



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего профессионального образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования



@RVISION_TIP_QA_BOT

Разработка чат-бота для ответов на вопросы по продукту ТПР

Научный руководитель:
к.ф.-м.н., доцент кафедры СП
С.А. Иванов

Автор:
студент группы КЭ-404
Д.В. Тропин

Челябинск, 2024



ПРОДУКТ R-VISION TIP

R-Vision



@RVISION_TIP_QA_BOT

Платформа анализа информации об угрозах R-Vision TIP: собирает, обрабатывает и агрегирует данные об угрозах информационной безопасности, а также позволяет автоматизировать часто повторяющиеся операции над ЭТИМИ ДАННЫМИ



ЦЕЛЬ И ЗАДАЧИ



@RVISION_TIP_QA_BOT

Цель работы:

Разработка чат-бота для ответов на вопросы по продукту TIP

Задачи:

1. Провести анализ предметной области и обзор существующих решений
2. Спроектировать и реализовать чат-бота
3. Провести ряд экспериментов для подбора оптимальных параметров системы
4. Провести тестирование разработанного чат-бота



АКТУАЛЬНОСТЬ



@RVISION_TIP_QA_BOT

Чат-боты получили широкое применение по ряду причин:

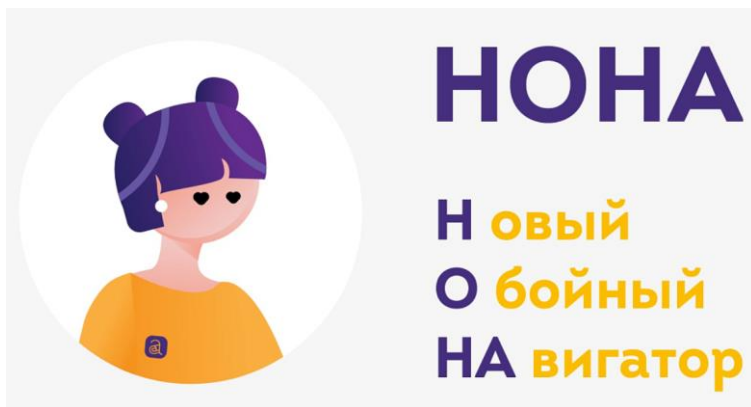
- моментальный ответ на интересующий вопрос, как следствие, повышение удовлетворенности клиентов
- автоматизация процессов освобождает время специалистов для решения более сложных задач
- возможность сбора и анализа часто задаваемых вопросов, использование этой информации для улучшения предоставляемых продуктов и услуг



ПРИМЕРЫ УСПЕШНЫХ ИНТЕГРАЦИЙ ВОПРОСНО-ОТВЕТНЫХ СИСТЕМ



@RVISION_TIP_QA_BOT





ПОДХОДЫ РАЗРАБОТКИ ВОПРОСНО-ОТВЕТНЫХ СИСТЕМ



@RVISION_TIP_QA_BOT

На данный момент самыми популярными считаются следующие подходы:

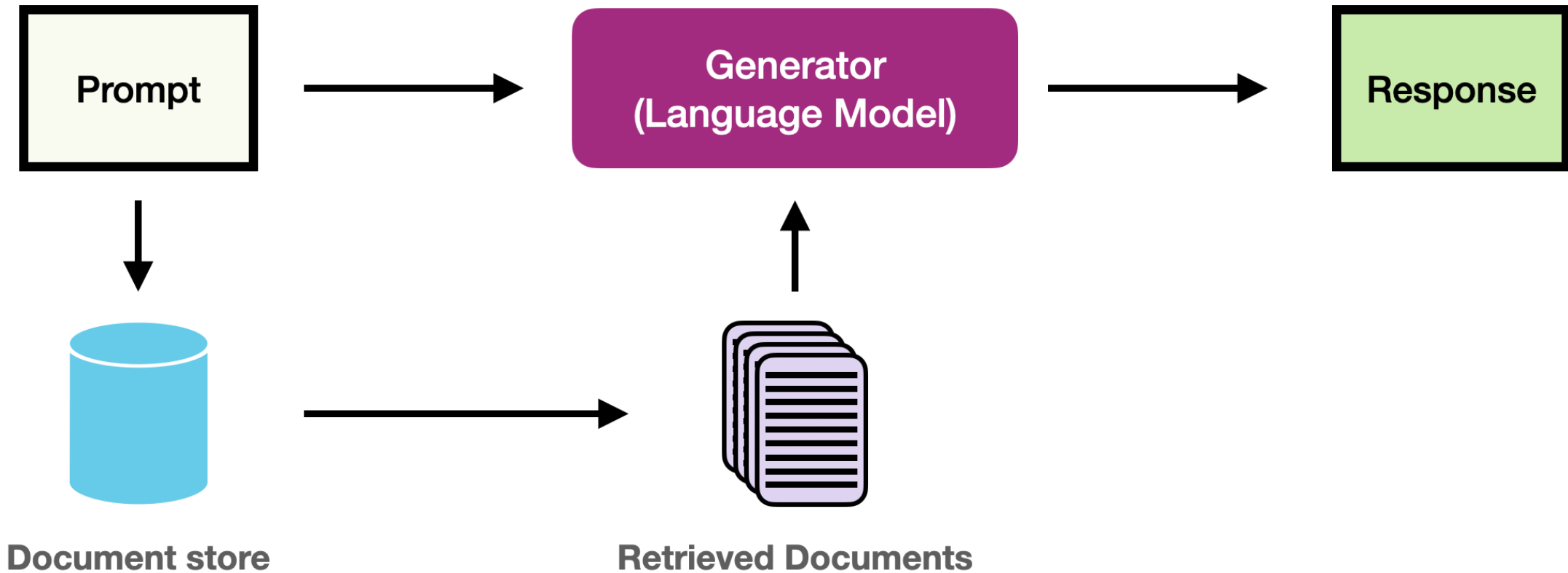
1. Векторный поиск готовых ответов по базе данных
2. Поиск и чтение – Retriever and Reader
3. Использование больших языковых моделей – Large Language Model (LLM)
4. Генерация с дополнительным поиском – Retrieval Augmented Generation (RAG)



