

УДК 004.415

Ю. С. Гаврилова, В. Р. Смородин, С. М. Бараксанов

Yu. S. Gavrilova, V. R. Smorodin, S. M. Baraksanov

Гаврилова Юлия Сергеевна, руководитель лаборатории интеллектуального анализа данных, КГПИ ФГБОУ ВО «КемГУ», г. Новокузнецк, Россия.

Смородин Валерий Русланович, ведущий программист лаборатории интеллектуального анализа данных, КГПИ ФГБОУ ВО «КемГУ», г. Новокузнецк, Россия.

Бараксанов Сергей Максимович, студент, КГПИ ФГБОУ ВО «КемГУ», г. Новокузнецк, Россия.

Gavrilova Yulia Sergeevna, Head of the Laboratory of Data Mining, Kuzbass Humanitarian Pedagogical Institute of Kemerovo State University, Novokuznetsk, Russia.

Smorodin Valery Ruslanovich, the leading programmer of the laboratory of intellectual data analysis, Kuzbass Humanitarian Pedagogical Institute of Kemerovo State University, Novokuznetsk, Russia.

Baraksanov Sergey Maksimovich, student, Kuzbass Humanitarian Pedagogical Institute of Kemerovo State University, Novokuznetsk, Russia.

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО АССИСТЕНТА ПРЕПОДАВАТЕЛЯ

DEVELOPMENT OF AN INTELLIGENT TEACHER ASSISTANT

Ю. С. Гаврилова, В. Р. Смородин, С. М. Бараксанов 2025-06-23

Аннотация. В статье рассматривается разработка интеллектуального ассистента преподавателя в виде Telegram-бота, предназначенного для автоматизации рутинных задач. Описываются реализованные функции бота, а также инструменты управления для преподавателей. Применение такого ассистента позволяет снизить нагрузку на преподавателей и повысить доступность обучения.

Annotation. The article discusses the development of an intelligent teaching assistant in the form of a Telegram bot designed to automate routine tasks. The implemented functions of the bot are described, as well as management tools for teachers. The use of such an assistant makes it possible to reduce the burden on teachers and increase the availability of training.

Ключевые слова: telegram-бот, искусственный интеллект, нечеткая логика, расстояние Левенштейна.

Keywords: telegram bot, artificial intelligence, fuzzy logic, Levenshtein distance.

Современная система образования все чаще сталкивается с необходимостью внедрения цифровых технологий для повышения эффективности учебного процесса, персонализации обучения и уменьшения не связанной с проведением занятий нагрузки преподавателей [2]. Одним из перспективных направлений внедрения цифровых технологий в образовательный процесс является разработка интеллектуальных ассистентов преподавателя – программных решений на основе искусственного интеллекта, способных автоматизировать рутинные задачи.

Интеллектуальные ассистенты могут взаимодействовать с клиентами, выполняя роль первой линии технической поддержки, операторов сбора отзывов и предложений (обратной связи), администраторов, напоминающих о записи на оказание услуги и т.п. Реализуют таких ассистентов, как правило, в популярных социальных сетях или мессенджерах. Например, Telegram-боты уже стали неотъемлемой частью цифровой экосистемы многих организаций, сочетая в себе простоту использования, широкие возможности автоматизации различных информационных процессов и интеграции с различными сервисами [1]. Применение же чат-ботов студентам на регулярной основе на занятиях и во время самоподготовки позволяет им отработать выполнение различных базовых заданий и снизить страх перед ошибками за счет отсутствия прямого контроля со стороны преподавателя [3].

Telegram-бот, предназначенный для помощи преподавателю, поможет студентам решить следующие вопросы, связанные с их обучением:

1. Функция ответа на вопросы. Часто у студентов возникают различные вопросы по пройденному материалу, и чтобы их задать – нужно дождаться занятия или прийти на консультацию. Данная функция бота позволяет студентам получить ответы на интересующие их вопросы. Для активации данной функции студенту необходимо написать в поле ввода сообщения вопрос, после чего программа найдет на него ответ в базе данных и передаст его пользователю. Для того чтобы избежать дополнения различных формулировок одного и того же вопроса (например, «Что такое база данных?» и «Расскажи про базы данных») была реализована нечеткая логика, позволяющая задавать вопросы в разных формулировках.
2. Функция диалогов. Данная функция позволяет студенту написать сообщение другому студенту или преподавателю. Внедрение такой функции в Telegram-бот позволяет применять технологию взаимного обучения (peer-to-peer learning): студент может задать

вопрос другому студенту и попросить помощи. Для того чтобы начать работу с данной функцией, зарегистрированный студент должен сообщить боту о необходимости отправки сообщения. После того как бот получает на вход команду для отправки сообщения, он предлагает студенту выбрать пользователя-получателя с помощью инлайн-кнопок, расположенных под сообщением. После выбора получателя бот просит ввести сообщение, которое будет отправлено получателю.

