

Образовательный чат-бот в Telegram с AI-генерацией тестов

Владимир В. Лапин¹, Оксана А. Естафиади¹, Алексей С. Смирнов², Алина Р. Багаутдинова³

¹ ГБОУ СК «Гимназия №25», Ставрополь

² ЦДО «Поиск», Ставрополь

³ Северо-Кавказского федерального университета, Ставрополь

Аннотация

В современном мире, где доступ к информации и обучению становится все более мобильным и персонализированным, возникает потребность в инструментах, способных адаптироваться к индивидуальным запросам пользователей. Чат-боты в мессенджерах, таких как Telegram, уже доказали свою эффективность в различных областях, включая образование. Они позволяют автоматизировать процессы обучения, предоставляя пользователям возможность получать знания в удобное время и в интерактивной форме. Разработан образовательный чат-бот для Telegram, использующий искусственный интеллект (AI) для генерации тестов, управления классами и анализа результатов. Бот автоматизирует процесс создания учебных материалов, адаптирует задания под уровень знаний учащихся и предоставляет учителям аналитику. Проект направлен на повышение эффективности и доступности образовательного процесса.

Ключевые слова

чат-бот, Telegram, искусственный интеллект, генерация тестов, образование, персонализированное обучение

1. Введение

Использование искусственного интеллекта для генерации тестов добавляет новое измерение в образовательные технологии. AI способен анализировать уровень знаний пользователя, подбирать задания соответствующей сложности и создавать уникальные тесты, что делает процесс обучения более гибким и эффективным. Это особенно важно в условиях, когда традиционные методы обучения не всегда могут удовлетворить потребности современных учащихся, стремящихся к самостоятельному и непрерывному образованию.

Кроме того, Telegram как платформа для образовательного бота обладает рядом преимуществ: высокая популярность среди пользователей, кроссплатформенность, удобство использования и возможность интеграции с другими сервисами. Это делает его идеальным инструментом для внедрения инновационных образовательных решений. Таким образом, разработка образовательного чат-бота с функцией AI-генерации тестов не только отвечает текущим трендам цифровизации образования, но и открывает новые возможности для повышения качества и доступности обучения.

Внедрение образовательных чат-ботов представляет собой многоэтапный процесс, который требует учета технических, педагогических и организационных аспектов. Одной из ключевых особенностей является интеграция бота в существующую образовательную экосистему, что подразумевает не только техническую настройку, но и адаптацию пользователей к новому инструменту. На начальном этапе важно определить целевую аудиторию и задачи, которые будет решать бот. Например, это может быть автоматизация проверки знаний, генерация

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2025 Workshop Proceedings

DOI: 10.47350/ICCS-DE.2025.39

индивидуальных заданий или предоставление справочной информации. После определения целей необходимо разработать архитектуру бота, которая включает в себя модули для обработки запросов, хранения данных и взаимодействия с пользователями. Важным элементом является выбор платформы для разработки, такой как Telegram, которая предоставляет удобный API и широкие возможности для интеграции.

Следующий этап — это создание контента и алгоритмов работы бота. Для образовательных чат-ботов особенно важно обеспечить качественную генерацию тестов и заданий, что требует использования специализированных алгоритмов и баз данных. Например, можно использовать шаблоны для создания вопросов разных типов: множественный выбор, открытые вопросы или задания на соответствие. Кроме того, необходимо предусмотреть возможность адаптации заданий под уровень знаний пользователя, что может быть реализовано с помощью анализа предыдущих ответов и результатов.

Особое внимание следует уделить тестированию и доработке бота перед его запуском. На этом этапе важно проверить работу всех функций и выявить возможные ошибки. Например, можно провести тестовый запуск среди небольшой группы пользователей и собрать их отзывы для улучшения функционала. После запуска необходимо регулярно обновлять контент и алгоритмы работы бота, чтобы он оставался актуальным и полезным для пользователей.

Таким образом, схема архитектуры чат-бота демонстрирует последовательность взаимодействия между пользователем, платформой Telegram, сервером, базой данных и AI-модулем. Каждый элемент системы играет важную роль в обеспечении функциональности и удобства использования образовательного чат-бота.

2. Архитектура Телеграмм-бота

Архитектура Telegram-бота состоит из нескольких ключевых компонентов:

Telegram API: предоставляет API для взаимодействия с ботами. Через API бот получает сообщения от пользователей и отправляет ответы. Для работы с API используется библиотека `python-telegram-bot`, которая упрощает взаимодействие.

Ядро бота (логика): это основной модуль, который обрабатывает входящие сообщения, выполняет команды и генерирует ответы. Логика может включать обработку текста, вызов внешних API (например, для генерации тестов), работу с базами данных и т.д.