ГЕНЕРАЦИЯ С ДОПОЛНИТЕЛЬНЫМ ПОИСКОМ



@RVISION_TIP_QA_BOT





ФУНКЦИОНАЛЬНЫЕ И НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ



@RVISION_TIP_QA_BOT

Функциональные требования:

- Чат-бот должен предоставлять ответ на вопрос, касающийся непосредственно продукта ТПР или сферы информационной безопасности
- Чат-бот должен иметь механизм обработки нерелевантных вопросов

Нефункциональные требования:

- Система чат-бота должна быть реализована на языке Python версии 3.10 или выше
- Чат-бот должен быть развернут в виде бота в мессенджере Telegram
- Чат-бот должен использовать архитектуру глубокой нейронной сети T5
- Система чат-бота должна быть развернута как Docker-контейнер
- Система чат-бота должна иметь гибкий механизм логирования



СРЕДСТВА РАЗРАБОТКИ



@RVISION_TIP_QA_BOT

- Среда разработки: PyCharm Professional Edition 2024.1
- Язык программирования Python версии 3.10
- Библиотеки и фреймворки:
 - pyTelegramBotAPI
 - transformers
 - torch
 - chromadb
 - evaluate
- Другие инструменты:
 - Docker и docker-compose
 - MS Excel



НАБОРЫ ДАННЫХ



@RVISION_TIP_QA_BOT

Исходные данные представляют собой документацию продукта ТІР, в виде 58 документов в формате docx, на основе этих данных вручную было составлено три набора данных:

