

УДК 378.147:004.8
DOI 10.17513/snt.40794



АЛГОРИТМ «ЦЕПОЧКА ПРОМПТОВ» В ЗАДАЧЕ ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНЫХ ОБРАЗОВАТЕЛЬНЫХ ТРАЕКТОРИЙ БУДУЩИХ ИТ-СПЕЦИАЛИСТОВ

**Капустин Д. А. ORCID ID 0009-0008-4254-319X,
Суворова Е. Ю. ORCID ID 0000-0003-4308-9765,
Зинченко В. О. ORCID ID 0000-0002-4469-525X,
Швыров В. В. ORCID ID 0009-0005-0163-2952**

*Федеральное государственное бюджетное образовательное учреждение высшего образования
«Луганский государственный педагогический университет», Луганск,
Российская Федерация, e-mail: kap-kapchik@mail.ru*

В контексте современных вызовов ИТ-образования, связанных с необходимостью динамической персонализации обучения будущих ИТ-специалистов при сохранении качества подготовки, представлена разработка алгоритма «Цепочка промптов» для решения задачи формирования индивидуальных образовательных траекторий. Цель работы – создание алгоритма, обеспечивающего системную декомпозицию задачи на 10 управляемых этапов, от уточнения требований до финальной интеграции компонентов траектории с педагогически обоснованной последовательностью. Ключевым инновационным решением является использование адаптивной JSON-структуры для передачи контекста между этапами, что позволяет динамически включать или исключать отдельные шаги в зависимости от типа образовательной задачи, сохраняя при этом педагогическую целесообразность. Алгоритм реализован на Python с интеграцией произвольных больших языковых моделей и поддерживает механизмы динамической адаптации сложности заданий, междисциплинарной интеграции знаний через кластеризацию компонентов, а также актуализации содержания на основе генерации с дополненной выборкой. В статье представлены результаты первичной апробации при формировании ИОТ по дисциплине «Web-разработка на Django», которые продемонстрировали работоспособность алгоритма: обеспечено автоматическое выявление пробелов в знаниях, генерация персонализированных заданий и адаптация сложности. Научная новизна заключается в гибридной структурированно-генеративной архитектуре, сочетающей педагогический дизайн и возможности языковых моделей, что отличает предложенный подход как от хаотичного промптинга, так и от онтологических систем с жесткими правилами.

Ключевые слова: «Цепочка промптов», большие языковые модели, индивидуальные образовательные траектории, ИТ-образование

THE «PROMPT CHAIN» ALGORITHM FOR THE FORMATION OF INDIVIDUAL LEARNING TRAJECTORIES FOR FUTURE IT SPECIALISTS

**Kapustin D. A. ORCID ID 0009-0008-4254-319X,
Suvorova E. Yu. ORCID ID 0000-0003-4308-9765,
Zinchenko V. O. ORCID ID 0000-0002-4469-525X,
Shvyrov V. V. ORCID ID 0009-0005-0163-2952**

*Federal State Budgetary Educational Institution of Higher Education
«Lugansk State Pedagogical University», Lugansk, Russian Federation,
e-mail: kap-kapchik@mail.ru*

In the context of modern challenges in IT education, associated with the need for dynamic personalization of training for future IT specialists while maintaining the quality of education, the development of the “Prompt Chain” algorithm for solving the problem of forming individual learning paths is presented. The aim of the work is to create an algorithm that ensures systematic decomposition of the task into 10 manageable stages, from clarifying requirements to the final integration of trajectory components. The key innovative solution is the use of an adaptive JSON structure for transferring context between stages, which allows dynamically including or excluding individual steps depending on the type of educational task, while maintaining the pedagogical appropriateness of the sequence. The algorithm is implemented in Python with the integration of arbitrary large language models and supports mechanisms for dynamic adaptation of task complexity, interdisciplinary integration of knowledge through component clustering, as well as content updating based on the retrieval augmented generation approach. The results of testing the algorithm in forming individual educational trajectories for the discipline “Web Development with Django” demonstrated its operability: automatic identification of knowledge gaps, generation of personalized tasks, and difficulty adaptation were achieved. The scientific novelty lies in the hybrid structured-generative architecture that combines pedagogical design and the capabilities of language models, distinguishing the proposed approach from both chaotic prompting and ontology-based systems with rigid rules.

