

# РАЗРАБОТКА АВТОМАТИЗИРОВАННОГО ПОМОЩНИКА В СОЦИАЛЬНЫХ СЕТЯХ ПО РАСПОЗНАВАНИЮ ЕСТЕСТВЕННОГО ЯЗЫКА

## DEVELOPING AN AUTOMATED SOCIAL MEDIA ASSISTANT FOR NATURAL LANGUAGE RECOGNITION

**K. Tagirov  
T. Tagirov  
L. Dmitriev**

*Summary.* An automated assistant capable of natural language recognition is becoming increasingly popular in social media. This paper discusses the development of a chatbot that uses machine learning techniques for natural language recognition. The authors describe the process of integrating the helper into a social network, as well as its functionality to improve user communication and service quality. Test results have shown the effectiveness of the developed automated social networking assistant.

*Keywords:* machine learning, chat-bot, natural language, speech recognition, social networks.

**Тагиров Кадир Межвединович**

Ассистент, аспирант,  
ФГБОУ ВО «Нижневартковский  
государственный университет»  
kadir.tagirov1997@gmail.com

**Тагиров Тагир Межвединович**

преподаватель,  
БУ «Нижневартковский строительный колледж»  
tagir.tagirov97vostan@gmail.com

**Дмитриев Леонид Романович**

БУ «Нижневартковский строительный колледж»  
Leoredzhi@ya.ru

*Аннотация.* Автоматизированный помощник, способный распознавать естественный язык, становится все более востребованным в социальных сетях. В данной статье рассматривается разработка чат-бота, который использует методы машинного обучения для распознавания естественного текста. Авторы описывают процесс интеграции помощника в социальную сеть, а также его функциональные возможности, которые позволяют улучшить коммуникацию пользователей и повысить качество обслуживания. Результаты тестирования показали эффективность разработанного автоматизированного помощника в социальных сетях.

*Ключевые слова:* машинное обучение, чат-бот, естественный язык, распознавание речи, социальные сети.

### Введение

Тема данной работы является актуальной, так как социальные сети стали незаменимой частью нашей жизни [3]. Сегодня мы используем социальные сети не только для общения, но и для познания новостей, поиска информации и много другого. Однако рост количества контента для пользователей с каждым днём всё растёт и становится труднее определять одну статью от другой и на этой почве появляется требование в идентификации одного текста от другого. На сегодняшний день развитие технологий достигает возможности создания искусственного интеллекта, который обучается и может делать многие вещи: управлять автомобилем, рисовать, писать статьи, песни. Машинное обучение, основанное ещё в конце 20 века [4], на сегодняшний день становится самой популярной и актуальной задачей в сфере технологий.

Для того чтобы развивать такой искусственный интеллект его удобно размещать в публичный доступ — в социальной сети, для того чтобы пользователи смогли обучать его и наполнять его память данными для обработки. Одной из самых популярных социальных сетей

является Telegram (Телеграм) — насчитывающий около 50 миллионов пользователей [7].

Целью данной работы является автоматизированная и доступная каждому пользователю система для анализа текста и искусственного языка,

Задачи для достижения данной цели:

- разработка программа по распознаванию естественного языка и текстов;
- разработка бота-помощника, в основе которого будет стоять данная программа;
- запуск и оптимизация приложения.

Для создания такого помощника понадобится знания машинного обучения, умение писать код на языке программирования Python для того, чтобы иметь возможность создать как внутренний мозг бота, так и его внешнюю составляющую.

Объектом изучения данной работы является использование искусственного интеллекта в социальных сетях, на примере разработки чат-бота в Telegram.

Предметом статьи является разработка программы по распознаванию естественного языка и текстов на язы-

ке программирования Python, а также создание бота-помощника для Telegram на основе данной программы.

### Описание предметной области

Анализ систем распознавания и определения естественного языка показал, что тексты имеют разные стили, а необходимость их распознавания растёт [2].

Машинное обучение — это подраздел искусственного интеллекта, который позволяет компьютерам учиться на основе данных и выполненных задач. Вместо жесткого программирования, где программа закодирована до последнего действия, в машинном обучении компьютер учится находить закономерности в данных и создавать свои правила на основе обработки и анализа информации.

