МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ШКОЛА КОМПЬЮТЕРНЫХ НАУК

Кафедра программного обеспечения

ОТЧЕТ  
О РАЗРАБОТКЕ ПРОЕКТА ПО ДИСЦИПЛИНЕ  
«Интеллектуальные системы поддержки принятия решений»

ТЕМА ПРОЕКТА

«Разработка чат бота для формирования индивидуальных инструкций (дорожных карт) с использованием подходов обработки естественного языка для клиентов МФЦ Тюменской области»

| Выполнил  обучающийся 4 курса,  МОиАИС-20.03 группы | (подпись) | Шеремет Арсений Андреевич |
| --- | --- | --- |
| Выполнил  обучающийся 4 курса,  МОиАИС-20.03 группы | (подпись) | Устинов Илья Александрович |
| Научный руководитель проекта  Доцент, к.т.н. | (подпись) | Плотоненко Юрий Анатольевич |

«Тюмень 2024»

# ОГЛАВЛЕНИЕ

# 

[**ОГЛАВЛЕНИЕ 2**](#_458jj9uqsx1z)

[**ВВЕДЕНИЕ 3**](#_ibernp1uhq4x)

[**ГЛАВА 1. ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА 6**](#_yamtzfp99st0)

[**ГЛАВА 2. РАЗРАБОТКА ПРОЕКТА 10**](#_894lwd5cu88q)

[**2.1. ТРЕБОВАНИЯ К ПРОЕКТУ И УЧИТЫВАЕМЫЕ ОГРАНИЧЕНИЯ. 10**](#_msat5qak2uju)

[2.1.1. ТРЕБОВАНИЯ К ПРОЕКТУ 10](#_kqun5u2db04x)

[2.1.2. УЧИТЫВАЕМЫЕ ОГРАНИЧЕНИЯ 13](#_rtiuc1xzt79z)

[2.2. ОПРЕДЕЛЕНИЕ ЗАДАЧ ДЛЯ РЕАЛИЗАЦИИ ПРОЕКТА 16](#_3zm1n1346t9)

[**2.3. СОБСТВЕННЫЕ ТЕОРЕТИЧЕСКИЕ РАЗРАБОТКИ ПРИ РЕАЛИЗАЦИИ ПРОЕКТА 19**](#_pben2pgvnjwh)

[2.3.1. ИСПОЛЬЗОВАНИЕ BERT В ПРОЕКТЕ 19](#_8rz08418egg9)

[2.3.2. ИСПОЛЬЗОВАНИЕ ГЕНЕРАТИВНЫХ МОДЕЛЕЙ В ПРОЕКТЕ 22](#_1y60hp7h16de)

[2.3.3. СОБСТВЕННЫЕ ПРАКТИЧЕСКИЕ РЕШЕНИЯ, РЕАЛИЗОВАННЫЕ В ПРОЕКТЕ. 26](#_aypbaj9o8rp4)

[**ГЛАВА 3. ДЕТАЛИ РЕАЛИЗАЦИИ ПРОЕКТА 28**](#_p80bohpeut7j)

[3.1. ОПИСАНИЕ АРХИТЕКТУРЫ 28](#_lyy5ru20g7kr)

[3.2. ОПИСАНИЕ ПРОЦЕССА ОБРАБОТКИ ЗАПРОСОВ 30](#_2m0lneokbtia)

[**ГЛАВА 4. ФУНКЦИОНАЛ И ТЕСТИРОВАНИЕ ПРОЕКТА 32**](#_qopkvuh5b1u8)

[4.1. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ПРОЕКТА 32](#_cotk257s1yy9)

[4.2. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ПРОЕКТА. 36](#_90yg8prrc5qb)

[4.3.ТЕСТИРОВАНИЕ ПРОЕКТА 38](#_d0dmfcvz6qa5)

[**ЗАКЛЮЧЕНИЕ 40**](#_h06s4qmgmd9u)

[**СПИСОК ЛИТЕРАТУРЫ 42**](#_skx9tok29jkw)

[**ПРИЛОЖЕНИЕ 1 43**](#_fcg07dt985)

[**ПРИЛОЖЕНИЕ**](#_fcg07dt985) [**2 44**](#_yhn2cahq0b3g)

[**ПРИЛОЖЕНИЕ**](#_fcg07dt985) [**3 45**](#_k5d02880w4wm)

[**ПРИЛОЖЕНИЕ**](#_fcg07dt985) [**4 46**](#_ilnrkv4em2jr)

[**ПРИЛОЖЕНИЕ**](#_fcg07dt985) [**5**](#_e72bdfjvun1k) **47**

# ВВЕДЕНИЕ

Современные многофункциональные центры (МФЦ) сталкиваются с необходимостью улучшения взаимодействия с гражданами, которые обращаются за получением услуг. Одной из ключевых проблем является отсутствие интерфейса, позволяющего пользователям быстро и эффективно получать информацию о необходимом наборе документов для выполнения конкретных запросов. Эта проблема особенно актуальна, учитывая разнообразие пользователей и их запросов, которые могут существенно различаться по содержанию и форме.

**Проблематика** Текущий процесс взаимодействия граждан с МФЦ часто требует нескольких посещений, что создает неудобства для пользователей и увеличивает нагрузку на офисы.  
Основные этапы традиционного взаимодействия:

1. Посещение офиса для записи и получения информации о необходимых документах.
2. Повторное посещение для подачи документов (часто с ошибками из-за неправильной информации или неполного набора документов).
3. Финальное посещение для получения результата.

Такая схема имеет ряд недостатков:

* предоставление неверной или неполной информации со стороны сотрудников,
* увеличение количества визитов из-за ошибок,
* рост очередей и нагрузки на офисы МФЦ,

**Идея проекта** Предлагаемое решение заключается в создании чат-бота, который позволит гражданам МФЦ сократить количество визитов и повысить качество подготовки к посещению. С помощью бота пользователи смогут:

* узнать, какие документы необходимы для конкретной услуги,
* заранее подготовить полный пакет документов,
* сократить число посещений до двух: подача полного комплекта документов и получение результата.

Реализация бота предполагает использование документоориентированной базы данных, где каждый документ содержит полную и неделимую информацию. Это обеспечивает точность предоставляемых данных и минимизирует вероятность ошибок.

Данный подход позволяет сократить число визитов, снизить нагрузку на офисы МФЦ за счет оптимизации рабочих процессов сотрудников, а также повысить удовлетворенность граждан.

**Актуальность** Заказчиком проекта выступает заместитель директора МФЦ Тюменской области Дмитрий Сальников. Основная задача, поставленная заказчиком, заключается в создании системы, которая поможет пользователям заранее подготовить набор документов, необходимых для получения услуги. Дополнительно заказчик выразил пожелание реализовать возможность предварительной записи на конкретные услуги с уже сформированным пакетом документов.

