

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, растёт 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 нормативным документом 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 эмбеддингов. 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 эмбеддинговая модель *intfloat/multilingual-e5-base*, которая преобразует текстовые данные в числовые векторные представления (эмбеддинги), 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 бо-

лее 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 рассуждениями, действиями и последствиями, что закладывает основу для создания безопасных и предсказуемых AI-агентов.

Ключевые слова: LLM-агент, контролируемость, верификация, безопасность AI, Guardrails, архитектурный паттерн, автономные системы.

THE PROBLEM OF CONTROLLABILITY AND VERIFICATION IN AUTONOMOUS LLM AGENTS

Yu. D. Youzhanka, K. S. Kurochka

Sukhoi State Technical University of Gomel, Republic of Belarus

A fundamental shift is underway from using large language models (LLMs) as passive content generation tools to employing them as a reasoning core for autonomous AI agents capable of planning and executing multi-step tasks. However, their stochastic nature directly conflicts with the safety and reliability requirements of control systems. The aim of this work is to systematize the risks inherent in LLM agents by architectural control patterns known as "Guardrails". The research methodology includes the analysis and classification of failures at the planning and action execution stages, as well as the systematization of multi-level approaches to their prevention. The result is a proposed three-level "Guardrails" model that provides control over reasoning, actions, and consequences, laying the foundation for creating safe and predictable AI agents.

Keywords: LLM agent, controllability, verification, AI safety, Guardrails, architectural pattern, autonomous systems.

Большие языковые модели (LLM) эволюционируют от пассивных генераторов текста к центральным компонентам исполнительных систем или AI-агентов [1].