3. Регистрация. Данная функция позволяет пользователю пройти регистрацию для того, чтобы получить доступ к функциям тестирования и диалогов.
4. Тестирование. Данная функция позволяет пользователю пройти тестирование в ходе текущего контроля (например, таким тестированием можно заменить устный опрос на занятии) [5]. Для того чтобы начать тестирование зарегистрированный пользователь выбирает дисциплину для тестирования, написав ее название боту. После выбора предмета бот сообщит пользователю об установленных параметрах тестирования и попросит подтвердить готовность, написав «Да» или отказаться от тестирования, написав «Нет» (рис. 1). В случае готовности к тестированию бот генерирует тест из случайных вопросов, дисциплина которых совпадает с выбранной пользователем.

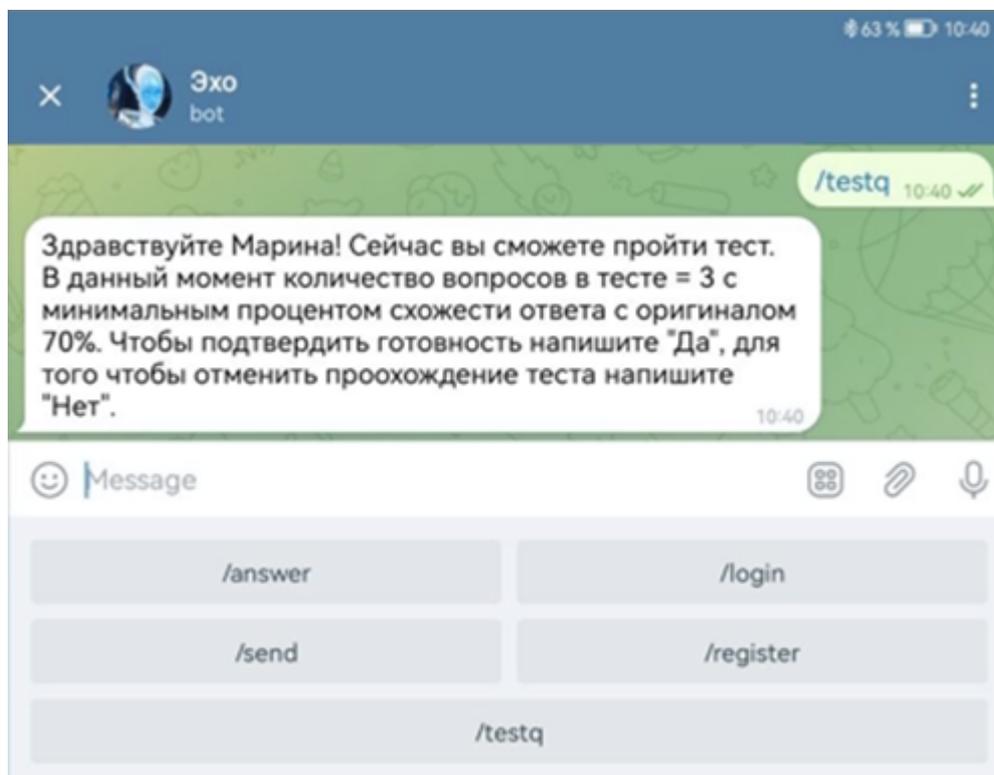


Рисунок 1. Запуск функции тестирования

Для того чтобы пройти авторизацию с правами «Преподаватель» необходимо написать боту команду, после чего он запросит пароль. В случае совпадения хеша полученного на вход пароля с хешем пароля, хранящегося в базе данных, бот сообщит об успешной авторизации, после чего автоматически даст доступ к служебным командам. В случае несовпадения бот сообщит об этом пользователю и прервет процесс авторизации, соответственно, доступа к служебным командам у такого пользователя не будет. Интеллектуальный ассистент не позволяет преподавателю дважды указать себе один и тот же предмет, а в случае, если с другого устройства зайдет пользователь с такими же данными авторизации, то бот сообщит об этом и предупредит, что доступ к служебным командам ограничен.