Важно отметить, что строгих требований к проекту не было выдвинуто, а сам проект рассматривается как отправная точка для дальнейшего развития системы. Ключевая цель заключается в создании прототипа, который возможно сократит количество визитов, минимизирует ошибки на этапе подготовки документов и оптимизирует взаимодействие граждан с МФЦ.

# ГЛАВА 1. ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

Методы обработки естественного языка (NLP) :

* **Токенизация и нормализация текста**: разделение текста запроса клиента на токены (слова или предложения), приведение текста к стандартной форме (например, приведение к нижнему регистру, удаление пунктуации). [Минцер Т., Гольдберг Й., Маннинг К, с. 80]

**Вывод**: Эти методы позволяют снизить шум в данных и повысить точность классификации запросов.

* **Лемматизация или стемминг**: приведение слов к их исходной форме для лучшего понимания запроса.

**Вывод**: Приведение слов к базовой форме помогает устранить разницу между склонениями/формами слова при классификации.

* **Классификация запросов**: определение типа запроса с помощью моделей машинного обучения, например, классификаторов на основе трансформеров (BERT) или более простых моделей, таких как SVM, Naive Bayes или LDA (Latent Dirichlet Allocation). [Минцер Т., Гольдберг Й., Маннинг К, с. 91]

**Вывод:** Использование алгоритмов для анализа ключевых аспектов запроса, чтобы обеспечить точное соответствие между запросом и формируемой инструкцией.

Модели машинного обучения:

* **Модели трансформеров (Transformers)**: трансформеры, такие как BERT [Devlin J., Chang M.-W., Lee K., Toutanova K]., GPT [Vaswani A., Shazeer N., Parmar N], T5, можно использовать для анализа запроса и генерации/экстракции текстовых инструкций на основе контекста запроса.

**Вывод:** Использование трансформеров оправдано, если проект требует обработки сложных текстов с учетом контекста.

* **LDA (Latent Dirichlet Allocation)**: LDA используется для автоматического выявления тем в больших коллекциях текстов, таких как новостные статьи, научные публикации, социальные сети и т.д. [Mikolov T., Chen K., Corrado G., Dean J]
* **Doc2Vec**: полученные векторы документов использовать для различных задач, таких как классификация документов, кластеризация, поиск схожих документов и т.д.

**Вывод:** Методы Doc2Vec и LDA подходят для анализа больших массивов текстов.

* **GloVe (Global Vectors for Word Representation)**: использует глобальные статистические данные о частоте встречаемости слов и их сочетаний в корпусе текстов. Это позволяет учитывать как локальные, так и глобальные свойства слов.
* **LSi (Latent Semantic Indexing)**: Основная идея LSI заключается в том, чтобы преобразовать высокоразмерное пространство терминов (слов) в низкоразмерное пространство концепций, что позволяет лучше понять семантические связи между документами и терминами.

**Вывод:** GloVe и LSi обеспечивают компактное и информативное представление данных.

Обработка данных и база данных:

* **База данных**: использование существующей базы данных с типовыми сценариями, инструкциями, описанием услуг или продуктов, которые могут быть использованы в дорожных картах.

**Вывод:** Готовая база данных обеспечивает основу для генерации инструкций и сокращает время обработки запроса.

* **Системы поиска информации (IR)**: использование методов поиска релевантной информации по запросам, возможно с использованием векторных представлений документов (например, через ElasticSearch или Faiss). [Manning C., Raghavan P., Schütze H с. 247]

**Вывод:** Эффективный поиск необходим для масштабируемости системы.

Интерфейс (Frontend):

* **Telegram Bot API**: для взаимодействия с пользователями через Telegram. Это позволит отправлять и получать сообщения, обрабатывать нажатия на кнопки, команды и другие интерактивные элементы. Для создания логики бота будет использоваться библиотека, такая как python-telegram-bot или aiogram.
* **Интерактивные элементы**: кнопки, меню и другие элементы для взаимодействия с пользователем. Например, клиент может выбирать услуги через кнопки.
* **Webhook**: для связи с backend (FastAPI) может использоваться webhook (Telegram будет отправлять запросы на API backend-а).

**Вывод:** Использование Telegram обеспечивает доступность сервиса для пользователей

Backend (API):

* **FastAPI**: для создания высокопроизводительного и масштабируемого backend API. FastAPI отлично поддерживает асинхронные запросы, что будет полезно для обработки множества взаимодействий с ботом в реальном времени.
* **Обработка запросов от Telegram**: FastAPI будет получать и обрабатывать данные от Telegram через webhook. API будет также принимать пользовательские запросы и передавать их в модели обработки естественного языка (NLP) для генерации ответов. [8, Handling HTTP requests and responses]
* **Маршруты и эндпоинты**: создание RESTful API для работы с разными типами запросов бота (получение команд, генерация инструкций, получение статуса и т.д.). [8, Implementing path and query parameters]

**Вывод:** FastAPI гарантирует масштабируемость сервиса, что важно при увеличении нагрузки.

# ГЛАВА 2. РАЗРАБОТКА ПРОЕКТА

## 2.1. ТРЕБОВАНИЯ К ПРОЕКТУ И УЧИТЫВАЕМЫЕ ОГРАНИЧЕНИЯ.

### 2.1.1. ТРЕБОВАНИЯ К ПРОЕКТУ

1. **Интерфейс построенный на основе чат-бота**

Описание: Чат-бот должен иметь следующий набор кнопок для реализации соответствующего функционала: Задать вопрос, оставить отзыв. Также на интерфейсе должны выводится доуточнения вопросов, если таковые будут иметься.

Результат: Интерфейс должен давать доступ к данным из базы данных МФЦ. Интерфейс должен уметь задать уточняющие вопросы, позволять оставлять обратную связь.

1. **Обработка пользовательских запросов.**

Описание: Чат-бот должен принимать и предобрабатывать запросы пользователей для последующего анализа. Это включает в себя распознавание естественного языка, понимание контекста.

Результат: Пользователи должны получать быстрые и точные ответы на свои запросы, что повысит эффективность взаимодействия с МФЦ и улучшит общий пользовательский опыт. Это также снижает нагрузку на операторов.

1. **Составление списка необходимых документов по запросам пользователя.**

Описание: Получение необходимых данных, которые соответствуют запросу пользователя. Список должен быть полным, чтобы минимизировать количество повторных обращений.

Результат: Пользователи должны получать полный и точный список необходимых документов по своим запросам, что снижает количество повторных обращений в МФЦ. Это также способствует снижению ошибок от человеческого фактора.

1. **Уточнение вопросов на основе пользовательских вопросов.**

Описание: Чат-бот должен уметь задавать уточняющие вопросы на основе запросов пользователей, чтобы лучше понять их потребности и предоставить более точную информацию.

Результат: Чат-бот должен задавать уточняющие вопросы, что позволит лучше понять потребности пользователей и предоставить более точную и релевантную информацию. Это позволит снизить количество ошибок и повысит удовлетворенность пользователей, улучшая общий опыт взаимодействия с МФЦ.

1. **Возможность встраивания в текущую архитектуру МФЦ.**

Описание: Чат-бот должен быть совместим с текущей архитектурой МФЦ и интегрироваться с существующими системами и базой знаний.

