**НИТУ «МИСИС»**

**Кафедра инженерной кибернетики**

**ОТЧЕТ**

по

**ЛАБОРАТОРНОЙ РАБОТЕ №3**

*«Разработка чат-бота с помощью конструктора для мессенджера Telegram»*

учебная дисциплина

«Методы искусственного интеллекта»

**Студент: Прокуденко Кирилл Игоревич**

**Группа:** \_\_БПМ-21-3**\_\_\_\_**

**Преподаватель: Хонер П. Д.**

**Оценка: \_\_\_\_\_\_\_\_\_**5**\_\_\_\_\_\_\_\_**

**Дата защиты: \_\_**02.12.24\_\_\_\_

**2024 г.**

Содержание

[1. Введение 3](#_Toc183967601)

[2. Описание задачи 3](#_Toc183967602)

[3. Выбранные средства для разработки ПО 3](#_Toc183967603)

[4. Исходные данные 3](#_Toc183967604)

[5. Результат 4](#_Toc183967605)

[6. Анализ полученных результатов 8](#_Toc183967606)

[7. Выводы по лабораторной работе 9](#_Toc183967607)

[8. Список использованных источников 9](#_Toc183967608)

# Введение

В данном лабораторной работе целью является обрести навыки по создания чат-ботов для мессенджера Telegram без разработки программного кода с использованием специализированного инструментария – конструктора чат-ботов.

# Описание задачи

Требуется разработать Telagram-бота с уровнем вложенности не менее 5. Целью бота будет квиз по компьютерной игре Dota 2 со случайными вопросами, которые будут генерироваться с помощью NLP-инструментов.

# Выбранные средства для разработки ПО

Для выполнения задачи был выбран онлайн конструктор Botmother. В качестве API для генерации текста была взята модель Qwen 2.5, взаимодействие с которой будет осуществляться с помощью сайта HuggingFace.

# Исходные данные

Для генерации вопросов и ответов на них использовались следующие промты.

* **Системный промпт:** «Ты — интеллектуальный помощник для создания вопросов и ответов квиза по игре Dota 2. Следуй следующим правилам:

**Тематика**: Все вопросы должны строго относиться к игре Dota 2.

Все ответы должны быть короткими, на английском языке и 100% соответствовать игре (каждый свой ответ, сначала проверь на корректность, и потом измени на верный). В итоге ты должен вывести только верный ответ без лишних комментариев.

**Оригинальность**: Каждый новый вопрос должен быть уникальным и не повторять предыдущий. В итоге формулировка: Генерируй вопросы четко и однозначно, избегай двусмысленности. Они должны быть понятными и не слишком длинными.

**Уровень сложности**: Вопросы могут быть разной сложности (от базовых до глубоких знаний игре), но избегай слишком простых и тривиальных вопросов.

**Ответ**: Отвечай на запрос пользователя только текстом самого ответа, без пояснений или лишних комментариев.

**Стиль**: Вопросы должны быть на русском языке, грамматически правильными и соответствовать стилистике произведения.

Примеры недопустимых вопросов: Вопросы, уже заданные ранее, или вопросы, не имеющие отношения к Dota 2».

* **Промпт для генерации вопроса:** «Сгенерируй вопрос для квиза по игре Dota 2, который не будет повторяться с твоими предыдущими вопросами и обязательно будет отличаться от **{{last\_question}}**. В качестве твоего ответа на мой этот запрос напиши только сам сгенерированный новый вопрос на русском языке». Данные вопрос записывается в **“generated\_question”.**
* **Промпт для генерации неверного ответа:** «Сгенерируй 1 максимально краткий неправильный, но подходящих по теме ответ на вопрос: **{{generated\_question**}}. Напиши только сам ответ, без твоих лишних комментариев». Данные ответ записывается в **“right\_answer”.**
* **Промпт для генерации верного ответа:** «Ответь правильно и максимально кратко на вопрос по Dote 2: **{{generated\_question}}**. Напиши только сгенерированный тобой ответ без лишних комментариев». Данные ответ записывается в **“wrong\_answer”.**

# Результат

Экраны бота в порядке из возможного вызова:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис. 1 – Часть 1 структуры бота.

Изображение выглядит как текст, программное обеспечение, число, Значок на компьютере

Автоматически созданное описание

Рис. 2 – Часть 2 структуры бота.

Изображение выглядит как текст, программное обеспечение, дизайн, снимок экрана

Автоматически созданное описание

Рис. 3 – Часть 3 структуры бота.

Изображение выглядит как текст, снимок экрана, диаграмма, программное обеспечение

Автоматически созданное описание

Рис. 4 – Общая структуры бота.

Задача бота задавать пользователю случайные вопросы по теме и предлагать ответить на них пользователю.

После неправильного ответа пользователю будет показан его счёт – на сколько вопросов подряд он смог верно ответить.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рис. 5 – Пример задачи вопросов ботом.

Если пользователь неверно ответит на вопрос, то квиз закончится и бот выведет на сколько вопросов подряд смог ответить игрок.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рис. 6 – Пользователь неверно ответил на вопрос.

Если же человек захочет попробовать ещё раз, то квиз начнётся заного и счёт будет сброшен. В другом случае бот остановит свою диалог с человеком.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 7 – Пользователь не захотел продолжать квиз.

Чтобы запустить бота заново нужно воспользоваться кнопками из меню, где также закончить квиз в любой момент.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 8 – Меню бота.

# Анализ полученных результатов

Как видно из примеров работы, бот может успешно взаимодействовать с пользователем и генерировать необходимый вопросы. Но из-за нестабильности API или багов конструктора бот может в случайный момент перестать корректно обрабатывать запросы. Чтобы это починить приходится в конструкторе удалять и заново вставлять промпт.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рис. 9 – Пример сбоя работы бота.

Также API иногда может генерировать неосмысленные вопросы и ответы на них.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рис. 10 – Пример некорректной работы API.

# Выводы по лабораторной работе

По представленным результатам можно понять, что Botmother является хороший и довольно интуитивным инструментом для создания Telegram-ботов. При этом этот конструктор поддерживает интеграцию сторонних API, но иногда некорректно обрабатывает встраивание строковых переменных в тело запроса.

Документация конструктора также является довольно понятной, легко находится, есть как на английском, так и на русском языках.

# Список использованных источников

1. Документация Botmother // Botmother Knowledge Base URL: <https://docs.botmother.com/> (дата обращения: 26.11.24).
2. Документация модели Qwen2.5 // Hugging Face URL: <https://huggingface.co/Qwen/Qwen2.5-72B-Instruct> (дата обращения: 26.11.24).