Keywords: «Prompt Chain» algorithm, adaptive learning, large language models, hybrid architectures, individual learning trajectories, IT education

Введение

Концепция цепочки промптов (chain-of-thought), предложенная в работе [1], стала одним из ключевых методов повышения точности и надежности больших языковых моделей (БЯМ, англ. LLM – Large Language Models) при решении сложных задач. Суть в том, что модель не выдает ответ сразу, а сначала выстраивает цепочку промежуточных рассуждений. По сути, сложный вопрос разбивается на несколько понятных шагов, что дает возможность отследить логику и найти ошибку на любом этапе. Авторы показали, что такой подход заметно повышает качество решения арифметических и символьных задач, а также помогает лучше справляться с вопросами, где нужны дополнительные знания из контекста. Именно такой многошаговый метод необходим там, где важно не просто получить правильный ответ, но и понять, как он был получен. К таким задачам относятся, например, построение индивидуальной образовательной траектории (ИОТ).

В сфере IT-образования проблема персонализации стоит особенно остро. Технологии развиваются настолько стремительно, что содержание дисциплин быстро устареет. Кроме того, студенты приходят с очень разным уровнем подготовки, и преподавателям приходится оперативно адаптировать материал под группу или даже под каждого обучающегося. Традиционные подходы к формированию ИОТ недостаточно гибки и не успевают за изменениями. Большие языковые модели, в свою очередь, открывают новые возможности. С их помощью можно создавать адаптивные образовательные системы, способные генерировать образовательный контент в режиме реального времени. Однако и тут есть существенные ограничения. Большинство существующих решений либо выдают разрозненные материалы в ответ на однократный запрос, либо используют универсальные цепочки промптов, не учитывающие педагогическую специфику. Возникает потребность в таком способе взаимодействия с БЯМ, который сочетал бы структурированность педагогического дизайна и гибкость генеративных моделей.

Целью настоящей работы является разработка и апробация алгоритма «Цепочка промптов», обеспечивающего системную декомпозицию задачи формирования ИОТ на управляемые этапы с передачей контекста через адаптивную JSON-структуру. Новизна подхода заключается в том, что предложенная последовательность промптов не просто дробит задачу, а реализует пе-

дагогически обоснованный сценарий, где каждый этап решает конкретную образовательную подзадачу – от диагностики пробелов до генерации междисциплинарных заданий, а сквозная JSON-модель сохраняет преемственность решений.

В отличие от общих цепочек промптов и онтологических систем с жесткими правилами, предложенная архитектура сочетает фиксированную педагогическую логику с динамической адаптацией через JSON-контекст и внешние данные, что позволяет перестраивать последовательность шагов в зависимости от типа задачи (например, коррекция существующей траектории).

Понятие ИОТ прошло в отечественной педагогике путь от теоретического осмысления к практической реализации. Ранние работы Е. П. Носовой, В. К. Абримова, В. В. Лебидько [2; 3] заложили основы понимания ИОТ как способа построения обучения, учитывающего личностные особенности, интересы и темп продвижения студента. Дальнейшие исследования С. А. Вдовиной, И. М. Кунгуровой, Н. Ю. Шапошниковой [4; 5] систематизировали понятийный аппарат, разграничив смежные термины (индивидуальный образовательный маршрут, индивидуальный учебный план) и предложив критерии выделения структурных компонентов траектории. На современном этапе акцент сместился на интеграцию ИОТ в цифровую среду и связь с профессиональными компетенциями. Так, С. Е. Старостина [6] рассматривает ИОТ не просто в виде формы организации обучения, а как средство развития самостоятельности. По ее мнению, выстраивая собственный путь, студент учится ставить цели, планировать и оценивать свои действия. В свою очередь, Л. В. Шилова и Л. Ю. Фетисова [7] связывают персонализацию с запросами профессии, подчеркивая, что учет индивидуальных интересов должен сочетаться с требованиями конкретной сферы деятельности. Современные работы [8; 9] обсуждают уже накопленный практический опыт внедрения индивидуальных траекторий в вузе и в инженерном образовании. Однако при всей развитости теоретической базы традиционные подходы на деле остаются преимущественно статичными, фиксируя набор элективных курсов, и не успевают подстраиваться ни под текущий уровень студента, ни под быстро меняющиеся условия рынка. Для IT-области, где технологии обновляются каждый год, это особенно серьезное упущение.