Процесс начинается с обучающего набора данных, который содержит примеры, на которых будет тренироваться модель. Эти данные используются для построения модели, которая может классифицировать новые данные или делать прогнозы. Как только модель создана, она может быть использована для анализа новых данных и чтобы делать прогнозы.

Чат-боты — это программы, которые имитируют человеческое общение в мессенджерах и социальных сетях [1]. Они могут быть использованы для различных задач, начиная от автоматизации бизнес-процессов до общения с клиентами.

Telegram — это мессенджер, который пользуется популярностью по всему миру. Данная социальная сеть обеспечивает высокую безопасность и конфиденциальность общения. Telegram имеет удобный и интуитивно понятный интерфейс, который позволяет быстро и легко находить нужные функции и настройки, что делает его привлекательным для широкой аудитории.

Язык программирования Python хорошо подходит для решения всех этих задач: для создания чат-бота в Telegram и искусственного интеллекта. Данный язык предоставляет широкие возможности для создания искусственного интеллекта, благодаря мощным библиотекам и фреймворкам, таким как TensorFlow, Keras, PyTorch и другим. Python имеет большое количество библиотек и инструментов, позволяющих значительно упростить процесс разработки. Например, библиотека NLTK предоставляет широкие возможности для обработки естественного языка, а библиотека Scikit-learn позволяет создавать мощные модели машинного обучения. Кроме того, Python поддерживает множество инструментов для автоматического тестирования и непрерывной интеграции, что позволяет значительно упростить процесс разработки и обеспечить высокое качество продукта [5].

### Библиотеки используемые для разработки приложения

**Natural Language Toolkit (NLTK)** — это библиотека Python для обработки естественного языка. Она содержит множество инструментов и ресурсов для работы с текстовыми данными, такими как токенизация, стемминг, лемматизация, разметка частей речи и многое другое. NLTK имеет обширную документацию и активное сообщество разработчиков, что делает ее одной из лучших библиотек для обработки естественного языка.

**TensorFlow** — это библиотека для машинного обучения, которая широко используется для разработки моделей глубокого обучения. TensorFlow также имеет инструменты для работы с естественным языком, такие как TensorFlow Text, которые предоставляют высокоуровневые API для обработки текстовых данных. Эта библиотека может быть хорошим выбором для разработки более сложных алгоритмов и моделей глубокого обучения для обработки естественного языка.

Так же будет использована библиотека **PyTorch**, которая также является мощным инструментом для работы с нейронными сетями и глубоким обучением. Она обладает более простым и интуитивно понятным интерфейсом, чем TensorFlow, что может упростить разработку и отладку нейронных сетей.

В качестве библиотеки для работы с чат-ботом в социальной сети Telegram, был выбран **python-telegram-bot**. Это популярная библиотека для создания Telegram-ботов, которая предоставляет широкий спектр функций для работы с API Telegram.

### Разработка телеграм бота

Для работы с синтаксическим анализом в NLTK необходимо создать грамматику, которая определяет правила синтаксиса для конкретного языка. Грамматика в NLTK представляется в виде контекстно-свободной грамматики (CFG).

Пример грамматики в NLTK для русского языка листинг 1.

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.parse import DependencyGraph,
DependencyEvaluator
from nltk.corpus import udhr

# загружаем корпус зависимостей
corpus = udhr.parsed_sents(language='rus')

# обучаем грамматику на основе корпуса
grammar = nltk.grammar.DependencyGrammar.from_
corpus(corpus)
```

```

parser = nltk.parse.ProjectiveDependencyParser(grammar)