Рассмотрим функционал доступный преподавателю:

1. Добавление и удаление вопроса из базы данных. Исходя из названия, данные функции позволяют редактировать базу данных вопросов и ответов без обращения к СУБД [4]. При добавлении вопроса в базу

- данных учитывается предмет, для которого корректируется список вопросов.
2. Удаление аккаунта пользователя. Данная функция позволяет удалить аккаунт пользователя, ограничив ему доступ к командам, требующим авторизации.
 3. Получения списка вопросов и ответов. После ввода данной команды бот отправит пользователю список вопросов и ответов.
 4. Составление отчета по тестированию. Данная функция позволяет получить результаты тестирования по выбранным предметам. Для получения отчета необходимо написать фразу «Сформируй отчет по тестированию» и указать название предмета так, как он указан в базе данных. После написания команды бот сформирует отчет в виде таблицы в формате *.xls и передаст его пользователю.
 5. Предоставление или ограничение доступа к ответам на вопросы. Данная функция позволяет пользователям, авторизованным в качестве преподавателей, предоставлять или блокировать доступ к функции ответа на вопросы. Блокировка функции ответа на вопросы реализована для того, чтобы у студентов не было доступа к ответам на время тестирования.
 6. Очистка результатов тестирования. Данная функция очищает таблицу, в которой хранятся результаты тестирования, в случае если она перегружена данными.
 7. Редактирование параметров тестирования. Данная функция позволяет задать количество вопросов в тесте, а также указать минимальный процент схожести, при котором ответ на вопрос будет считаться правильным.

Рассмотрим модель общения с интеллектуальным ассистентом преподавателя. Получаемый на вход текст классифицируется как вопрос или активатор действия. Алгоритм, позволяющий отличить вопрос от активатора действия выглядит следующим образом:

1. Программа получает на вход сообщение пользователя.
2. Формируется список имеющихся в базе данных вопросов.
3. Проводится сравнение полученного на вход текста с каждым вопросом из базы данных. Проценты схожести полученного от пользователя текста с текстом вопроса из базы данных добавляются в список.
4. Проводится сравнение полученного на вход текста с каждым элементом списка, в котором хранятся шаблоны для фраз-функций. Проценты схожести полученного текста также добавляются в отдельный список.
5. Из каждого списка выбираются максимальные проценты схожести.
6. Проводится сравнение. Если значение, полученное в результате действий 3 и 5 больше, чем значение, полученной в результате действий 4 и 5, то полученный на вход текст классифицируется как вопрос и программа запускает алгоритм поиска ответа. В противном случае запускается алгоритм выполнения команды.

При ответе на вопрос алгоритм нечеткого поиска аналогичен рассмотренному ранее алгоритму:

1. Полученный на вход текст сравнивается с каждым вопросом из базы данных.
2. Проценты схожести введенного пользователем текста и текста вопроса из базы данных добавляются в список.
3. Вычисляется позиция вопроса, у которого процент схожести с полученным на вход текстом максимален.

4. Программа выдает пользователю ответ, находящийся в базе данных на позиции, полученной из действия 3.
5. В случае если полученный максимальный процент схожести меньше 70 %, программный продукт выдает сообщение о том, что ответ на вопрос не найден и предлагает адресовать этот вопрос преподавателю.

Разработанный алгоритм позволяет писать команды и вопросы к боту в формулировках, отличающихся от стандартных (рис. 2), что обеспечивает большую гибкость при общении с ассистентом.