Параллельно развивались технологии искусственного интеллекта. Исследования последних лет демонстрируют широкий

потенциал БЯМ для решения образовательных задач: автоматическое оценивание, генерация контента, создание интеллектуальных тьюторских систем. Систематизированный обзор 50 научных публикаций, проведенный Ч. К. Ло [10], показывает, что в образовательном контексте наиболее эффективны запросы, содержащие несколько примеров и побуждающие модель к пошаговым рассуждениям. Однако использование БЯМ в «одиночном» режиме часто приводит к поверхностным результатам, особенно при многошаговых рассуждениях и учете множества факторов. Так, в работе [11] авторы столкнулись с необходимостью не просто классифицировать ответ, но и выделять в нем желательные и нежелательные компоненты, что потребовало более тонкого подхода, чем стандартный промптинг. А в исследовании [12] ограниченность предметных знаний БЯМ предложено компенсировать поиском релевантной информации с использованием RAG. Для преодоления ограничений многошаговых задач была предложена техника цепочек промптов [1], повышающая точность и прозрачность генерации, что подтверждено в различных предметных областях: от классификации юридических документов [13] до семантического моделирования [14].

В образовательном контексте цепочки промптов начали применяться для создания пошаговых примеров решения задач [15], анализа стратегий саморегуляции обучения [16] и выявления уязвимостей в коде [17]. Однако эти применения носят фрагментарный характер – они решают отдельные дидактические задачи, но не охватывают формирование целостной образовательной траектории, связывающей диагностику, проектирование, генерацию контента и оценку в единый педагогически обоснованный цикл.

Еще одно важное направление связано с повышением достоверности контента через интеграцию с внешними базами знаний. Так называемый подход RAG снижает риск «галлюцинаций» и устаревания информации [18]. Методы кластеризации учебных компонентов позволяют выявлять междисциплинарные связи для построения комплексных заданий [19]. Однако эти инструменты используются либо самостоятельно, либо в составе узкоспециализированных систем, не интегрированных в процесс проектирования ИОТ.

Анализ работ, посвященных применению LLM в образовании, выявляет устойчивые ограничения. Во-первых, это недостаточная системность, поскольку большинство решений генерируют отдельные

материалы, не встраивая их в целостную траекторию, согласованную по целям и пререквизитам. Во-вторых, отсутствуют механизмы учета междисциплинарных связей, что приводит к формированию изолированных компетенций, непригодных для комплексных профессиональных задач. Е. В. Комарова в работе [20] отмечает серьезное противоречие в современных подходах к использованию языковых моделей в образовании. Суть проблемы в том, что даже самые продвинутые генеративные возможности модели сами по себе не обеспечивают педагогической целостности учебного материала. Отдельные фрагменты контента могут быть корректными с точки зрения фактов, но при этом не складываются в целостную, логически выстроенную систему знаний у обучающегося. Кроме того, существует большая проблема контроля качества сгенерированного материала. Без четких критериев и инструментов оценки внедрить подобные решения в учебный процесс крайне затруднительно.

Таким образом, даже имея в распоряжении такие инструменты, как цепочки промптов, RAG, методы кластеризации, и опираясь на развитую теоретическую базу формирования индивидуальных образовательных траекторий, все еще существует разрыв между потенциалом больших языковых моделей и их практическим применением для создания по-настоящему персонализированного обучения. Для преодоления этого разрыва требуется разработка такой архитектуры, в которой педагогически продуманная структура, адаптивность модели и актуальные внешние данные работают не параллельно, а во взаимодействии. Авторы предлагают алгоритм «Цепочка промптов», который как раз реализует многоэтапный подход к формированию ИОТ. Его особенность в педагогически выверенной декомпозиции задачи, при этом передача контекста между этапами организована через адаптивную JSON-структуру. Такой подход, во-первых, делает процесс прозрачным, то есть можно отследить, как формируется траектория на каждом этапе, и во-вторых, допускает динамическую корректировку траектории без перестройки всей системы заново.

Материалы и методы исследования

В основе данного исследования лежит синтез пяти методологических подходов к проектированию индивидуальных образовательных траекторий будущих IT-специалистов, а именно системно-деятельностного, конвергентного, компетентностного, технологического и персонализированного.

ного. В качестве практической реализации этой методологии разработан алгоритм «Цепочка промптов», последовательно разбивающий задачу формирования траектории на десять логически связанных этапов. Его принципиальное отличие – адаптивная JSON-структура для передачи контекста между этапами. Она позволяет гибко варьировать последовательность шагов, например при корректировке существующей траектории можно пропустить начальные этапы анализа и сразу перейти к актуальным правкам, сохраняя при этом целостность процесса.