# тестовое предложение
sentence = 'Мама мыла раму'

# токенизация предложения
tokens = word_tokenize(sentence, language='russian')

# синтаксический анализ
parsed_sent = next(parser.parse(tokens))

# вывод дерева зависимостей
graph = DependencyGraph(parsed_sent.to_conll(10))
print(graph.tree())

```

Листинг 1.

Здесь загружается корпус зависимостей для русского языка из NLTK, обучается грамматика на основе этого корпуса и создаётся функция. Затем анализируется тестовое предложение «Мама мыла раму», токенизируется и производится синтаксический анализ. Результат выводится в виде дерева зависимостей.

Обработка текстовой информации является важной частью многих приложений, таких как анализ текстов, машинное обучение на текстовых данных, обработка естественного языка и многое другое.

Класс `TextProcessor` будет представлять основной функционал для обработки текстовых данных. Он будет содержать методы для токенизации, лемматизации, стемминга, синтаксического анализа и других операций над текстом.

Для разработки чат-бота нужно будет определить класс `ChatBot`. Этот класс будет отвечать за логику обработки запросов пользователя и формирование ответов.

Класс `ChatBot` будет иметь следующие методы:

**`__init__`**: конструктор класса, который будет инициализировать переменные для работы чат-бота.

**`get_response`**: метод, который будет принимать текстовый запрос пользователя и возвращать ответ, сформированный чат-ботом.

Реализация механизма обработки запросов пользователей в социальной сети включает в себя разработку алгоритмов для обработки запросов и определения ответа на запрос пользователя [6].

Для получения сообщений от пользователей необходимо создать обработчик сообщений (`telegram.ext.MessageHandler`) и зарегистрировать его в экземпляре

```

1  import telegram
2  from telegram.ext import Updater, MessageHandler, Filters
3  TOKEN = '5892394227:AAEKn1cxZK-jхupKe1K1rcZX_9SWImir6Ik'
4
5  bot = telegram.Bot(TOKEN)
6
7
8
9  updater = Updater(TOKEN, use_context=True)
10 dispatcher = updater.dispatcher
11
12 def handle_text(update, context):
13     text = update.message.text
14     # обработка текстового сообщения и формирование ответа
15     # ...
16     bot.send_message(chat_id=update.message.chat_id, text=response_text)
17
18 text_handler = MessageHandler(Filters.text, handle_text)
19 dispatcher.add_handler(text_handler)
20 |