Результат: Чат-бот должен быть встроен в текущую архитектуру МФЦ, что обеспечит минимальные затраты на интеграцию. Это также позволит использовать существующие данные и ресурсы, улучшая общую эффективность и производительность системы.

1. **Фильтрация трафика.**

Описание: Система должна ограничивать пользователя в отправке по несколько запросов в секунду.

Результат: Система будет ограничивать пользователей, что обеспечит стабильную работу. Это также помогает предотвратить перегрузку серверов и поддерживать высокую производительность чат-бота.

1. **Сбор обратной связи.**

Описание: Чат-бот должен предоставлять пользователям возможность оставлять отзывы, предложения по улучшению работы. Это включает в себя функции для оценки качества предоставленной информации, а также возможность оставить текстовые комментарии.

Результат: Сбор обратной связи позволит постоянно улучшать и адаптировать чат-бот под потребности пользователей. Это способствует повышению качества использования чат-бота пользователями и улучшению общего опыта взаимодействия с МФЦ.

### 2.1.2. УЧИТЫВАЕМЫЕ ОГРАНИЧЕНИЯ

Ограничения для BERT:

1. **Размер модели:**
   * **BERT Base** (~110M параметров) требует около **4-6 ГБ VRAM** для обработки последовательностей длиной до 512 токенов.
   * **BERT Large** (~340M параметров) требует **12-16 ГБ VRAM**, особенно для задач fine-tuning.

Все ограничения на VRAM указаны в таблице 1.

1. **Длина последовательности:**
   * Увеличение длины входной последовательности существенно увеличивает потребление VRAM. Например, увеличение длины с 512 до 1024 токенов удваивает потребление памяти.
2. **Размер батчей:**
   * Большие размеры батчей требуют больше памяти. Для обучения с батчами >16 потребуется свыше **8 ГБ VRAM** (для BERT Base).
3. **Режим использования:**
   * **Fine-tuning:** Требует больше памяти, так как сохраняются промежуточные градиенты.
   * **Инференс:** Занимает меньше памяти, так как не нужно хранить градиенты.

Ограничения для генеративных моделей (GPT):

1. **Размер модели:**
   * **GPT-2 Small (124M параметров):** Требует ~4-5 ГБ VRAM для инференса.
   * **GPT-2 Medium (355M параметров):** Требует ~8-10 ГБ VRAM.
   * **GPT-3-RU Medium (345M параметров):** Требует ~8-10 ГБ VRAM
2. **Длина последовательности:**
   * Длина последовательности (context window) напрямую влияет на VRAM. GPT-3, например, имеет контекстное окно до 2048 токенов, что увеличивает требования к памяти при генерации длинных текстов.
3. **Батчи и параллелизм:**
   * Генерация текста в батчах или параллельное обслуживание нескольких запросов требует значительного увеличения VRAM.
4. **Оптимизация:**
   * Модели могут быть оптимизированы с помощью квантования (например, до INT8) или разделения между несколькими GPU, что уменьшает потребление памяти, но может снижать точность.

Таблица 1

Примерные ограничения VRAM

| **Модель** | **Количество параметров** | **VRAM для инференса** | **VRAM для обучения** |
| --- | --- | --- | --- |
| BERT Base | 117 млн | 4-6 ГБ | 6-8 ГБ |
| BERT Large | 340 млн | 12-16 ГБ | 16-24 ГБ |
| GPT-2 Small | 124 млн | 4-5 ГБ | 6-8 ГБ |
| GPT-2 Medium | 355 млн | 8-10 ГБ | 12-16 ГБ |
| rugpt3small\_based\_on\_gpt2 | 118 млн | 4-5 ГБ | 6-8 ГБ |
| rugpt3medium\_based\_on\_gpt2 | 345 млн | 8-10 ГБ | 12-16 ГБ |

## 2.2. ОПРЕДЕЛЕНИЕ ЗАДАЧ ДЛЯ РЕАЛИЗАЦИИ ПРОЕКТА

Для выполнения проекта был выделен список задач, который представлен в таблице 2.

Таблица 2

Описание задач

| Задача | Ответственный | Сроки выполнения | Описание |
| --- | --- | --- | --- |
| Изучить текущий бизнес процесс заказчика | Шеремет А. А. | с 16.09 по 23.09 | Анализ существующего бизнес-процесса заказчика, места встраивания проекта. Собрать информацию о текущем процессе, его слабых местах. |
| Изучить подходы к экстракции текста, кроме трансформеров | Устинов И. А. | с 23.09 по 30.09 | Изучить и протестировать различные методы экстракции текста, которые не связаны с моделями трансформерами (например, LSI) чтобы выбрать наилучший подход для поставленной задачи. |
| Создать макет интерфейса | Устинов И. А. | с 30.09 по 07.10 | Создать макет для телеграмм-бота, продумать визуальные элементы. |
| Сделать интерфейс телеграмм бота | Устинов И. А. | с 07.09 по 14.10 | Реализовать интерфейс телеграмм-бота с возможностью взаимодействия с пользователем, проработать функционал кнопок и команд для выполнения пользовательских запросов. |

## 

Продолжение таблицы 2

| Реализация подходов к  экстракции текста, кроме трансформеров | Устинов И. А. | с 28.11 по 11.11 | Реализовать изученные методы экстракции текста. |
| --- | --- | --- | --- |
| Получение доступа к API | Шеремет А. А. | с 21.10 по 04.11 | Получить доступ к API МФЦ для работы с базой данных МФЦ. |
| Изучение и поиск моделей трансформеров | Шеремет А. А. | с 21.10 по 04.11 | Проанализировать и протестировать доступные модели трансформеров для экстракции текста. Оценить их эффективность и применимость к поставленной задаче. |
| Выгрузка данных из базы данных с помощью API | Шеремет А. А. | с 04.11 по 11.11 | Извлечь данные из базы данных с помощью API. |
| Предобработка данных, полученных с помощью API | Шеремет А. А. | с 11.11 по 18.11 | Выполнить предобработку текстовых данных, удалив лишние элементы и приведя текст к стандартному виду. |
| Получение примеров клиентских запросов | Шеремет А. А. | с 21.10 по 04.11 | Получить примеры реальных запросов от клиентов для дальнейшего использования в тестировании. |
| Использование готовых моделей или алгоритмов по экстракции текста | Шеремет А. А. | с 18.11 по 25.11 | Использовать те методы для экстракции текста, которые реализованы и подходят к текущей задаче проекта. |

Продолжение таблицы 2

| Проверить качество моделей-трансформеров | Шеремет А. А. | с 25.11 по 05.12 | Оценить качество работы моделей на реальных примерах. Внести необходимые корректировки в модели и алгоритмы для повышения точности. |
| --- | --- | --- | --- |
| Протестировать ПО | Устинов И. А. | с 02.12 по 09.12 | проверить функциональность программного обеспечения. Проверить корректность работы всех компонентов системы. |

## 2.3. СОБСТВЕННЫЕ ТЕОРЕТИЧЕСКИЕ РАЗРАБОТКИ ПРИ РЕАЛИЗАЦИИ ПРОЕКТА

### 2.3.1. ИСПОЛЬЗОВАНИЕ BERT В ПРОЕКТЕ

**Разметка данных для BERT**

Мы использовали метод LDA (Latent Dirichlet Allocation), разделяя данные на пять тематических групп, чтобы упростить процесс дальнейшей категоризации и анализа. После этого мы вручную проанализировали результаты и разделили документы на несколько категорий на основе ключевых слов, выделенных LDA.