Как это работает на практике? JSON-структура выступает сквозным «носителем состояния» формируемой траектории. На первом этапе в нее записывается исходная задача, затем, на каждом следующем шаге, в структуру добавляется новый вложенный объект. В нем хранятся входные данные, которые представляют собой результаты предыдущих шагов и структурированный выход текущего этапа. Таким образом, система «помнит» не только первоначальный запрос, но и всю историю уточнений, выявленные ограничения, риски и принятые решения.

Алгоритм разбивает процесс формирования ИОТ на десять последовательных этапов, каждый из которых выстроен в соответствии с педагогической логикой. Первые два этапа связаны с анализом исходной ситуации, происходит уточнение целей и ограничений, а затем планирование общей структуры траектории. Далее следует информационно-аналитический сбор данных об успеваемости студента и внешних требованиях, выявление пробелов в знаниях и формулирование вариантов учебных модулей и заданий. На этапах оценки и детализации происходит обоснованный выбор компонентов с учетом междисциплинарных связей, формируется развернутая структура модулей с учетом выявленных взаимосвязей. Завершающие шаги включают генерацию учебных материалов, проверку баланса нагрузки и финальную интеграцию в единую траекторию. Любой этап можно пересмотреть или скорректировать без полного перезапуска процесса (рисунок).

Особенность алгоритма в том, что он подстраивается под уровень студента. На этапах 5-8 сложность заданий меняется в зависимости от его прогресса. Благодаря этому нагрузка остается оптимальной, индивидуальной для каждого. На шестом этапе задействуется метод кластеризации учебных компонентов, цель которого – найти связи между темами из разных дисциплин и на их основе создать комплексные междисципли-

нарные задания. А RAG-подход, используемый уже на этапе сбора информации, связывает генерацию материалов с действующими профессиональными стандартами. Таким образом обеспечивается соответствие программы современным требованиям отрасли.

Программная часть системы реализована на языке Python, а для разработки интерфейса использован фреймворк Flask. Интеграция с языковыми моделями DeepSeek-r1 и YandexGPT-5-Lite-8B-instruct осуществляется через библиотеку Ollama, обеспечивающую локальный сервер и загрузку моделей. Такой подход позволил отказаться от внешних API и минимизировать риски утечки данных.

Результаты исследования и их обсуждение

Для проверки работоспособности алгоритма была поставлена задача: сформировать ИОТ для студента с начальным уровнем Python (знание основ, отсутствие опыта работы с базами данных и фреймворками) с целью освоения Django за 16 недель.

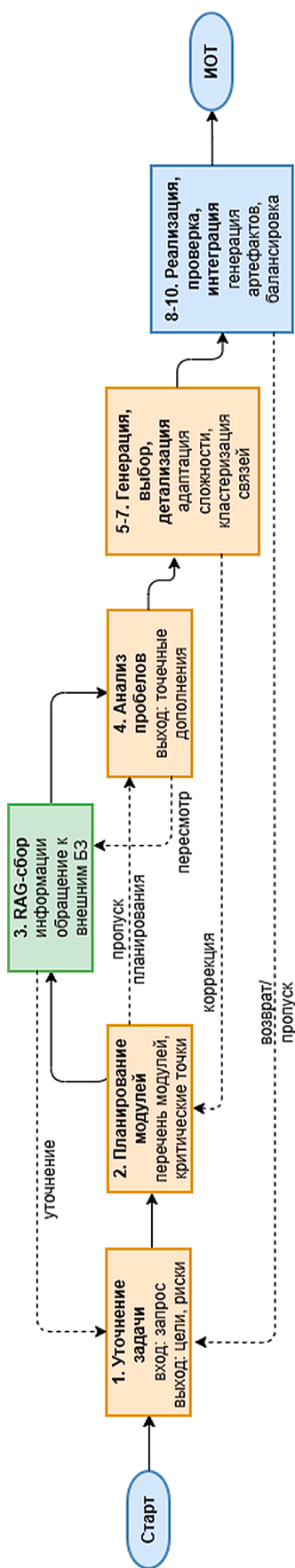
Входные данные. Дисциплина: «Web-разработка на Django». Текущий уровень знаний (оценивается студентом): «Понимаю основы Python (функции, классы), HTML/CSS на уровне верстки простых страниц. Никогда не работал с базами данных, фреймворками, Git». Цель: научиться делать и разворачивать простое блог-приложение. Ниже представлена поэтапная реализация алгоритма формирования ИОТ (табл. 1).