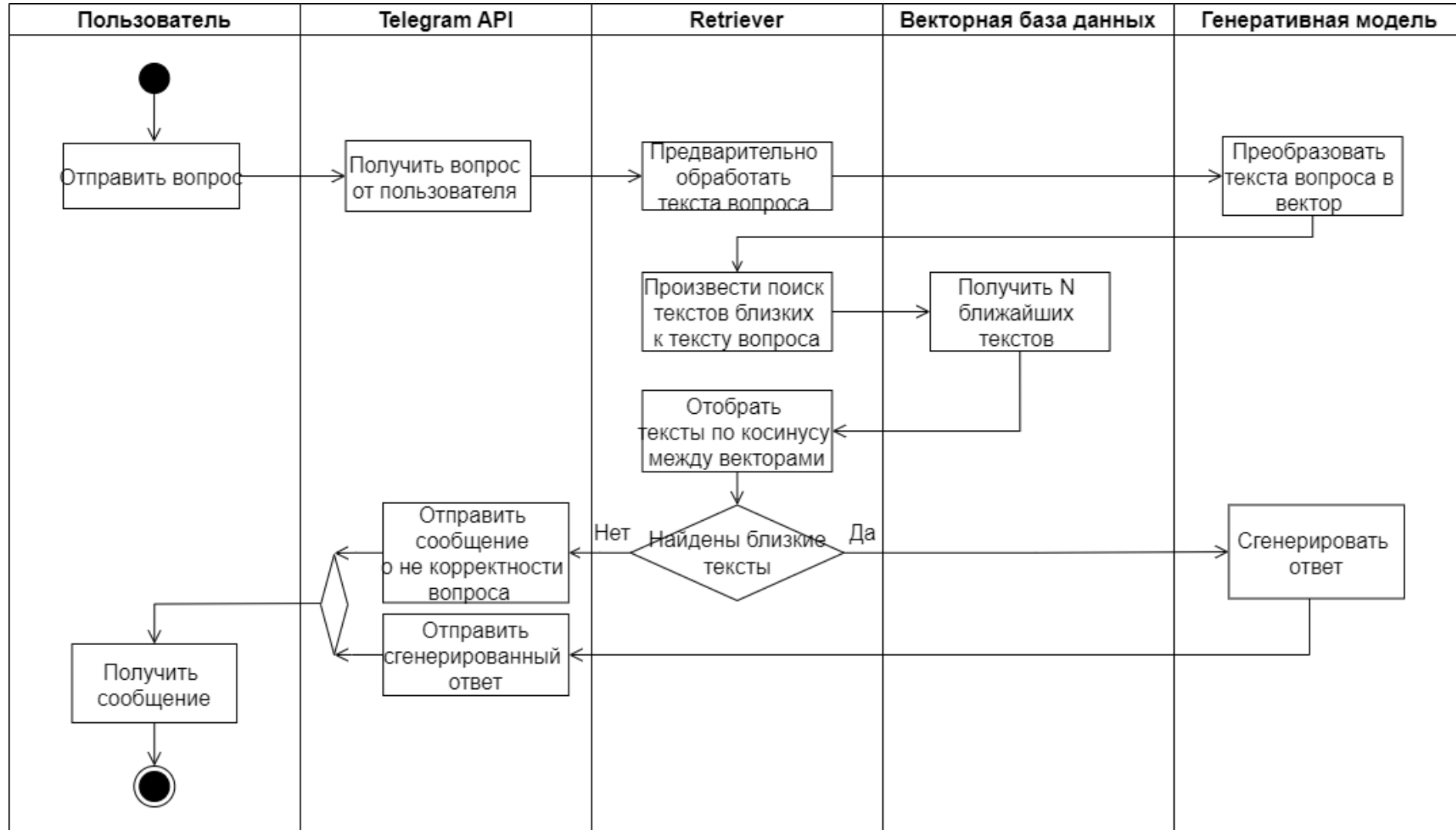
- объединение всех текстов в один документ и дальнейшее разбиение на небольшие части (chunk)
- разбиение текстов документации на тематические разделы
- перечень вопросов и ответов, охватывающих каждый раздел документации



ПОЛУЧЕНИЕ ОТВЕТА НА ВОПРОС



@RVISION_TIP_QA_BOT





ГЕНЕРАТИВНАЯ МОДЕЛЬ



@RVISION_TIP_QA_BOT

В качестве генеративной модели используется готовая большая языковая модель, построенная на архитектуре Transformers. Данная LLM является дообученной русскоязычной моделью T5 от Сбербанка: FRED-T5-LARGE_text_qa

Вход модели:

```
<SC6>Текст: {context}\nВопрос: {question}\nОтвет: <extra_id_0>
```

Выход модели:

