологии займут прочное место в производстве сельскохозяйственных машин.

Литература

1. Сенин, П. В. Аддитивные технологии для производства и ремонта сельскохозяйственной техники / П. В. Сенин, М. Н. Чаткин, Е. А. Кильмяшкин. – Саранск : МГУ, 2024. – С. 586–591.
2. Шкуро, А. Е. Технологии и материалы 3D-печати: учеб. пособие / А. Е. Шкуро. – Екатеринбург : УГЛТУ, 2017. – С. 97–98.
3. Коряков, А. Е. Применение 3D-печати в машиностроении: эффективность, история и тенденции развития / А. Е. Коряков. – Минск : БНТУ, 2022. – С. 100–102.

АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ СОЗДАНИЯ УЧЕБНЫХ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

А. А. Глушенок, О. Д. Асенчик

*Гомельский государственный технический университет
имени П. О. Сухого, Республика Беларусь*

Представлена архитектура программного комплекса для автоматизации формирования учебных программ с использованием больших языковых моделей. Определены базовые модули, требования к ним и система их взаимодействия.

Ключевые слова: большие языковые модели (LLM), генерация с дополнительным поиском (RAG), автоматизация формирования учебных программ.

ARCHITECTURE OF A SOFTWARE SYSTEM FOR CREATING CURRICULUMS USING LARGE-SCALE LANGUAGE MODELS

A. A. Glushenok, O. D. Asenchik

Sukhoi State Technical University of Gomel, Republic of Belarus

This paper presents the architecture of a software system for automating the generation of curricula using large-scale language models. The basic modules, their requirements, and the system of their interaction are defined.

Keywords: large-scale language models (LLM), search-augmented generation (RAG), automation of curriculum generation.

Современная система образования находится в условиях постоянных изменений: быстро меняются требования рынка труда, обновляются образовательные стандарты, растет потребность в индивидуализации обучения. Развитие технологий искусственного интеллекта, в частности больших языковых моделей (*LLM*), открывает новые возможности для автоматизации и интеллектуальной поддержки в проектировании образовательных программ. Использование данной языковой модели позволяет генерировать структуру курсов, адаптировать содержание под уровень студентов, интегрировать актуальные источники знаний и обеспечивать гибкую настройку под разные образовательные траектории. В работе рассматривается разработка системы автоматизированного формирования учебных программ с использованием больших языковых моделей. В качестве основы разработана архитектура приложения с поддержкой генерации с дополнительным поиском (*RAG*) и многоагентной обработки, обеспечивающая модульность, масштабируемость и интеграцию внешних источников знаний. Система предполагает использование эмбедингов, векторной базы данных, специализированных агентов и агрегатора для консолидации ответов. Таким образом, работа сочетает исследование образовательных приложений и проектирование современного архитектурного решения на базе *LLM*, что позволяет создавать учебные программы, соответствующие актуальным стандартам и индивидуальным запросам обучающихся.

В данной работе описана архитектура приложения для создания учебных программ с использованием больших языковых моделей. Для разработки приложения был выбран динамически типизированный, язык программирования *Python*. В ходе сбора и анализа данных о требованиях к подобным приложениям был выделен следующий необходимый функционал: формирование учебных программ заданной структуры и в заданном нормативном документе формате; проверка уже созданных программ на предмет соответствия образцу и нормативным документам министерства образования Республики Беларусь, в случае отклонений должны выдаваться замечания, и формироваться правильные программы; генерация текстов как на основе ответов одной или нескольких *LLM*, так и с использованием генерации, дополненной поиском (*RAG*) для работы со специально подобранными эталонными источниками (учебниками, учебными пособиями, документацией); верификация, хранение, просмотр и редактирование дисциплин, программ, планов, промптов, настроек *LLM*, эталонных источников, генерация текстов программ.

Архитектура системы построена по многоуровневому принципу и включает несколько ключевых компонентов. На первом уровне находится клиентский слой, представляющий собой интерфейс для пользователя. Этот слой отвечает за прием запросов от пользователя и отображение полученных результатов.