LDA выделил множество ключевых слов, которые помогли нам условно разделить документы на следующие категории:

Тема 1: Документы и юридические аспекты

* Ключевые слова: документ, лицо, государственный, услуга, представитель, юридический, заявление, регистрация, налоговый
* Описание: В этой теме фокус на документах и юридических процессах, связанных с получением государственных услуг и регистрацией заявлений. Могут упоминаться юридические лица и различные государственные органы.

Тема 2: Земельные вопросы и объекты недвижимости

* Ключевые слова: земельный, участок, объект, строительство, территория, собственность, федеральный
* Описание: Эта тема касается земельных участков и прав на них, процесса предоставления участков для строительства и государственной собственности.

Тема 3: Социальные выплаты и семья

* Ключевые слова: гражданин, семья, ребёнок, социальный, выплата, почта, заявление, жилой
* Описание: В центре внимания социальные услуги и выплаты, связанные с детьми и семьями, оформление заявлений на получение социальных пособий и поддержка граждан.

Тема 4: Заявления и регистрация прав

* Ключевые слова: гражданин, орган, заявление, место, лицо, рф, регистрация, законный
* Описание: Основной акцент на регистрации прав граждан, оформлении заявлений, а также юридических действиях, связанных с правами на жилье и другие государственные услуги.

Тема 5: Социальные услуги и выплаты

* Ключевые слова: документ, заявитель, ребёнок, услуга, социальный, выплата, мфц, решение
* Описание: Эта тема охватывает социальные услуги и выплаты, связанные с детьми и гражданами, которые обращаются за помощью в МФЦ или другие государственные органы.

Использование LDA позволило нам эффективно разделить документы на тематические категории, что упростило дальнейшую разметку данных для обучения модели BERT. После на основе полученных категорий была обучена модель BERT.

Для оценки качества классификации использовались следующие метрики:

1. **Accuracy**: Процент правильно классифицированных запросов.
   * Результат: **74%**
2. **Precision**: Точность предсказания для каждой категории.
   * Средняя точность: **82%**
3. **Recall**: Полнота предсказания для каждой категории.
   * Средняя точность: **78%**
4. **F1-Score**: Среднее гармоническое между Precision и Recall.

Метрики модели на тестовой выборке указаны в таблице 3.

Таблица 3

Оценка результатов классификации для модели BERT

|  | **класс** | **precision** | **recall** | **f1-score** | **количество объектов** |
| --- | --- | --- | --- | --- | --- |
| **0** | **0.57** | **0.80** | **0.67** | **5** |
| **1** | **1.00** | **1.00** | **1.00** | **2** |
| **2** | **1.00** | **0.73** | **0.84** | **11** |
| **3** | **1.00** | **0.50** | **0.67** | **6** |
| **4** | **0.55** | **0.86** | **0.67** | **7** |
| **accuracy** | **-** | **-** | | **0.74** | **31** |
| **взвешенное среднее** | **0.83** | **0.78** | **0.75** | **31** |

**Выводы:**

* Модель BERT продемонстрировала высокую точность и способность понимать сложные запросы пользователей.
* Система эффективно классифицирует документы даже при ограниченном объеме данных благодаря предобученным параметрам модели.
* Основным ограничением является время обработки запросов, которое увеличено в связи с нашим объемом данных.

### 2.3.2. ИСПОЛЬЗОВАНИЕ ГЕНЕРАТИВНЫХ МОДЕЛЕЙ В ПРОЕКТЕ

Для решения задачи автоматизированной генерации ответов пользователю на основе базы знаний МФЦ в проекте применяются генеративные модели, способные формировать связные и информативные тексты. Данный раздел посвящён описанию выбранных подходов, алгоритма работы системы и анализа результатов, полученных с использованием различных моделей. Основная цель — обеспечить высокую точность, читаемость и соответствие ответов исходным данным, что является ключевым аспектом в создании удобного и эффективного инструмента взаимодействия с пользователями.

**Описание подхода генерации ответа пользователю**

Для генерации текстов использовалась модель **rugpt3medium\_based\_on\_gpt2**. Алгоритм работы системы включает следующие этапы:

1. **Предобработка запроса пользователя**. Очистка текста и нормализация.
2. **Классификация темы запроса**. Определение темы с помощью модели **BERT**.
3. **Поиск документа внутри темы**. Использование косинусного сходства между эмбеддингами запроса и документов темы.
4. **Извлечение ID документа**. Применение регулярных выражений для поиска числового идентификатора документа в строке *service\_name*.
5. **Формирование промта**. Учет ID документа и текстового контекста для создания запроса модели.
6. **Генерация текста**. Формирование связного ответа с использованием модели **rugpt3medium\_based\_on\_gpt2**.

Для поиска релевантного документа использовались эмбеддинги запросов и документов, сравниваемые по косинусному сходству. Для выбора подходящей модели был проведён эксперимент, в ходе которого разные модели генерировали ответы на один и тот же запрос.

**Пример запроса моделям:***«Опиши, какие шаги нужно предпринять для оформления услуги: Принятие решение о оформлении заграничного паспорта»  
Текст документа из базы знаний*

**Метрики оценки**

Результаты генерации оценивались по следующим критериям:

1. **Согласованность текста с исходными данными**. Соответствие информации в ответе содержимому базы знаний.
2. **Соответствие требованиям задачи**. Полнота и релевантность ответа запросу пользователя.
3. **Читаемость и полнота текста**. Простота восприятия текста и наличие всех необходимых деталей.

**Результаты экспериментов**

1. **GPT-2**:  
   Ответ:  
   «Восплайспендо в прединуновическив прасподений сулиетел - пред какие шаги шальборен сплакчного пред: в то постоянствующих удостоверениео продатьарте неропо иметь. Секда, го соормраздлова скрутиа в только пр»  
   Оценка:

* Текст не согласован с исходными данными.
* Не соответствует требованиям задачи.
* Текст не читаем.

1. **rugpt3small\_based\_on\_gpt2**:  
   **Ответ:***«Для кого: для всех тех кто хочет начать бизнес в интернете. Кому важно получить доступ к Интернету? Кто планирует зарабатывать от 300 до 600 долларов в месяц? А может быть у вас уже есть свой сайт или блог? Зарабатывать деньги можно на своем сайте!»***Оценка:**

* Текст не согласован с исходными данными.
* Не соответствует требованиям задачи.
* Текст читаем.