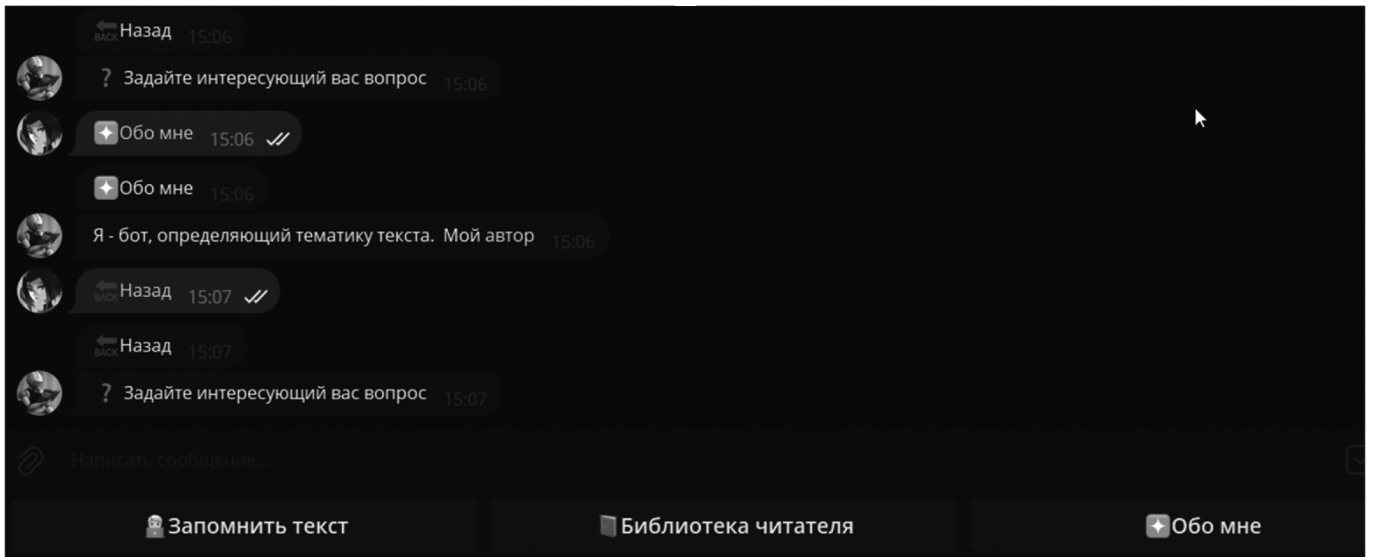
```

Источник: составлено автором

Рис. 1. Созданный обработчик сообщений

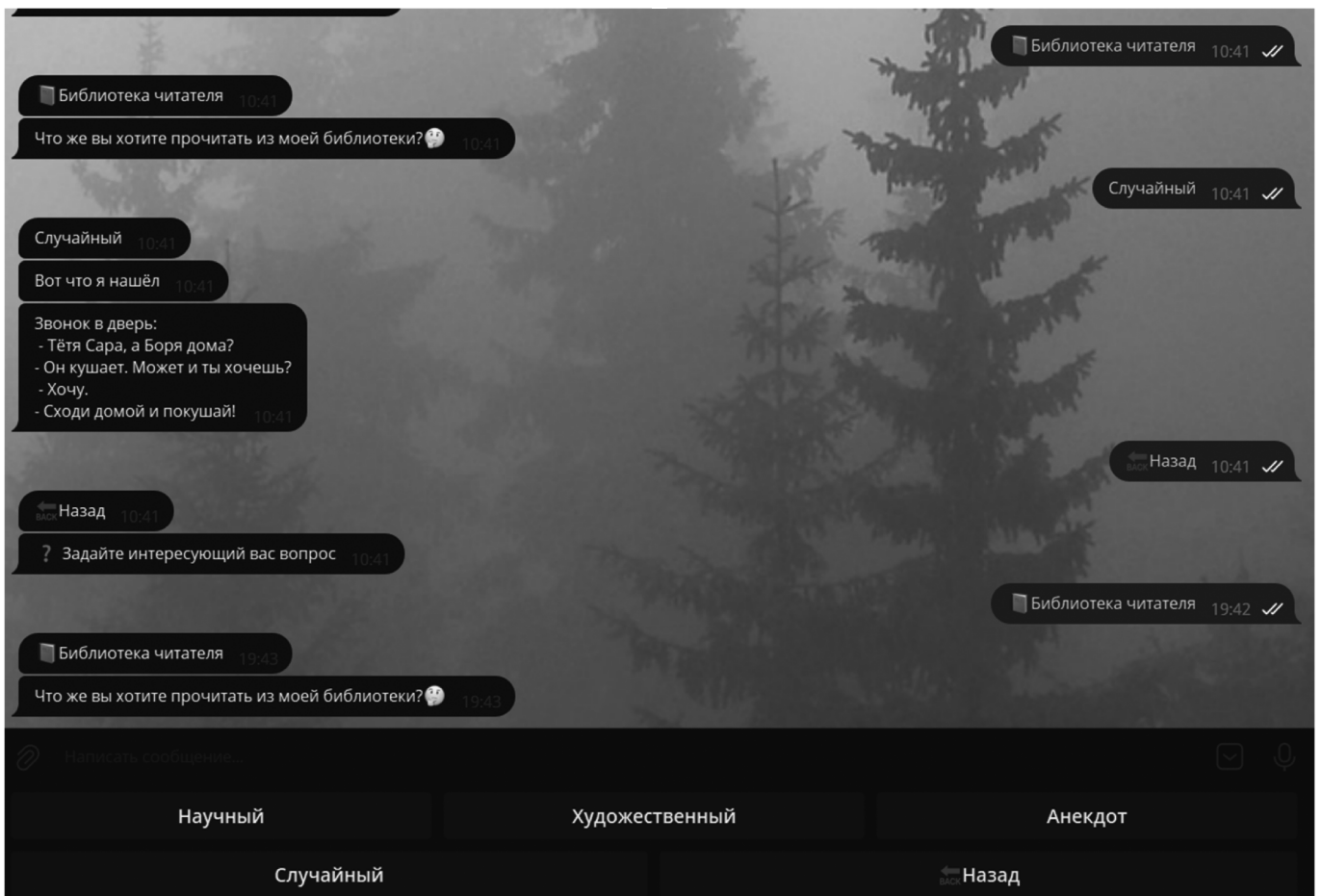
класса telegram.ext.Updater. Обработчик сообщений может быть настроен на обработку сообщений определенного типа (текстовые сообщения, изображения, видео

и т.д.). В данном случае код будет обрабатывать только текстовые сообщения. Созданный обработчик сообщений изображен на рисунке 1.



Источник: составлено автором

Рис. 2. Работа телеграм-бота



Источник: составлено автором

Рис. 3. Работа чат-бота и меню выбора текста из базы данных

После обработки текстовых сообщений чат-бот должен выполнить действия, соответствующие запросу пользователя. Например, если пользователь запросил показать текст статьи, программа бота должна запросить текст похожий на статью у базы данных и вернуть её пользователю.

Работа кнопок и самого чат-бота изображена на рисунках 2 и 3.

В результате нашей работы был разработан полноценный продукт, который может быть использован для автоматизации работы с текстовой информацией в социальной сети. Помощник на основе распознавания естественного языка может существенно ускорить процесс обработки информации и улучшить качество подаваемой информации на основе машинного обучения.

---

#### ЛИТЕРАТУРА

1. BotFather: как создать бота в Телеграм (инструкция на русском) [Электронный ресурс] / URL: <https://mrtext.ru/botfather-kak-sozdat-bota-v-telegram-instrukciya-na-russkom.html>
2. Арзуманова А.Г., Майоров Р.Ю. Применение методов обработки естественного языка в задачах мониторинга социальных медиа // Научно-технический вестник информационных технологий, механики и оптики. — 2018. — Т. 18. — №. 5. — С. 899–904.
3. Горбачев Максим Андреевич, Хитров Николай Олегович Проектирование веб приложения для интеграции сервисов с мессенджерами // StudNet. 2021. №7. [Электронный источник] / URL: <https://cyberleninka.ru/article/n/proektirovanie-veb-prilozheniya-dlya-integratsii-servisov-s-messenzherami> (дата обращения: 02.05.2023).