Декомпозиция здесь – это разбиение стандартной, линейной программы курса на адаптивную сеть взаимосвязанных учебных элементов (микромодулей, заданий, ресурсов) и построение для студента уникального графа прохождения по этой сети. Без алгоритма курс – это линейный список тем (лекция 1, лекция 2...), одинаковый для всех. Проблема студента на теме N приводит к «эффекту снежного кома» на теме N+1. С алгоритмом курс – это адаптивная карта знаний, где:

для Студента А путь от «Введения» к «Дипломному проекту» короткий и прямой (он уже много знает);

для Студента Б (из примера) на пути появляются обязательные точки ветвления и дополнительные петли (коррекционные модули, дополнительные упражнения и пр.);

для Студента В (который пришел, например, уже зная Django, но без понимания CSS) алгоритм построит путь, где он быстро проходит базовые модули фреймворка, но углубляется в модуль CSS для фронтенда своих проектов.



Блок-схема алгоритма «Цепочка промптов»
Примечание: составлено авторами

Таблица 1

Поэтапная реализация алгоритма формирования ИОТ

Этап	Действия алгоритма	Педагогический смысл
1. Уточнение задачи	Структурирование запроса, выделение целей и рисков	Сформулирована задача: «ИОТ по Django для создания блога за 16 недель». Выявлены риски: непонимание MVC/MVT, отсутствие опыта с БД и Git
2. Планирование	Декомпозиция на модули с определением критических точек	1. Модуль 0 (адапционный): Python для Web (декораторы, ООП, Git). 2. Модуль 1: ядро Django (MVT, шаблоны, приложение без БД). 3. Модуль 2: работа с данными (Models, ORM). 4. Модуль 3: формы, аутентификация. 5. Модуль 4: деплой. Критическая точка – переход к ORM
3. Информационный сбор	RAG-обращение к учебным материалам, документации Django, базе заданий	Получены актуальные источники для генерации контента: рабочая программа, лекции, личные ошибки
4. Анализ	Выявление пробелов в знаниях на основе входных данных	Обнаружено слабое понимание наследования классов. В Модуль 0 добавлен точечный контент по ООП с примерами, приближенными к моделям Django (класс Article с методами save, publish)
5-7. Генерация, выбор, детализация	Предложение вариантов учебных модулей с адаптацией под уровень усвоения	Для темы «ORM Django» сгенерированы три сценария: 1. Отличное усвоение – продвинутая практика с отношениями ManyToMany. 2. Среднее – пошаговый тренажер с аналогиями. 3. Проблемы – видеообъяснение и имитационное задание. Создан сквозной микропроект, связывающий шаблоны (модуль 1) и формы (модуль 3)
8-10. Реализация, проверка, интеграция	Генерация персонализированных артефактов и оценка сбалансированности	Сформированы ссылки на документацию, задание по деплою блога с инструкцией под ОС студента, чек-листы самопроверки. Проверено отсутствие перегруженности теорией и последовательная разработка проекта

Примечание: составлено авторами на примере дисциплины «Web-разработка на Django».

Таблица 2

Сравнительный анализ подходов к формированию индивидуальных образовательных траекторий