1. **rugpt3medium\_based\_on\_gpt2**:  
   **Ответ:***«Для начала надо иметь паспорт, а потом уже можно будет оформлять и гражданство. Нужны какие-то документы: документ об образовании или о среднем специальном образовании (для получения гражданства), свидетельство пенсионного страхования с указанием стажа работы по специальности не менее 3 лет; справка из банка о наличии средств на счете в банке от 30 до 100 тысяч рублей, если есть такой счет - ксерокопия паспорта РФ (если нет).»***Оценка:**

* Текст согласован с исходными данными.
* Соответствует требованиям задачи.
* Текст читаем.

**Сравнение моделей**

Результаты экспериментов показывают, что использование модели **GPT-2** приводит к низкому качеству текста из-за неспособности учитывать контекст. Использование модели **rugpt3small\_based\_on\_gpt2**, которая обучена на корпусе с русским текстом улучшило результаты, но текст по-прежнему оставался недостаточно согласованным с исходными данными. Применение модели **rugpt3medium\_based\_on\_gpt2** позволило достичь высоких результатов благодаря её способности учитывать контекст и генерировать точные, читаемые ответы, полностью соответствующие исходным данным.

Таким образом, модель **rugpt3medium\_based\_on\_gpt2** была выбрана для интеграции в проект в качестве основного инструмента генерации текстовых ответов.

# 

### 2.3.3. СОБСТВЕННЫЕ ПРАКТИЧЕСКИЕ РЕШЕНИЯ, РЕАЛИЗОВАННЫЕ В ПРОЕКТЕ.

# 

**1. Парсер базы знаний**

Для загрузки данных с API МФЦ написан скрипт на python, который позволяет выгрузить текст документов из базы знаний для последующего применения его в обучении моделей машинного обучения.

Итог: Автоматизация сбора данных (парсер API) позволила оперативно подготовить большой объем информации для анализа.

**2. Тематическая кластеризация с использованием LDA**

Для упрощения поиска и структурирования данных был реализован процесс тематической кластеризации документов с использованием модели LDA:

* **Цель**: Для достижения высоких результатов классификации и генерации текстов была разработана система тематического разделения данных. Основной целью являлась подготовка данных для более точного обучения моделей и упрощение анализа пользовательских запросов.
* **Реализация**:
  + Документы проходят предварительную обработку (токенизация, удаление стоп-слов и лемматизация).
  + LDA анализирует текст и формирует список тем, каждая из которых описывается набором ключевых слов.
  + Каждый документ привязывается к одной или нескольким темам на основании вероятностей.

Итог: Тематическая кластеризация с помощью LDA упростила структуру данных, что существенно сократило время поиска.

Для обработки пользовательского запроса реализован гибридный подход, который сочетает LDA и модель трансформера BERT:

* **Поиск по темам**:
  + Запрос пользователя преобразуется в векторное представление.
  + С помощью косинусного расстояния вычисляется сходство запроса с темами, сформированными LDA.
  + Выбираются наиболее похожие темы, чтобы ограничить объем документов для поиска.
* **Поиск по документам**:
  + В пределах выбранных тем используется модель BERT для оценки текстового сходства между запросом и содержимым документов.
  + Итогом является ранжированный список из 5 наиболее подходящих документов.

Итог: Гибридный подход (LDA + BERT) обеспечил сочетание тематического анализа и точного поиска по содержимому документов.

# 

# ГЛАВА 3. ДЕТАЛИ РЕАЛИЗАЦИИ ПРОЕКТА

## 3.1. ОПИСАНИЕ АРХИТЕКТУРЫ

**Архитектура проекта**

Архитектура проекта основана на модульном подходе с четко выделенными слоями. Графическое представление архитектуры представлено в приложении 1.

**Слои проекта**

1. **База данных:**База данных МФЦ, включает информацию о доступных услугах, необходимых документах, и процедурных инструкциях.
2. **Модуль NLP (представлен в приложении 2):**
   * **Функциональность:**Обработка текстовых запросов с использованием методов обработки естественного языка.
   * **Инструменты:**
     + **LDA (Latent Dirichlet Allocation):**Используется для тематического анализа запросов пользователей, чтобы выделять ключевые темы (например, "оформление документов", "получение субсидий").
     + **BERT (Bidirectional Encoder Representations from Transformers):**Применяется для семантического анализа и поиска наиболее релевантных документов в базе знаний.
3. **API:**
   * **Инструменты:  
     FastAPI** используется для разработки backend-части проекта.
   * **Функциональность:**
     + Обработка входящих запросов от Telegram-бота.
     + Передача запросов в модуль NLP и базу данных.
     + Формирование и отправка ответа.
   * **Преимущества выбора FastAPI:**
     + Удобная работа с асинхронными операциями.
4. **Интерфейс (Взаимодействие с интерфейсом представлено в приложении 3):**
   * **Инструменты:  
     python-telegram-bot** для взаимодействия с пользователями через мессенджер Telegram.
   * **Функциональность:**
     + Обработка кнопок и сообщений от пользователей.
     + Отправка структурированных ответов (текст, ссылки, файлы).

## 3.2. ОПИСАНИЕ ПРОЦЕССА ОБРАБОТКИ ЗАПРОСОВ

1. **Получение запроса:**
   * Пользователь нажимает на кнопки, а затем отправляет сообщение с запросом в Telegram боту.
2. **Передача запроса в API:**
   * Бот отправляет запрос на сервер через **FastAPI**.
   * Сервер определяет тип запроса:
     + Составление дорожной карты.
     + Уточнение запроса.
     + Оставление отзыва.
3. **Обработка запроса модулем NLP:**
   * **BERT** ищет релевантные документы в базе данных.
   * Если запрашиваемая информация отсутствует, модуль возвращает сообщение с просьбой уточнить запрос.
   * Если запрашиваемая информация присутствует, пользователю предлагается топ-5 документов, похожих на его запрос
4. **Взаимодействие с базой данных (базой знаний МФЦ):**
   * Получение информации о документах из базы знаний МФЦ, с последующим обучением и использованием моделей **BERT**, **LDA**.
5. **Формирование ответа:**
   * FastAPI формирует структурированный ответ (текст или документы).
   * Отправляет данные в интерфейс Telegram-бота.
6. **Отправка ответа пользователю:**
   * После того как пользователь выбрал подходящий документ из предложенных, модели трансформеру отправляется запрос наподобие: ”сгенерируй кратко, что нужно для оформления этой услуги \*текст документа\*. Ответ модели предоставляется пользователю

# 

# 

# ГЛАВА 4. ФУНКЦИОНАЛ И ТЕСТИРОВАНИЕ ПРОЕКТА

## 4.1. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ПРОЕКТА

Проект включает три основные функциональные возможности, которые обеспечивают интерактивное взаимодействие пользователя с ботом: **составление дорожных карт**, **уточнение запросов** и **сбор обратной связи**.