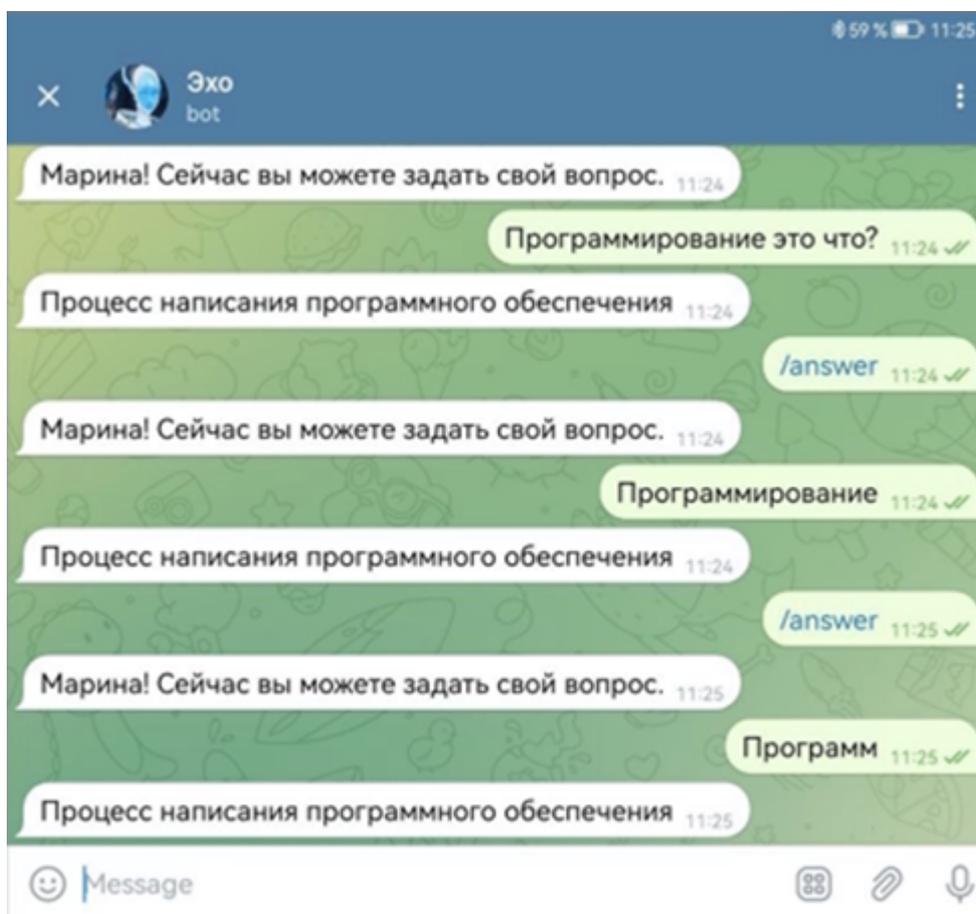


Рисунок 2. Реакция бота на разные формулировки вопроса

Для вычисления рассмотренного ранее процента схожести применяется формула расстояния Левенштейна (редакционное расстояние или дистанция редактирования), которое является метрикой, используемой для определения минимального количества операций редактирования, необходимых для превращения одной строки в другую. Функция расстояния Левенштейна определяет стоимость трех основных операций:

1. Вставка (Insertion) – добавление одного символа в строку.
2. Удаление (Deletion) – удаление одного символа из строки.
3. Замена (Substitution) – замена одного символа другим.

Каждой из этих операций присваивается некоторая стоимость, а затем функция расстояния Левенштейна вычисляет минимальную стоимость последовательности операций, которая превращает одну строку в другую. Одинаковые строки имеют нулевое расстояние: не нужно ничего редактировать, первая и так равна второй. «Привет» и «Привт» имеют расстояние 1 (пропущена одна буква, остальное не изменилось). «Привет» и «привты» имеют расстояние 3 (одна замена «П» на «п», удаление буквы «е» и вставка «ы» в конце слова).

Следует отметить, что интеллектуальный ассистент, обладающий таким функционалом, требует тщательной настройки: преподаватель должен будет сформировать и загрузить в бот список вопросов по своим дисциплинам, а также постоянно обновлять его, поддерживая актуальность по читаемым им темам. Этот процесс может занять достаточно продолжительное время, к тому же необходимо будет тщательно выбрать вопросы, добавляемые в базу данных: добавление простых вопросов (например, определение какого-либо понятия) не имеет смысла, поскольку ответ на них можно найти, воспользовавшись поиском по сети Интернет. Однако, полученные в результате применения такого бота преимущества позволят разнообразить образовательный процесс, сократить время, в течение которого студент получит ответ на свой вопрос, и снизить нагрузку на преподавателя от большого числа обращающихся к нему с вопросами студентов.

Таким образом, разработанный интеллектуальный ассистент преподавателя на базе Telegram-бота демонстрирует эффективное применение искусственного интеллекта и технологий автоматизации в образовательном процессе. Реализованный функционал, включая нечеткий поиск ответов, управление тестированием и диалоговую систему, позволяет снизить нагрузку на преподавателей за счет автоматизации рутинных операций, повысить доступность обучения благодаря круглосуточной поддержке студентов и обеспечить персонализацию через адаптивные тесты и гибкую систему взаимодействия.

Список литературы

1. Абрамова, А. И. Использование telegram-бота в образовательном процессе вуза / А. И. Абрамова - Текст : непосредственный. // Вестник науки. - 2022. - № 1 (46).
2. Гаврилова, Ю. С. Обзор игровых образовательных проектов по дисциплине «Базы данных» / Ю. С. Гаврилова. - Текст : непосредственный. // Развитие науки и практики в глобально меняющемся мире в условиях рисков : сборник материалов XXI Международной научно-практической конференции,

Москва, 15 сентября 2023 года. – Москва: Алеф, 2023. – С. 15-21.

3. Дукальская, И. В. Чат-боты в приложении Telegram как средство изучения английского языка / И. В. Дукальская, Е. О. Аликберова. – Текст : непосредственный. // Преподаватель XXI век. – 2023. – № 2-2. – С. 434-442.
4. Журавлев, А. А. Создание телеграм-бота с помощью среды разработки Visual Studio / А. А. Журавлев. – Текст : непосредственный. // Научно-образовательный журнал для студентов и преподавателей «StudNet». – 2020. – № 12. – С. 271-277.
5. Могилевская, Н. С. Чат-бот для тестирования знаний на платформе мессенджера Telegram / Н. С. Могилевская, Г. П. Самойленко. – Текст : непосредственный // Молодой исследователь Дона. – 2022. – № 3 (36).

© Гаврилова Ю. С., Смородин В. Р., Бараксанов С. М., 2025