Центральным элементом системы является оркестратор. Он управляет обработкой запросов, решает, когда достаточно простого ответа от *LLM*, а когда требуется подключение *RAG* или использование команды агентов. Оркестратор выбирает, какие агенты задействовать для обработки конкретного запроса, и контролирует последовательность их работы.

Система включает несколько типов агентов, каждый из которых выполняет специализированные задачи. *RAG*-агент обращается к векторной базе данных, обеспечивающей хранение и поиск эмбедингов. В данной системе применяется векторная база данных *Qdrant*, поскольку она обеспечивает высокую скорость и качество семантического поиска. Данная база данных обладает высокой производительностью и разворачивается локально, имеет открытую лицензию. *Reasoning*-агент напрямую использует *LLM* для логических и аналитических рассуждений. *Tool*-агент

взаимодействует с внешними сервисами и базами данных, выполняя необходимые операции и возвращая результаты.

LLM выполняет основную роль генеративного и аналитического ядра системы: формирует ответы, обобщает найденную информацию, проводит рассуждения и структурирует результаты. В данной системе выбрана модель *Qwen* согласно [1]. Данная модель была выбрана так как она демонстрирует высокое качество обработки русскоязычного текста, так же она обладает открытой лицензией, имеет контекстное окно до 256000 токенов, поддержку *reasoning*-операций и *RAG*-сценариев.

Для обеспечения поиска и отбора релевантной информации применяется эмбединговая модель *intfloat/multilingual-e5-base*, которая преобразует текстовые данные в числовые векторные представления (эмбединги), отражающие их семантическое сходство. Выбор модели осуществлялся по нескольким критериям: поддержка русского языка, высокое качество семантических представлений, обеспечивающее точный поиск по смыслу, а не только по ключевым словам, открытая лицензия и возможность локального использования без зависимости от внешних *API*. Эти векторы сохраняются во векторной базе данных и используются для поиска фрагментов, наиболее близких по смыслу к пользовательскому запросу. Результаты поиска передаются модели *Qwen*, что позволяет системе формировать более точные и обоснованные ответы при генерации учебных материалов согласно [2].

Агрегатор собирает частичные ответы, полученные от всех агентов, объединяет их в единый итоговый результат, добавляет информацию об источниках и передает ответ пользователю. Такая архитектура обеспечивает модульность, масштабируемость и возможность интеграции различных источников информации при формировании ответа.

Разработка и внедрение предложенной системы открывает перспективы для более гибкого и персонализированного образования, автоматизации рутинных процессов проектирования курсов, а также интеграции современных технологий искусственного интеллекта в образовательную практику.

Литература

1. Yuxia, W. Factuality of Large Language Models: A Survey / W. Yuxia // ArXiv preprint arXiv: 2402.02420v3. – 2024. – 11 с.
2. Renata, N. Exploring the use of retrieval-augmented generation models in higher education: A pilot study on artificial intelligence-based tutoring / N. Renata // Social Sciences & Humanities Open. – 2025. – Т. 12, N 1. – С. 1–10. – Art. 101751.

ПРОБЛЕМА КОНТРОЛИРУЕМОСТИ И ВЕРИФИКАЦИИ В АВТОНОМНЫХ LLM-АГЕНТАХ

Ю. Д. Евженко, К. С. Курочка

*Гомельский государственный технический университет
имени П. О. Сухого, Республика Беларусь*

Происходит фундаментальный сдвиг от использования больших языковых моделей как пассивных инструментов генерации контента к их применению в качестве ядра рассуждений для автономных AI-агентов, способных к планированию и выполнению многошаговых задач. Однако их стохастическая природа вступает в противоречие с требованиями к безопасности и надежности систем управления. Целью работы является систематизация рисков, присущих LLM-агентам, при помощи использования архитектурных паттернов контроля, известных как «Guardrails». Методика исследования включает анализ и классификацию сбоев на этапах планирования и выполнения действий, а также систематизацию многоуровневых подходов к их предотвращению. В качестве результата предложена трехуровневая модель «Guardrails», обеспечивающая контроль над