**1. Составление дорожных карт**

* **Описание функции:**Бот формирует индивидуальные инструкции (дорожные карты) для пользователей в зависимости от их запросов, связанных с обслуживанием в МФЦ.
* **Особенности:**
  1. Запросы классифицируются по темам
  2. На основе введенного пользователем запроса бот определяет необходимую последовательность действий, включая:
     1. Перечень документов, которые нужно подготовить.
     2. Этапы обращения в МФЦ.
  3. Инструкции предоставляются в структурированном виде с понятными шагами.
* **Пример сценария:**
  1. Пользователь вводит запрос: *"Как оформить загранпаспорт?"*
  2. Бот формирует ответ и отправляет пять результатов на основе запроса пользователя которые будут выглядеть примерно так: \*”оформление загранпаспорта”\* и еще 4 примерно похожих на запрос пользователя названий услуг.
  3. Пользователь выбирает подходящий к его запросу ответ чат-бота и получает примерно следующий ответ:  
     \*"Для оформления загранпаспорта выполните следующие шаги:
     1. Подготовьте документы: паспорт гражданина РФ, заявление, квитанцию об оплате пошлины.
     2. Запишитесь на прием в ближайшее отделение МФЦ и Подайте документы.
     3. Дождитесь уведомления о готовности паспорта."

**2. Уточнение запросов пользователей**

* **Описание функции:**Бот помогает пользователям уточнять их запросы, если изначальный ввод недостаточно конкретен или не позволяет найти нужные данные.
* **Особенности:**
  1. Если запрос не точный, бот задает уточняющие вопросы, чтобы детализировать потребность пользователя.
  2. Лимит уточнений — 3 итерации.
  3. После достижения лимита бот предлагает переформулировать запрос.
* Пример сценария:
  1. Пользователь вводит запрос: *"Оформление пособий."*
  2. Бот не находит документы, которые соответствуют запросу пользователя.
  3. Бот выводит сообщение с просьбой уточнить запрос.
  4. Пользователь отвечает: *"Детское пособие."*
  5. После получения необходимых уточнений бот предоставляет нужную информацию.

**3. Сбор обратной связи**

* **Описание функции:**Бот принимает отзывы от пользователей, чтобы понять, на что стоит обратить внимание работников МФЦ.
* **Особенности:**
  1. Возможность оставить отзыв через специальную кнопку.
  2. Проверка валидности сообщения (например, определение спам сообщений).
  3. Сохранение валидных отзывов в базе данных для последующего анализа.
* **Пример сценария:**
  1. Пользователь нажимает кнопку *"Оставить отзыв"*.
  2. Бот предлагает: *"Пожалуйста, напишите ваш отзыв"*
  3. Пользователь вводит текст: *"Очень полезный сервис, помог разобраться с документами!"*
  4. Бот отвечает: *"Ваш отзыв успешно отправлен!"*
  5. Отзыв сохраняется в базе данных вместе с датой и временем отправки.

Такой интерактивный подход взаимодействия с пользователем возможен с помощью с внутренним свойством Telegram-bot-api, **inline mode** [6, inline-mode]

Основные особенности **inline mode**:

1. **Запросы без команд**: Пользователи не обязаны вводить команды, они могут использовать бота непосредственно в процессе общения. А в нашем случае продолжать общение после первого выполнения команды /start.
2. **Мгновенные ответы**: Бот предоставляет ответы в реальном времени, обычно это текст, изображения, кнопки или другие элементы интерфейса.

## 4.2. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ПРОЕКТА.

Интерфейс чат-бота включает в себя две основные кнопки “Составить запрос” и “Составить отзыв” с соответствующими сценариями, блок-схема с сценарием представлена в приложении 3:

**Сценарий 1: Кнопка "Составить запрос"**

1. **Реакция на нажатие кнопки:**
   * Бот отвечает сообщением: *"Введите ваш запрос"*
2. **Ввод пользователем запроса:**
   * Пользователь вводит текст запроса.
3. **Поиск и обработка запроса:**
   * Бот обрабатывает запрос с помощью алгоритмов NLP по типу BERT или LDA.
   * Производится поиск подходящих документов в базе данных.
4. **Результаты поиска:**
   * Если бот находит документы:
     + Бот отправляет сообщение с результатами: *"5 услуг, которые соответствуют вашему запросу:"*.
     + Выводит список из 5 наиболее похожих к запросу услуг.
     + Пользователь выбирает подходящий ему услугу. Модель трансформер выдает ему ответ с примерной дорожной картой того, что пользователю надо сделать чтобы получить услугу
   * Если бот не находит документов:
     + Бот отвечает: *"Не удалось найти точного соответствия. Пожалуйста, уточните ваш запрос."*
     + Пользователь может ввести уточняющий запрос до 3 раз.
     + Если после трех уточнений документы не найдены, бот отправляет сообщение с просьбой переформулировать запрос.

**Сценарий 2: Кнопка "Оставить отзыв"**

1. **Реакция на нажатие кнопки:**
   * Бот отвечает сообщением: *"Пожалуйста, напишите ваш отзыв"*
2. **Ввод сообщения пользователем:**
   * Пользователь вводит текст отзыва.
3. **Бот отправляет подтверждение: *"Ваш отзыв успешно отправлен!"***
4. **Проверка валидности отзыва:**
   * Бот проверяет сообщение на валидность:
     + Отсутствие некорректных символов или спама.
   * Если сообщение валидно:
     + Отзыв добавляется в базу данных с сохранением следующих данных:
       - Время отправки.
       - Текст отзыва.
   * Если сообщение невалидно:
     + Не происходит ничего, отзыв никуда не добавляется

# 

## 4.3.ТЕСТИРОВАНИЕ ПРОЕКТА

Тестирование проводилось с замером нескольких параметров:

1. **Согласованность текста с исходными данными**. Соответствие информации в ответе содержимому базы данных.
2. **Соответствие требованиям задачи**. Полнота и релевантность ответа запросу пользователя.
3. **Читаемость и полнота текста**. Простота восприятия текста и наличие всех необходимых деталей.
4. **Время**. Промежуток времени между выбором варианта услуги и ответом чат-бота с необходимой информацией

1 тестирование (скриншот переписки с чат ботом в приложении 4):

1. Текст согласован с исходными данными
2. Так как на данный момент нету возможности составить дорожную карту, это не выполняется.
3. Текст читаем и полон деталями
4. Время 2.52 секунды

2 тестирование (скриншот переписки с чат ботом в приложении 5):

1. Текст согласован с исходными данными
2. Так как на данный момент нету возможности составить дорожную карту, это не выполняется.
3. Текст читаем и полон деталями
4. Время 4.14 секунды

Тестирование показало, что текстовые ответы чат-бота согласованы с исходными данными, а также являются читаемыми. Однако, из-за отсутствия функционала по составлению дорожной карты, требование по полноте задачи не выполнено. Среднее время отклика составляет 3.3 секунды

## 

# ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта была решена задача автоматизации предоставления информации пользователям многофункциональных центров (МФЦ). Разработано решение в виде Telegram-бота, который обеспечивает пользователей точной информацией о необходимых документах для получения услуг, снижает вероятность ошибок на этапе подготовки документов.