База данных (опционально): Для хранения данных пользователей, результатов тестов или других данных используется база данных (например, SQLite, PostgreSQL или MongoDB).

Внешние сервисы (опционально): бот может взаимодействовать с внешними сервисами, такими как AI-модели для генерации контента, платежные системы или облачные хранилища.

Сервер: Бот должен быть размещен на сервере, чтобы работать 24/7. Для этого можно использовать облачные платформы, такие как Heroku, AWS или VPS.

Ниже на рисунке 1 приведен пример простого Telegram-бота, который генерирует тесты на основе входящих сообщений.

Для работы с Telegram API и генерацией текста установим библиотеки: `pip install python-telegram-bot transformers`

```

4
5 # Инициализация модели для генерации вопросов
6 question_generator = pipeline("text2text-generation", model="valhalla/t5-small-qa-qg-hl")
7 # Функция для генерации вопросов
8 def generate_questions(text, num_questions=3):
9     questions = []
10    for _ in range(num_questions):
11        result = question_generator(text, max_length=50)
12        question = result[0]['generated_text'].strip()
13        questions.append(question)
14    return questions
15
16 # Команда /start
17 async def start(update: Update, context: CallbackContext):
18     await update.message.reply_text("Привет! Отправь мне текст, и я создам тест.")
19
20 # Обработка текстовых сообщений
21 async def handle_message(update: Update, context: CallbackContext):
22     text = update.message.text
23     questions = generate_questions(text)
24     # Формируем тест
25     test = "Бот твой тест:\n\n"
26     for i, question in enumerate(questions):
27         test += f"{i + 1}. {question}\n"
28     await update.message.reply_text(test)
29
30 # Основная функция
31 def main():
32     # Создаем приложение бота
33     application = ApplicationBuilder().token("YOUR_TELEGRAM_BOT_TOKEN").build()
34     # Регистрируем обработчики команд и сообщений
35     application.add_handler(CommandHandler("start", start))
36     application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, handle_message))
37     # Запускаем бота
38     application.run_polling()
39
40 if __name__ == "__main__":
41     main()

```

Рисунок 1: Пример простого Telegram-бота, который генерирует тесты на основе входящих сообщений.

Этот код представляет собой реализацию Telegram-бота на Python, который использует библиотеку `python-telegram-bot` для взаимодействия с Telegram API и модель `transformers` для генерации вопросов на основе текста. Бот позволяет пользователю отправлять текст, после чего автоматически генерирует тест с вопросами и отправляет его обратно пользователю.

Каждый компонент системы выполняет определенные функции:

Пользователь: Взаимодействует с ботом через Telegram, отправляет команды и получает обратную связь.

Telegram Bot: Обрабатывает команды, управляет взаимодействием между пользователем и API, осуществляет подключение к Yandex GPT.

Yandex GPT API: Создает тестовые задания и управляет их генерацией.

Классы (bot_data): Хранит информацию о классах, их студентах и кодах доступа.

Тесты (bot_data): Хранит информацию о созданных тестах и их параметрах.

Результаты (bot_data): Хранит результаты тестирования, включая детализированные данные по каждому ученику.

3. Используемые технологии

Для реализации проекта используются следующие технологии:

1. Python 3.10+: Основной язык разработки, выбранный за его простоту и мощные библиотеки для работы с API и асинхронного программирования.
2. `pyTelegramBotAPI`: Библиотека для работы с Telegram Bot API, позволяющая легко управлять сообщениями и командами.
3. `aiohttp`: Асинхронная библиотека для выполнения HTTP-запросов. Обеспечивает эффективность и позволяет обрабатывать несколько запросов одновременно, что критично для работы с API.
4. JSON: Формат для хранения данных — используется локально для хранения информации между сессиями работы.

