Санкт-Петербургский Государственный Технологический Институт (Технологический университет)

Кафедра Системного Анализа и Информационных Технологий

**Проект**

**Создание Telegram-чат-бота с использованием методов ИИ**

Выполнил: Коготков М.А.

Проверил:Мусаев А.А.

Санкт-Петербург

2024 г.

Разработать интерактивного Telegram-чат-бота, который будет использовать методы искусственного интеллекта для выполнения одной из следующих задач:

* Ответы на часто задаваемые вопросы (Результат матча по Дота 2)

1. Данный бот работает в области развлечений конкретно в области предсказания матчей по компьютерной игре.
2. Ключевые функции бота вывод списка всех героев, точности модели предсказаний и предсказания результата матча по введенным командам, игрокам и героям.
3. Зарегистрируйте бота в Telegram и получите токен API.

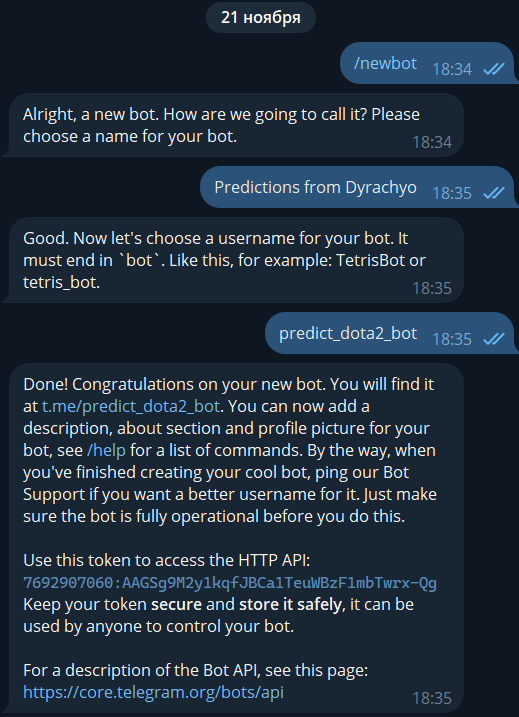


Рисунок 1 – регистрация бота и получение токена

1. Настройте базовую структуру бота с использованием Python и библиотеки python-telegram-bot (или другой).

Для реализации пользовательского интерфейса использовался телеграмм бот, со следующими функциями. (Рисунок 11) Пример работы бота на Рисунках 12-14. Для реализации бота использовались библиотеки telebot и spacy.

Изображение выглядит как текст, электроника, снимок экрана, число

Автоматически созданное описание

Рисунок 11 – функции бота

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 13 – пример выполнения функции вывод точности

Изображение выглядит как текст, Человеческое лицо, одежда, человек

Автоматически созданное описание

Рисунок 14 – пример предсказания результатов матча

Для реализации загружаем модель SpaCy для работы с русским языком. Вызываем экземпляр класса dota\_ai (реализующего модель для предсказания матчей игры Dota 2). Затем происходит обучение модели на данных с размером выборки n (ai.train(n)). Далее происходит получение списка доступных героев из ai.df\_hero. Создается словарь для хранения пользовательских данных. Затем происходит инициализация Telegram-бота с заданным токеном. (Рисунок 15)

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

Автоматически созданное описание

Рисунок 15 – реализация пользовательского интерфейса

Далее рассмотрим функцию обработки команды /start. При запуске бота создаётся клавиатура с кнопками: "Начать использование" — для перехода к функционалу бота. И "Помощь" — для отображения информации о функциях бота. (Рисунок 16)

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 16 - реализация функции обработки команды /start

Данная функция обрабатывает текстовые сообщения от пользователя для этого, текст приводится к нижнему регистру (lower) и обрабатывается моделью SpaCy. В обработке SpaCy разбивает предложение на токены и находит базовые формы слов (леммы). Затем если текст содержит слово "помощь", вызывается функция show\_help. (Рисунок 18) Эта функция отправляет пользователю информацию по использованию бота. Если текст содержит слово "начать", вызывается функция start\_usage. (Рисунок 19) Данная функция создает клавиатуру с действиями: "Вывод точности модели", "Предсказание результата матча" и " Список героев". Если текст содержит слово "точность", отправляется сообщение с точностью модели, вызвав метод accuracy\_test\_data(). В этом методе пользователь вводит данные для предсказания: происходит преобразование данных через функцию text\_parser.parse\_input\_text\_to\_data. Если эти данные успешно обрабатываются, то предсказывается победитель (ai.result). В противном случае пользователю отправляется сообщение об ошибке. Также при нахождении в тексте слова "Герой" выводит пользователю список доступных героев.Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 17 – функция обработки сообщений

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

Автоматически созданное описание

Рисунок 18 – функция show\_help

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 19 – функция start\_usage

В функции parse\_input\_text\_to\_data происходит обработка текста через SpaCy и инициализации словаря (Рисунок 20)

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

Автоматически созданное описание

Рисунок 20 - обработка текста и инициализация словаря

Далее в функции происходит заполнение словаря. (Рисунок 20) Входной текст предварительно разбивается на предложения с использованием spaCy. Каждый элемент в цикле for sent in doc.sents представляет отдельное предложение из текста. Если в текущем предложении встречается фраза "Команда Radiant", это означает, что оно содержит название команды Radiant. Название извлекается с помощью разбиения строки по символу ":". Берется часть строки после двоеточия, убираются лишние пробелы и точка в конце строки. Аналогично, для фразы "Команда Dire". После извлечения названия текущая команда устанавливается в Radiant или Dire для дальнейшей обработки. Если в предложении есть фраза "Игроки:", это указывает на перечисление игроков и героев. Список игроков и героев разбивается на элементы с помощью split(", "), чтобы каждая пара обрабатывалась отдельно. Для каждой пары проверяется наличие слова "на" (что подтверждает формат "Игрок на Герой"). Если пара корректна, она делится на имя игрока и имя героя с помощью split(" на ").

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

Автоматически созданное описание

Рисунок 20 – заполнение словаря

Сообщения от пользователя сохраняются в таблицу базы данных PostgreSQL с помощью функции