**Основные достижения проекта:**

1. Разработана архитектура, включающая классификацию запросов, поиск релевантных документов и генерацию текстовых ответов.
2. Внедрены генеративные модели, включая **rugpt3medium\_based\_on\_gpt2**, которая показала наилучшие результаты в задаче генерации ответов среди протестированных моделей, благодаря своей способности учитывать контекст.
3. Проведены эксперименты для оценки качества работы различных моделей, подтверждено, что выбранный подход обеспечивает читаемость, согласованность и соответствие текстов исходным данным.
4. Создан функциональный прототип системы, готовый к дальнейшей интеграции в рабочие процессы МФЦ.

**Практическая значимость проекта**Решение позволяет оптимизировать взаимодействие граждан с МФЦ за счёт сокращения количества визитов и предоставления полной информации в удобном текстовом формате. Это снижает нагрузку на сотрудников центров и увеличивает удовлетворённость пользователей.

**Возможности дальнейшего развития:**

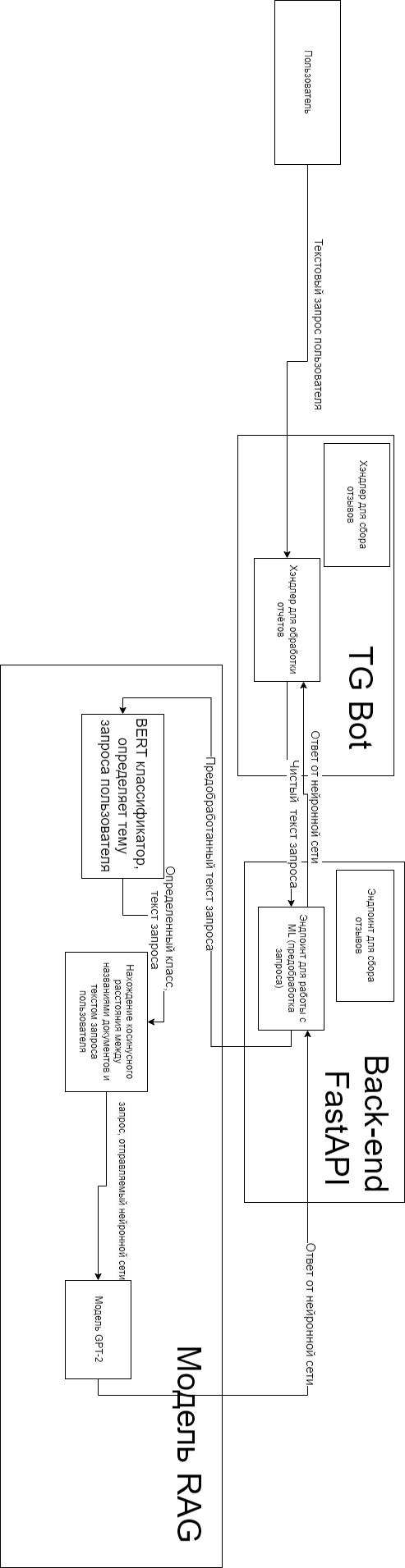
* **Масштабирование функционала.** Добавление новых типов запросов.
* **Улучшение классификации.** Повышение точности определения темы запроса с помощью дообучения модели BERT.
* **Интеграция с системой предварительной записи.** Возможность сразу записаться на услугу после получения информации о необходимых документах.
* **Формирование дорожных карт:** внедрение возможности составления дорожных карт, что обеспечит пользователям пошаговое руководство для достижения их целей, начиная от получения информации до записи на услугу. Это повысит удобство использования и эффективность взаимодействия с инфраструктурой МФЦ.

**Вывод:**Результаты проекта показали, что использование современных методов обработки естественного языка, включая классификацию запросов и генерацию текстов, может существенно улучшить процессы взаимодействия с пользователями в государственных учреждениях. Разработанное решение подтверждает свою эффективность в условиях ограниченных ресурсов и служит отправной точкой для дальнейшего развития систем автоматизации в МФЦ.

# **СПИСОК ЛИТЕРАТУРЫ**

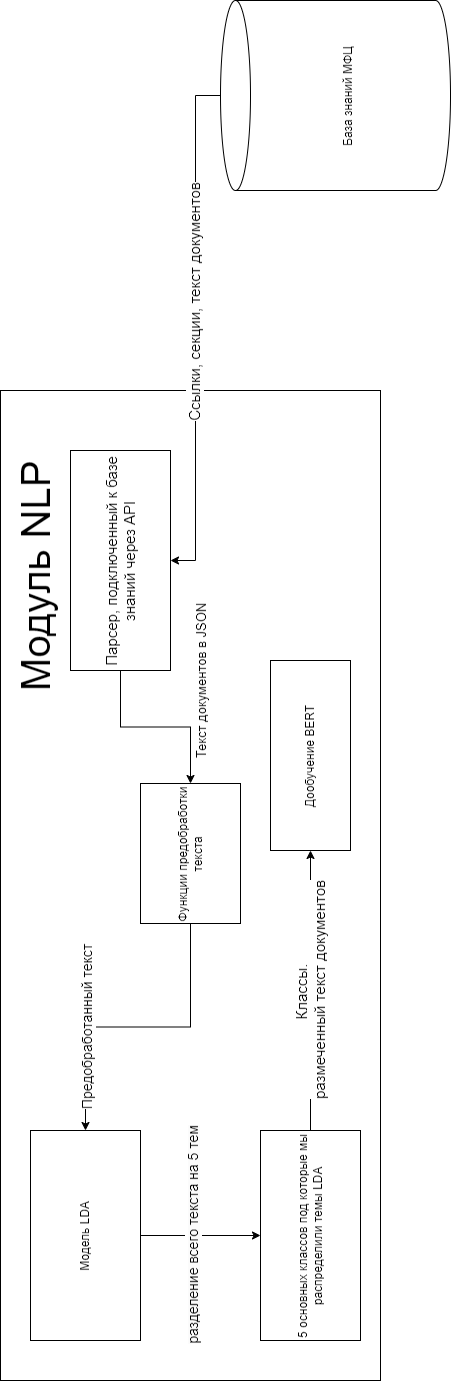
1. Vaswani A., Shazeer N., Parmar N., et al. Attention Is All You Need // *Advances in Neural Information Processing Systems (NIPS)*. 2017. P. 5998-6008.
2. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // *Proceedings of NAACL-HLT*. 2019. P. 4171-4186.
3. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // *arXiv preprint arXiv:1301.3781*. 2013.
4. Manning C., Raghavan P., Schütze H. Введение в информационный поиск // *М.: Диалектика*. 2011. 592 с.
5. Минцер Т., Гольдберг Й., Маннинг К. Приложения обработки естественного языка // *Communications of the ACM*. 2015. Т. 58, № 3. С. 77-97.
6. Telegram Bot API Documentation [Электронный ресурс] // *Telegram.org*. URL:<https://core.telegram.org/bots/api> (дата обращения: 16.12.2024).
7. FastAPI: Введение и практика применения[Электронный ресурс]. DEV Community. URL: [https://dev.to](https://dev.to/devasservice/fastapi-best-practices-a-condensed-guide-with-examples-3pa5) (дата обращения: 16.12.2024).

