УДК 004

РЕАЛИЗАЦИЯ СИСТЕМЫ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА С ИСПОЛЬЗОВАНИЕМ МАШИННОГО ОБУЧЕНИЯ И NLP

Золотарёв Н.С., студент гр. ИТб-202, IV курс
Научный руководитель: Протодьяконов А.В, к.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева,
Кемерово

Данная статья посвящена разработке и реализации системы обработки естественного языка Natural Language Processing, NLP с использованием машинного обучения. Основной целью проекта является создание виртуального агента, способного анализировать текстовые запросы пользователей и предоставлять соответствующие ответы. Кроме того, мы намерены подробно рассмотреть важные компоненты кода, а также рассмотреть возможные варианты применения методов машинного обучения в контексте данной модели.

Целью данного проекта является разработка и реализация системы обработки естественного языка, которая способна анализировать текстовые запросы пользователей и предоставлять релевантные ответы. Примеры целей включают улучшение обслуживания клиентов, автоматизацию процессов обработки запросов и повышение эффективности коммуникации.

Предполагаемый функционал системы.

Для достижения поставленных целей система должна обладать следующим функционалом:

- Анализ текстовых запросов с использованием NLP-модели.
- Предоставление релевантных ответов на запросы.
- Интеграция с пользовательским интерфейсом для удобства взаимодействия с агентом.

Функциональные требования включают в себя:

Обработка текстовых запросов с выделением ключевых фраз и смысла.

Предоставление ответов, соответствующих запросу пользователя.

Возможность обучения системы на новых данных для улучшения качества ответов.

Пользователи системы – сотрудники, требующие поддержки в решении технических вопросов.

В рамках данного проекта исследуется технология обработки естественного языка NLP с использованием методов машинного обучения. Основные этапы работы с NLP включают в себя:

- Токенизацию: разделение текста на отдельные слова или фразы.
- Лемматизацию: приведение слов к их базовой форме.
- Анализ синтаксиса: определение структуры предложения.

Важные части кода включают:

- Использование библиотеки spaCy для обработки текста на естественном языке.

```
import spacy
#Загрузка модели spacy для русского языка
nlp = spacy.load("ru_core_news_md")
python - m spacy download ru_core_news_md
```

- Создание Flask API для взаимодействия с пользовательским интерфейсом.

```
from flask import Flask, request, jsonify
from nlp_script import process_text
```

- Интеграция пользовательского интерфейса WinForms с бэкэндом Flask API.

```
using System.Text.Json;
using System.Net.Http;
```

- Реализация методов на языке C#, для получения ответа от модели NLP.

```
    //обращаемся к http, которое открывается на localhost от
FlaskAPI
    using (HttpClient client = new HttpClient())
    {
        HttpResponseMessage response = await client.PostA-
sync(PythonApiUrl, new StringContent(userInput));

        if (response.IsSuccessStatusCode)
        {
            string responseContent = await response.Con-
tent.ReadAsStringAsync();

            //Декодируем последовательности Unicode
```

```
            string decodedResponse = DecodeUnicodeEscapeSe-
quences(responseContent);
            // Добавляем запрос и ответ в текстовое поле
winform на новых строках
                                txtDialog.AppendText($"{User}:
{userInput}{Environment.NewLine}");
                               txtDialog.AppendText($"{Agent}:
{decodedResponse}{Environment.NewLine}{Environment.NewLine}")
;
        }

    }
```

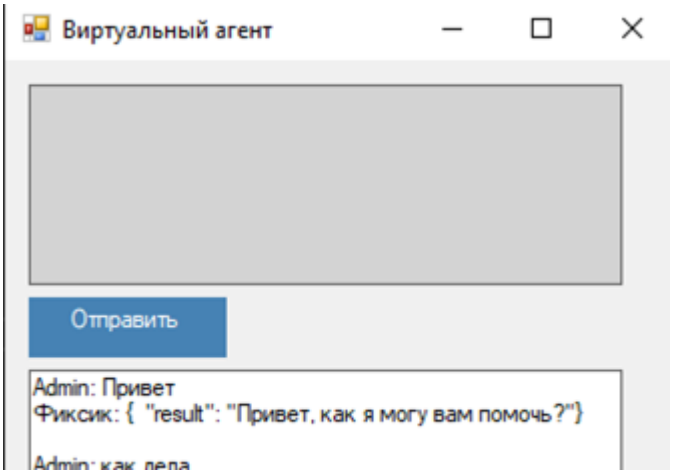

Рисунок 1. Пример использования 1

Рисунок 2. Пример использования 2

Возможные реализации машинного обучения.

Машинное обучение предоставляет множество возможностей для улучшения системы обработки естественного языка. Давайте рассмотрим несколько примеров реализации их работы:

1) Обучение на размеченных данных:

Этот метод предполагает использование большого объема размеченных данных (текстовых запросов и соответствующих ответов), чтобы обучить модель на основе существующих примеров.

Например, можно обучить модель на предоставленных исторических данных запросов и ответов, чтобы она могла адаптироваться к конкретным запросам пользователей и предоставлять более точные ответы в будущем.

2) Использование алгоритмов классификации:

Этот подход позволяет определить тип запроса пользователя и соответствующий ему ответ.

Например, можно использовать алгоритм классификации, такой как Random Forest или Support Vector Machines, чтобы автоматически определять, является ли запрос пользователя запросом о помощи, запросом информации или чем-то еще, и в соответствии с этим предоставлять соответствующий ответ.

3) Применение алгоритмов кластеризации:

Этот метод позволяет группировать запросы по схожим темам или контекстам, что обеспечивает более эффективную обработку.

Например, можно использовать алгоритм кластеризации, такой как K-means, чтобы автоматически группировать запросы, связанные с определенной темой или проблемой, в один кластер, что позволит системе обрабатывать их более эффективно и предоставлять более точные ответы.

4) Рекуррентные нейронные сети для последовательной обработки текста:

Этот метод позволяет моделировать последовательность слов или фраз в тексте и выявлять зависимости между ними.

Например, можно использовать рекуррентные нейронные сети (RNN) или Long Short-Term Memory (LSTM) сети для обработки последовательности слов в тексте запроса пользователя, что позволит модели лучше понимать контекст и смысл запроса. И на основе RAG-модели, совместно с OpenLLM построить полноценную модель на доступном токене.

Каждая из этих реализаций машинного обучения предоставляет уникальные возможности для улучшения системы обработки естественного языка. Их успешная интеграция может значительно повысить качество обслуживания клиентов и эффективность коммуникации в рамках предприятия.

Данная система демонстрирует эффективное использование методов машинного обучения и NLP для обработки текстовых данных и предоставления пользователю релевантных ответов. Основываясь на реализации и тестировании системы, можно сделать вывод о ее потенциале для улучшения процессов обслуживания клиентов и оптимизации коммуникации в рамках предприятия.

Дальнейшее развитие проекта может включать в себя расширение функционала, улучшение качества обработки текста и добавление новых методов машинного обучения для улучшения производительности и качества обслуживания.

Список литературы:

1. Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных. М.: ВШЭ, 2017.
2. Документация SpaCy. [Электронный ресурс] URL: https://spacy.io/usage/

3. Документация API Flask. [Электронный ресурс] URL: https://flask.palletsprojects.com/en/latest/api/
4. Юрий Васильев. Обработка естественного языка. Python и spaCy на практике. — СПб.: Питер, 2021. — 256 с.: ил.