Критерий/подход	Традиционные статические планы	Адаптивные тыторские системы	LLM-генерация по единому промпту	Общие цепочки промптов	Предлагаемый алгоритм «Цепочка промптов»
Уровень системности	Высокий, но статический	Средний. Фокус на адаптации внутри модуля/темы, реже – на всей траектории	Низкий. Генерация разрозненных материалов, упреждений без целостной логики курса	Средний/высокий. Решает сложные задачи, но без гарантии педагогической системности	Высокий. Целенаправленная 10-этапная композиция, гарантирующая построение целостной траектории от анализа до финальной интеграции
Механизм адаптации	Отсутствует или основан на ручном выборе элементов	Жесткие правила, предопределенные пути на основе онтологии предметной области	Единовременная адаптация на основе описания студента в промпте. Качество непредсказуемо	Пошаговая адаптация через контекст цепочки. Не специализирована под педагогические задачи	Динамическая, многоуровневая адаптация. Адаптация на этапах анализа, генерации и выбора на основе структурированного контекста (JSON) и внешних данных (RAG)
Учет междисциплинарных связей	Фиксируется вручную при разработке программы	Возможен через связи в онтологии, но требует сложной ручной настройки	Случайный, зависит от знаний модели. Несистематический	Может быть реализован через специальные промпты, но не является архитектурной особенностью	Архитектурно заложенный механизм. Этап 6 включает автоматическую кластеризацию смежных дисциплин для формирования комплексных кейсов
Работа с актуальностью контента	Трудоемкий периодический пересмотр программ	Сложность и стоимость обновления онтологий и правил	Зависит от актуальности данных LLM на момент обучения. Непрозрачно	Может использоваться RAG, но не является стандартизированным компонентом	Интегрированный RAG-механизм. На этапе 3 систематически запрашиваются актуальные данные из внешних БЗ
Контролируемость и прозрачность процесса	Высокая (все этапы ручные), неадаптивна.	«Черный ящик» на основе правил; логика может быть сложна для анализа	Нулевая. Единый промпт → единственный ответ, процесс генерации невидим	Частичная. Видны промежуточные шаги, но их педагогическая целесообразность не гарантирована	Максимальная. Модульная архитектура с четкими входами/выходами каждого этапа и сквозной передачей контекста через JSON позволяет отлаживать каждый шаг
Требования к вычислительным ресурсам	Низкие	Средние (запуск системы логического вывода)	Высокие (вызов крупной LLM, но разовый)	Очень высокие (многочисленные вызовы LLM)	Высокие, но оптимизированные. 10 последовательных вызовов LLM компенсируются локальным развертыванием и четкой структурой, снижающей количество «холостых» итераций
Роль преподавателя (человека в цикле)	Разработчик и контролер всей траектории	Разработчик правил и интерпретатор действий системы	Редактор сырого стенированного материала	Настройщик цепочки и интерпретатор результатов	Стратег. Система берет на себя рутинную композицию, оставляя эксперту финальное одобрение и стратегические правки
Гибкость к изменению входных условий	Крайне низкая. Требуется полного пересмотра	Низкая. Изменение требует переписывания правил/онтологии	Средняя. Можно изменить промпт, но результат непредсказуем	Высокая. Можно перестроить цепочку, но нужны экспертные знания	Динамическая гибкость. Адаптивная JSON-структура позволяет пропускать или включать этапы в зависимости от типа задачи (коррекция существующей ИОТ)

Примечание: составлено авторами на основе анализа литературы и экспериментального сравнения прототипа с открытыми аналогами.

В рамках одной дисциплины такой алгоритм критически необходим, потому что он решает главную проблему массового обучения: пропасть между сильными и слабыми студентами внутри одной группы. Преподаватель физически не в состоянии уделить индивидуальное внимание каждому из 30 студентов, моментально подбирать альтернативные объяснения и задания для каждого, кто не понял тему. Алгоритм берет эту нагрузку на себя. Он превращает статичный учебный ресурс в интерактивного, адаптивного консультанта, который удерживает логику дисциплины, обеспечивая не просто набор материалов, а продуманную, уникальную для каждого обучающегося последовательность шагов. Это и есть высшая форма персонализации в образовании, которая работает не на уровне выбора курса, а на уровне индивидуальной учебной траектории студента внутри него.

Несмотря на положительные результаты тестирования, исследование имеет ряд ограничений, которые необходимо учитывать при интерпретации результатов. На данный момент апробация проведена на примере одной дисциплины с одним типовым профилем студента. Отсутствуют количественные сравнения с традиционными и другими адаптивными системами, а также статистически значимые данные о влиянии алгоритма на успеваемость, мотивацию или вовлеченность студентов. Высокая вычислительная стоимость алгоритма требует локального развертывания мощных БЯМ (не менее 8-16 ГБ VRAM), что влечет за собой дополнительные затраты на инфраструктуру. Помимо этого, параметры системы приходится подбирать вручную под каждую программу, поскольку универсального шаблона не существует, а это также увеличивает порог внедрения. Кроме того, базы знаний, задействованные в RAG-компоненте, быстро теряют актуальность и требуют регулярного обновления, а механизм автоматического обновления в текущей версии не реализован. Наконец, сгенерированные материалы обязательно должны проходить финальную проверку преподавателями. Это снижает автономность системы, но является осознанным проектным решением для обеспечения педагогической ответственности. Данные ограничения не отменяют ценности представленного прототипа как доказательства концепции, но требуют обширных эмпирических данных. Тем не менее для объективной оценки места разработанного алгоритма «Цепочка промптов» среди существующих решений необходимо сопоставить его не только с идеальным случаем, но и с реально работающими аналогами. Поэтому

ниже проведем систематическое сравнение с основными классами существующих подходов к автоматизации и персонализации обучения. Сравнение выполнено по ключевым критериям, определяющим эффективность и практическую применимость в динамичной сфере IT-образования (табл. 2).

