

ВЕБ-ПЛАТФОРМА ДЛЯ ОБРАЗОВАТЕЛЬНЫХ КУРСОВ С ИНТЕЛЛЕКТУАЛЬНЫМ УПРАВЛЕНИЕМ

Е. А. Цуранков

yauhenitsurankou@gmail.com;

*Научный руководитель — В. А. Нифагин, кандидат физико-математических наук,
доцент*

Целью данной работы является разработка и реализация специализированной платформы для образовательных IT курсов с использованием интеллектуального управления, позволяющего автоматизировать процессы обучения, повысить их качество и эффективность. В рамках данной статьи будут рассмотрены основные принципы работы веб-платформы, ее архитектура и функциональные возможности, а также методы оптимизации процессов при помощи средств машинного обучения.

Ключевые слова: образовательная платформа; GPT; OpenAI; рекомендательный алгоритм; машинное обучение.

Технологии развиваются все быстрее, поэтому бумажные учебники устаревают еще до того, как будут напечатаны и выпущены в продажу. Тем не менее, благодаря этим же технологиям появилась возможность получать знания при помощи дистанционного онлайн-обучения. Для организации такого учебного процесса используются веб-платформы [1].

МЕХАНИЗМ РАБОТЫ ВЕБ-ПЛАТФОРМЫ

Предлагаемая платформа включает следующие компоненты: модуль управления курсами и дисциплинами, менторскую систему, систему управления проектами и модуль управления заявками на участие в курсе. Роль менторской системы состоит в поддержке студентов опытными программистами, которые выступают в роли менторов и преподавателей. Администраторы и модераторы обеспечивают унифицированную работу с изучаемыми дисциплинами, чтобы избежать разночтений в названиях и описании дисциплин для разных курсов платформы. Для оценки знаний студентов реализована система тестирования, которая также позволяет ранжировать студентов на основе успехов и отслеживает их прогресс в течение курса. Упрощенная система создания курсов предоставляет авторам возможность использовать нейросеть в качестве помощника при создании учебного материала. При помощи нейросети реализована генерация описания курса, создание материалов курса на основе предпочтений автора.

Система управления проектами используется для отслеживания выполнения заданий курса. Задания представлены в виде списка или доски задач со статусами, а система управления является аналогом системы Jira [2].

Структура платформы подразделяется на несколько независимых частей: клиентская часть, в которую входит два приложения на фреймворке Angular, серверную часть с приложением на платформе Node.js с использованием фреймворка Nest.js и базы данных PostgreSQL, и две сопровождающие страницы с системой управления контентом для каждого из клиентских приложений. В качестве способа организации кода выбрана структура монорепозитория [3]. При разработке архитектуры платформы использовалась практика DDD (Domain-Driven Design) [4].

СРЕДСТВА ИНТЕЛЛЕКТУАЛЬНОГО УПРАВЛЕНИЯ

При помощи языка Python и библиотеки scikit-learn реализован рекомендательный алгоритм для курсов платформы. Рассмотрим теоретические основы реализованного алгоритма.

Контентный рекомендательный алгоритм – это метод, используемый для предоставления рекомендаций пользователям на основе анализа характеристик контента и сопоставления с профилем предпочтений пользователя [5]. Процесс разработки метода контентной фильтрации включает следующие шаги: представление контента в виде характеристик, создание профилей предпочтений пользователей, извлечение признаков с использованием TF-IDF для оцифровки данных, оценку схожести между курсами на основе их содержания.

TF-IDF – статистическая мера, используемая для оценки важности слова в документе, являющегося частью коллекции документов [6].

Формула для определения меры имеет вид: $TF - IDF = TF \times IDF$, где TF – частота слова в конкретном документе, IDF – обратная частота документа.