4. Качков Михаил Сергеевич Создание нейронной сети для решения различных прикладных задач // Известия ТулГУ. Технические науки. 2023. №2. [Электронный источник] / URL: <https://cyberleninka.ru/article/n/sozдание-neyronnoy-seti-dlya-resheniya-razlichnyh-prikladnyh-zadach> (дата обращения: 02.05.2023).
5. Китаева Ольга Игоревна Интеллектуальный анализ образовательных данных учебной дисциплины с использованием программы orange // Информационные и математические технологии в науке и управлении. 2023. №1 (29). [Электронный источник] / URL: <https://cyberleninka.ru/article/n/intellektualnyy-analiz-obrazovatelnyh-dannyh-uchebnoy-distipliny-s-ispolzovaniem-programmy-orange> (дата обращения: 02.05.2023).
6. Лобов А.В., Шевлякова Т.В. Использование методов обработки естественного языка для автоматизации ответов на социальных сетях // Вестник Иркутского государственного технического университета. — 2020. — Т. 24. — №. 3. — С. 25–31.
7. Низомутдинов Б.А., Углова А.Б., Богдановская И.М. Автоматизированный кластерный анализ коммуникативных стратегий образовательных telegram-каналов // Научно-технический вестник информационных технологий, механики и оптики. 2023. № 1. [Электронный источник] / URL: <https://cyberleninka.ru/article/n/avtomatizirovannyi-klasternyy-analiz-kommunikativnyh-strategiy-obrazovatelnyh-telegram-kanalov> (дата обращения: 02.05.2023).

---

© Тагиров Кадир Межвединович ([kadir.tagirov1997@gmail.com](mailto:kadir.tagirov1997@gmail.com)); Тагиров Тагир Межвединович ([tagir.tagirov97bosman@gmail.com](mailto:tagir.tagirov97bosman@gmail.com));  
Дмитриев Леонид Романович ([Leoredzhi@ya.ru](mailto:Leoredzhi@ya.ru))

Журнал «Современная наука: актуальные проблемы теории и практики»