Как видно, предложенный алгоритм занимает уникальную нишу между полностью ручным трудоемким педагогическим дизайном, когда преподаватель детально прописывает каждый шаг, и генеративными моделями, работающими по принципу «черного ящика», когда БЯМ выдает результат, но логика его получения остается скрытой. Алгоритм не просто использует набор технологий (LLM, RAG и цепочки промптов), а организует их в педагогически осмысленную последовательность шагов. Передача данных между ними осуществляется через специализированную JSON-структуру, которая фиксирует состояние траектории на каждом этапе. В этом смысле алгоритм сохраняет структурированность, характерную для онтологических систем, где все строго формализовано, но при этом не требует жесткого прописывания правил. Вместо этого он использует контекстно зависимую генерацию, позволяя работать с нечеткими входными данными. Кроме того, алгоритм не полагается исключительно на внутренние знания LLM, которая может устареть или «галлюцинировать». Применяемые механизмы RAG и кластеризации обеспечивают постоянную привязку генерации образовательного контента к актуальным внешним источникам: профессиональным стандартам, учебным материалам, требованиям рынка труда и пр. Такой подход позволяет выйти за пределы фиксированного набора тем и дает возможность превратить статичный курс в динамическую сеть микромодулей, которая реагирует на индивидуальные затруднения студента, а не предлагает всем одну и ту же линейную траекторию.

Заключение

В ходе исследования разработан и апробирован алгоритм «Цепочка промптов», который последовательно декомпозирует процесс формирования индивидуальной образовательной траектории будущих IT-специалистов на 10 этапов с передачей контекста между ними через JSON-структуру. Проверка на реальной дисциплине, в частности на курсе «Web-разработка на Django», продемонстрировала работоспособность такого способа организации взаимодействия с LLM. Он позволяет гибко адаптировать обучение под уровень и запросы

студентов. Среди достоинств подхода: целостность траектории, заложенная в саму логику разбиения задачи; динамическое изменение сложности заданий в зависимости от уровня усвоения материала студентом; автоматическая генерация вариантов заданий и междисциплинарных проектов. Кроме того, модульная архитектура и сквозная JSON-модель делают процесс прозрачным: на любом этапе видно, какие решения принимались и на каком основании. Алгоритм не требует от преподавателя значительных дополнительных усилий и легко адаптируется под разные образовательные программы. Его применение в профессиональной подготовке будущих IT-специалистов позволяет выстроить персонализированное обучение, не ослабляя при этом педагогического контроля.

Дальнейшие исследования будут направлены на разработку системы метрик для количественной оценки эффективности предложенного подхода, расширение выборки для проведения полноценного педагогического эксперимента, создание графического интерфейса для упрощения настройки под новые дисциплины и автоматизацию обновления баз знаний.