Частота слова в документе определяется следующей формулой: $TF = \frac{n_t}{\sum_{i=1}^k n_i}$, где n_t – количество отдельных слов в документе, $\sum_{i=1}^k n_i$ – общее количество всех слов в документе. Обратная частота документа определяется так: $IDF = \ln\left(\frac{n_c}{\sum_{m=1}^l n_m}\right)$, где n_c – количество всех документов, $\sum_{m=1}^l n_m$ – количество документов, в которых содержится интересное слово.

Оценка схожести между курсами на основе их содержания выполнена с использованием метрики косинусного сходства, используемой для оценки схожести между двумя векторами [5]. Формула для вычисления косинусного сходства: $\text{cosine_sim}(A, B) = (A \times B) : (||A|| \times ||B||)$, где A и B представляют собой два вектора, $(A \times B)$ представляет скалярное произведение между векторами, $||A||$ и $||B||$ представляют собой нормы векторов A и B соответственно.

Коллаборативный алгоритм – метод, используемый для предоставления рекомендаций на основе схожести предпочтений пользователя с предпочтениями других пользователей [5]. Он основывается на предположении, что если два пользователя имеют похожие предпочтения для определенного набора элементов, то их вкусы будут схожи и в других случаях.

Гибридный рекомендательный алгоритм – метод, который комбинирует контентную и коллаборативную фильтрации для предоставления более точных и персонализированных рекомендаций. В работе используется вариант каскадного применения, при котором первым шагом алгоритм использует контентную фильтрацию для сужения множества всех элементов, после чего применяется коллаборативная фильтрация для получения конечного списка рекомендаций.

Другим важным аспектом интеллектуального управления является интеграция языковой модели GPT [7]. Языковые модели – это алгоритмы машинного обучения, которые генерируют и обрабатывают текст на естественном языке [8].

Процесс создания курса с использованием GPT включает следующие шаги:

Указание основных параметров курса, таких как название, продолжительность, уровень сложности, язык программирования и технологии, которые будут рассмотрены в курсе. Данные параметры помогают GPT понять контекст и ориентироваться при генерации описания.

Общение с ChatGPT, описание требований к содержанию курса. GPT предлагает варианты описания, из которых автор выбирает наиболее подходящий и вносит свои дополнения.

Генерация содержания курса, включая разделение на модули, определение тем и установку целей для каждого модуля, а также создание практических заданий и тестов для оценки прогресса студентов.

Рецензирование и корректировка сгенерированного описания и содержания курса другими авторами или модераторами для обеспечения точности, полноты и качества материала.

Система тестирования использует модель GPT-3.5 для оценки качества кода практических заданий. Модель может интерпретировать код, опираясь на контекст и усвоенные знания, способна обнаруживать дублирование кода, проверять и оценивать его стиль, эффективность, читаемость, а также выявлять потенциальные уязвимости и другие проблемы. Важно отметить, что GPT-3.5 не специализирована под проверку кода, ее оценки и рекомендации основаны на общих практиках программирования и могут не учитывать специфические требования или контекст конкретного проекта.

Спроектирована расширяемая и легко поддерживаемая архитектура с использованием практик доменно-ориентированного дизайна приложений и структуры монорепозитория.

Разработаны клиентские приложения для управления проектами платформы и курсами платформы с использованием фреймворка Angular.

Создано серверное приложение на платформе Node.js с использованием фреймворка Nest.js и базы данных PostgreSQL.

Реализована система интеллектуального управления, включающая в себя рекомендательный алгоритм, систему автоматизированной проверки кода, систему отправки уведомлений, а также систему создания курсов при помощи языковой модели GPT 3.5.

Использование интеллектуального управления помогает создать более эффективные, персонализированные и качественные курсы, что обеспечивает более глубокое и полноценное обучение учащихся, повышает уровень образования и способствует развитию способностей студентов.

Библиографические ссылки

1. *Тиунова, Н.Н.* Образовательные платформы как средство интенсификации профессиональной подготовки студентов колледжа // *Профессиональное образование в России и за рубежом.* 2016. № 2. С. 103–108.