# Приложение 1

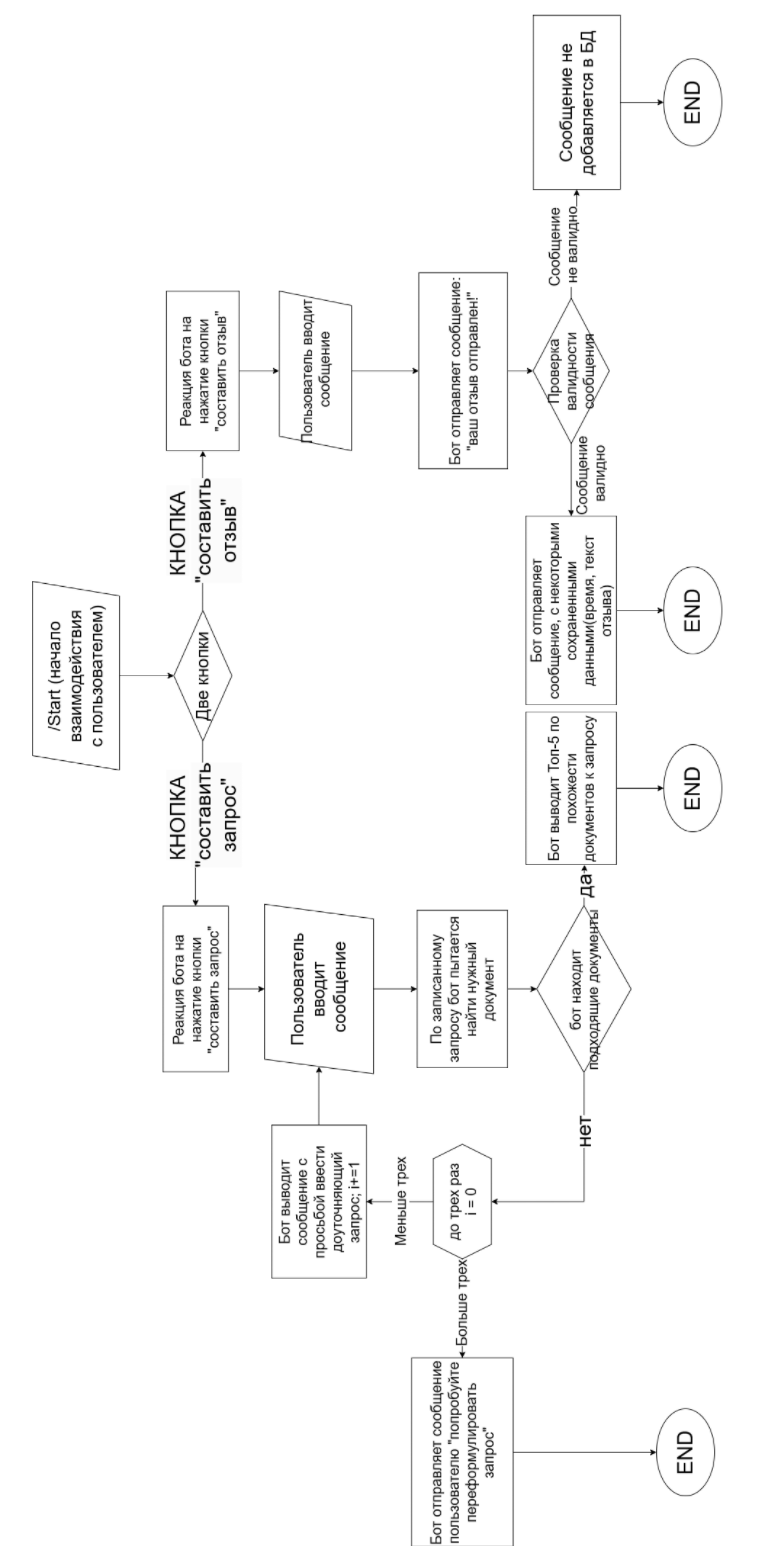
Полная архитектура взаимодействия модулей

# Приложение 2

Архитектура модуля NLP

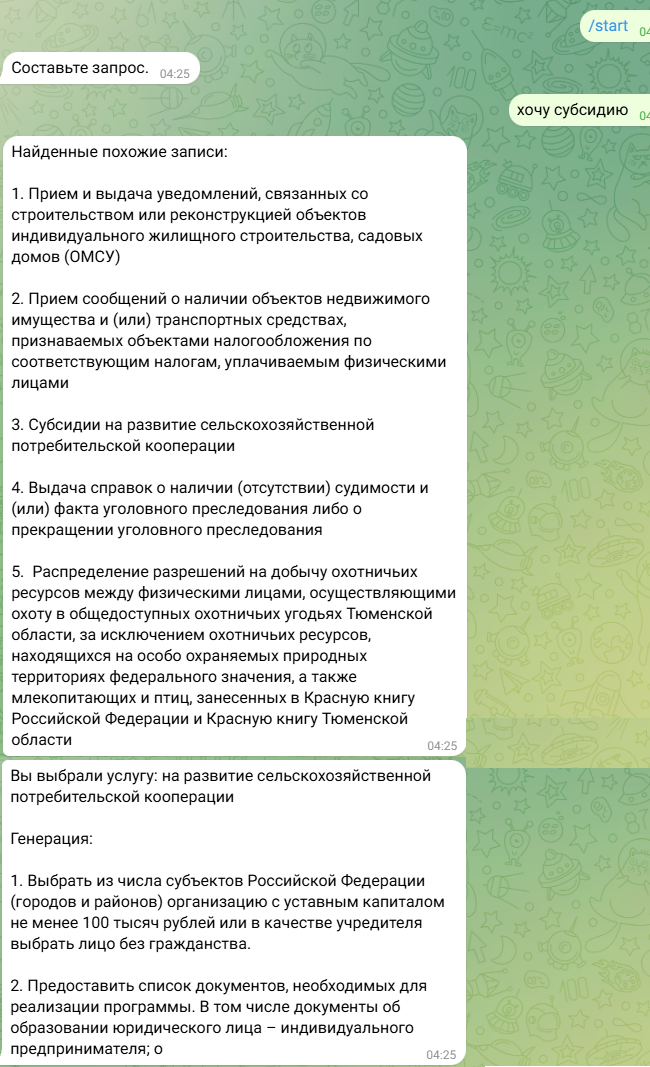


# Приложение 3

Блок-схема пользовательского сценария

# Приложение 4

Скриншот переписки с чат-ботом в телеграмме



# Приложение 5

Скриншот переписки с чат-ботом в телеграмме