На рисунке 2 представлена работа Телеграмм-бота

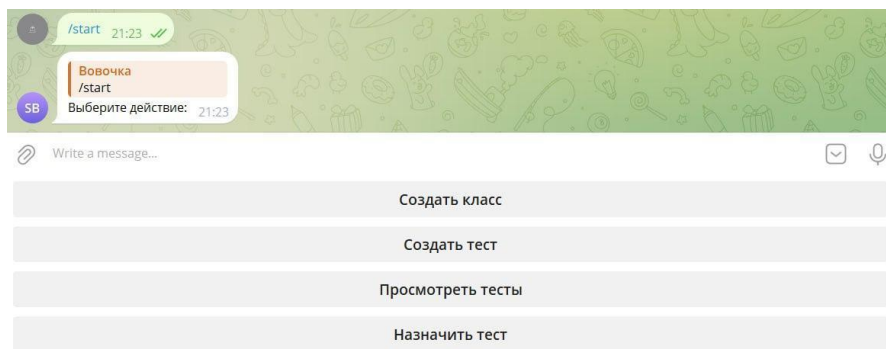


Рисунок 2: Начало работы Телеграмм-бота



Рисунок 3: Просмотр тестов чат-бота

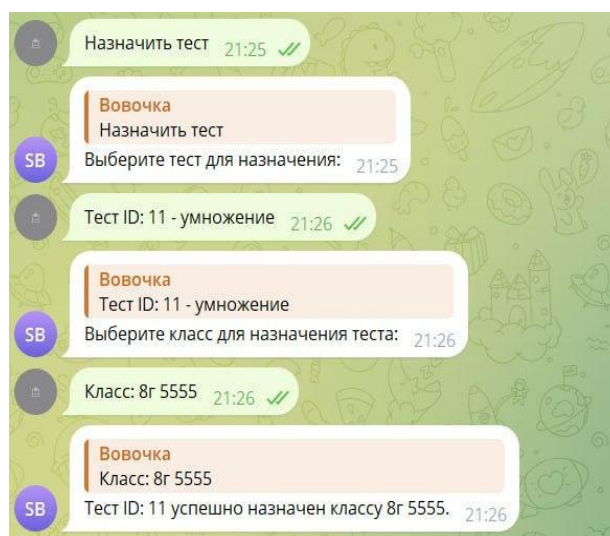


Рисунок 4: Работа чат-бота

4. Вывод

Проект демонстрирует эффективное использование современных технологий, таких как Telegram Bot API и Yandex GPT, для решения задач в сфере образования. Разработка бота, ориентированного на автоматизацию учебных процессов, создает новые возможности для оптимизации работы учителей, повышения вовлеченности учеников и улучшения качества образовательного процесса в целом. Telegram Bot API предоставляет удобный и мощный инструмент для создания ботов, которые могут взаимодействовать с пользователями в режиме реального времени, делая образовательный процесс более доступным и интерактивным. Бот выступает в роли посредника между учениками и учебными ресурсами, позволяя автоматизировать такие задачи, как генерация тестов, проверка ответов и анализ результатов. Интеграция Yandex GPT, мощной языковой модели, позволяет боту генерировать осмысленные и релевантные вопросы на основе предоставленного текста, что значительно экономит время учителей и повышает качество учебных материалов. Учителя получают возможность сосредоточиться на индивидуальной работе с учениками, в то время как бот берет на себя рутинные задачи, такие как создание тестов и проверка ответов. Для учеников бот становится удобным инструментом, который адаптирует тесты под их уровень знаний, предоставляет мгновенную обратную связь и делает обучение более персонализированным и интересным. Использование элементов геймификации, таких как баллы и рейтинги, повышает мотивацию учеников и делает учебный процесс более увлекательным. Кроме того, бот собирает данные о результатах тестирования, что позволяет учителям анализировать прогресс учеников и корректировать учебный процесс в соответствии с их потребностями. Таким образом, проект не только оптимизирует работу учителей, но и создает условия для более эффективного и вовлеченного обучения, демонстрируя, как современные технологии могут быть использованы для улучшения качества образования.

5. Благодарность

Проект реализован при поддержке ГБОУ СК «Гимназия №25» и «Центра Поиск».

6. Список литературы

- [1] Байкова, К. Д. "Анализ и разработка функционального Telegram-бота." *Молодой исследователь Дона* 6 (33) (2021): 15-20.
- [2] Гатулин, Р. Р., and Д. А. Колупаева. "Использование мессенджера Telegram для реализации технологии электронного обучения в вузе." *Санкт-Петербургский образовательный вестник* 11-12 (15-16) (2017): 31-33.
- [3] Филонов, Дмитрий Русланович, and Владислав Игоревич Тупикин. "Чат-бот для Telegram для помощи абитуриентам." *Заметки по информатике и математике*. 2017. 152-156.
- [4] Валинурова, Анна Александровна, Наталья Владимировна Балабанова, and Иван Алексеевич Маценков. "АЛГОРИТМ РАЗРАБОТКИ TELEGRAM-БОТА-ПРОДУКТИВНОГО ПОМОЩНИКА СОВРЕМЕННОГО БИЗНЕСА." *Современные наукоемкие технологии. Региональное приложение* 2 (74) (2023): 60-67.
- [5] Вяткин, Александр Николаевич, and Татьяна Инакентиевна Русских. "Telegram-боты и их разнообразие." *Наука XXI века: технологии, управление, безопасность: Материалы II национальной научной конференции, Курган, 21 апреля 2022 года*. 2022.
- [6] Вяткин, Александр Николаевич, and Татьяна Инакентиевна Русских. "Telegram-боты и их разнообразие." *Наука XXI века: технологии, управление, безопасность: Материалы II национальной научной конференции, Курган, 21 апреля 2022 года*. 2022.