Список литературы

- Kasneci E., Sebler K., Küchemann S., Bannert M., et al. ChatGPT for good? On opportunities and challenges of large language models for education // *Learning and Individual Differences*. 2023. Vol. 103. № 102274. P. 1–13. DOI: 10.1016/j.lindif.2023.102274.
- Носова Е. П. Индивидуальная образовательная траектория: сущность и механизмы проявления // *Известия Российского государственного педагогического университета им. А. И. Герцена*. 2009. № 12 (91). С. 138–144. EDN: JZVPDV.
- Абросимов В. К., Лебидько В. В. Индивидуальные образовательные траектории: формализация выбора, построения, управления // *Открытое образование*. 2009. № 4. С. 4–16. EDN: KUSQZZ.
- Вдовина С. А., Кунгурова И. М. Сущность и направления реализации индивидуальной образовательной траектории // *Науковедение*. 2013. Вып. 6. С. 1–8. EDN: SAKSCB.
- Шапошникова Н. Ю. Индивидуальная образовательная траектория студента: анализ трактовок понятия // *Педагогическое образование в России*. 2015. № 5. С. 39–44. EDN: TWLKHf.
- Старостина С. Е. Внедрение индивидуальных образовательных траекторий в вузе: проблемы и перспективы // *Ученые записки Забайкальского государственного университета*. 2023. Т. 18. № 3. С. 41–47. DOI: 10.21209/2658-7114-2023-18-3-41-47. EDN: OGSRON.
- Шилова Л. В., Фетисова Л. Ю. Индивидуальная образовательная траектория студента как инновационная модель профессионального и личностного развития // *Высшее образование сегодня*. 2020. Вып. 7. С. 40–43. DOI: 10.25586/RNU.NET.20.07.P.40. EDN: TMQTLF.
- Галушина П. С., Ражина Е. В., Неверова О. П., Горбунова О. С., Стахеева Л. М. Индивидуальная образовательная траектория студента вуза // *Образование и право*. 2024. № 5. С. 504–508. DOI: 10.24412/2076-1503-2024-5-504-508. EDN: RNZTNY.
- Донкова И. А., Якубовский Ю. Е., Карякин И. Ю., Карякин Ю. Е. Опыт формирования индивидуальных образовательных траекторий // *Инженерное образование*. 2024. № 35. С. 119–130. DOI: 10.54835/18102883_2024_35_11. EDN: JZDPQS.
- Lo C. K. What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature // *Education Sciences*. 2023. Vol. 13. № 4. Article 410. P. 1–15. DOI: 10.3390/educsci13040410.
- Uto M., Uchida Y. Automated Short-Answer Grading Using Deep Neural Networks and Item Response Theory // Bitten-court I., Cukurova M., Muldner K., Luckin R., Millán E. (eds.). *Artificial Intelligence in Education. 21st International Conference, AIED 2020, Proceedings, Part II*. Cham: Springer, 2020. P. 61–70. (Lecture Notes in Computer Science. Vol. 12164). DOI: 10.1007/978-3-030-52240-7_61.
- Jiang L., Bosch N. Short answer scoring with GPT-4 // *Proceedings of the Eleventh ACM Conference on Learning @ Scale*. New York: Association for Computing Machinery, 2024. P. 438–442. DOI: 10.1145/3657604.3664685.
- Сердюк Ю. П., Власова Н. А., Момот С. Р., Сулейманова Е. А. Анализ судебных решений с помощью больших языковых моделей // *Программные системы: теория и приложения*. 2026. Т. 17. № 1. С. 21–56. DOI: 10.25209/2079-3316-2026-17-1-21-56.
- Allemang D., Hendler J. *Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS, and OWL*. 2nd ed. Morgan Kaufmann, 2011. 384 p. URL: <https://www.sciencedirect.com/book/9780123859655/semantic-web-for-the-working-ontologist> (дата обращения: 15.02.2026).
- Gupta A., Reddig J.M., Calo T., Weitekamp D., MacLellan C.J. Beyond Final Answers: Evaluating Large Language Models for Math Tutoring // *Proceedings of the International Conference on Artificial Intelligence in Education (AIED 2025)*. Cham: Springer, 2025. P. 323–337. (Lecture Notes in Computer Science. Vol. 15877). DOI: 10.1007/978-3-031-98414-3_23.
- Li J., Lyu Q., Qiu W., Khong A.W. Improving Course Recommendation Systems with Explainable AI: LLM-Based Frameworks and Evaluations // *Proceedings of the 18th International Conference on Educational Data Mining*. 2025. P. 205–214. DOI: 10.5281/zenodo.15870185.
- Швыров В. В., Капустин Д. А., Сентяй П. Н., Шулика Т. И. Анализ наборов данных и больших языковых моделей для обнаружения уязвимостей в программном коде на императивных языках программирования // *Программная инженерия*. 2024. Т. 15. № 11. С. 555–569. DOI: 10.17587/prin.15.555-569. EDN: DAIBNQ.
- Amugongo L. M., Mascheroni P., Brooks S., Doering S., Seidel J. Retrieval augmented generation for large language models in healthcare: A systematic review // *PLOS Digital Health*. 2025. Vol. 4. № 6. Article e0000877. P. 1–15. DOI: 10.1371/journal.pdig.0000877.
- Sarioglu M., Sariyer G., Sozen M. E. LLM-based embeddings for clustering and predicting integrated reporting quality levels of companies // *Discover Computing*. 2025. Vol. 28. Article 95. P. 1–18. DOI: 10.1007/s10791-025-09590-6.
- Комарова Е. В. Выбор промта для больших языковых моделей: деловые коммуникации // *Филология и культура. Philology and Culture*. 2025. № 1 (79). С. 66–74. DOI: 10.26907/2782-4756-2025-79-1-66-74.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.

Финансирование: Авторы заявляют об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.