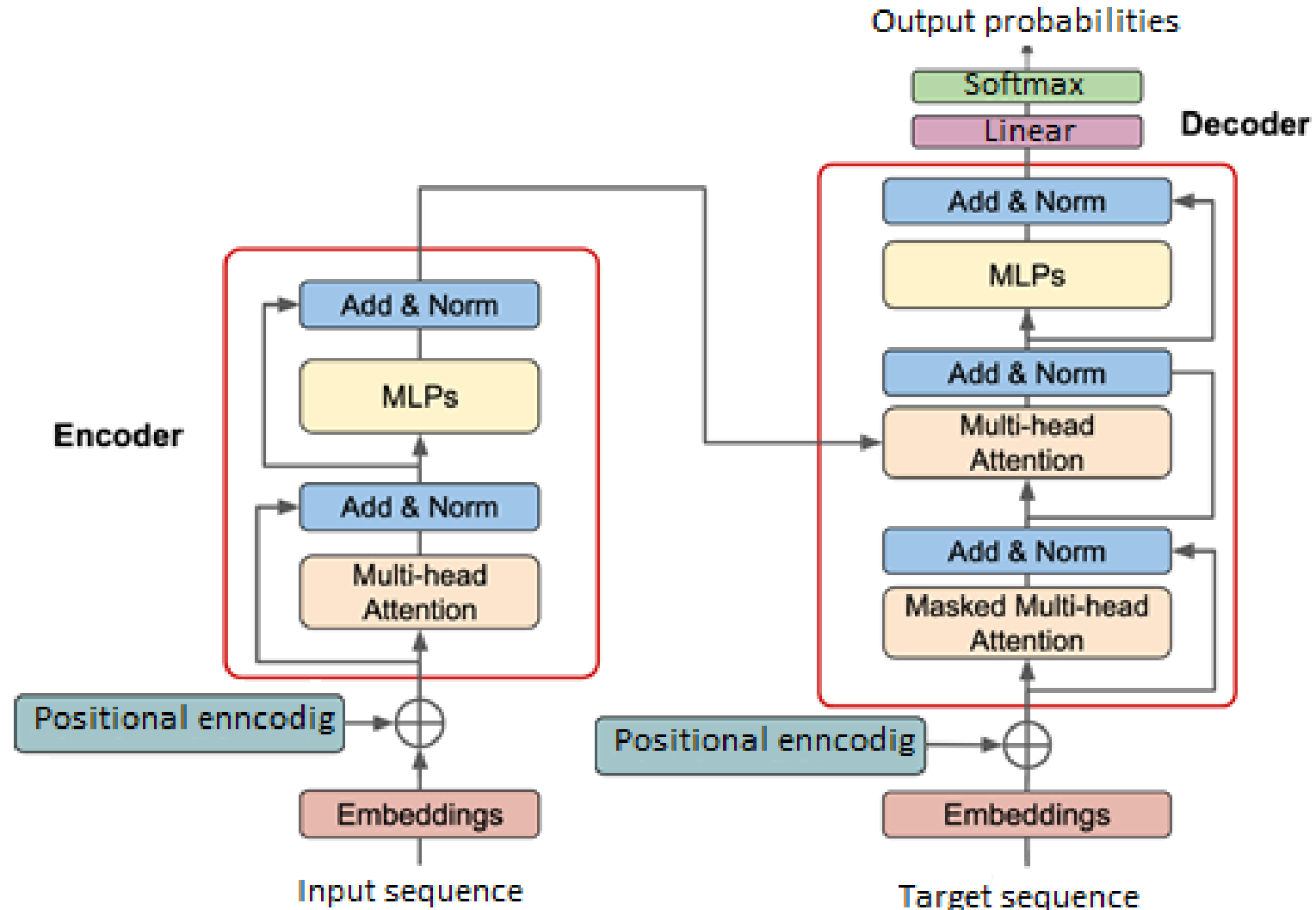
```
<SC6> {Текст ответа} <extra_id_0>
```



МОДЕЛЬ АРХИТЕКТУРЫ Transformer



@RVISION_TIP_QA_BOT





ВЕКТОРНАЯ БАЗА ДАННЫХ



@RVISION_TIP_QA_BOT

Каждый объект коллекции `vector_db` имеет следующие параметры:

- `id` – идентификационный номер документа
- `document` – текст фрагмента документации
- `embedding` – представление текста документации в векторной форме

| Document |
|-----------------------------------|
| <code>id: int</code> |
| <code>document: str</code> |
| <code>embedding: list[int]</code> |



ЭКСПЕРИМЕНТЫ ПО ОРГАНИЗАЦИИ ДАННЫХ ДЛЯ АРХИТЕКТУРЫ RAG



@RVISION_TIP_QA_BOT

Лучший подход: в модель в качестве контекста передается пара вопрос и ответ

| Предобработка | | Rouge1 | Rouge2 | RougeL | Rouge Lsum | Blue |
|---------------------|-------------------|---------------|---------------|---------------|---------------|---------------|
| для теста документа | для теста вопроса | | | | | |
| Нет | Нет | 0,2428 | 0,1330 | 0,2420 | 0,2401 | 0,2610 |
| Да | Нет | 0,2574 | 0,1478 | 0,2528 | 0,2554 | 0,3126 |
| Да | Да | 0,2594 | 0,1520 | 0,2574 | 0,2569 | 0,2541 |



РЕЗУЛЬТАТЫ МОДУЛЬНОГО ТЕСТИРОВАНИЯ



@RVISION_TIP_QA_BOT

```
Run: Python tests for test_text_process.TextProcessTest x
>> ✓ Tests passed: 7 of 7 tests – 66 ms
Test Results 66 ms
C:\Users\tropi\Desktop\TIP_bot\venv
Testing started at 18:32 ...
Launching unittests with arguments

Ran 7 tests in 0.070s

OK

Process finished with exit code 0
```



РЕЗУЛЬТАТЫ ФУНКЦИОНАЛЬНОГО ТЕСТИРОВАНИЯ



@RVISION_TIP_QA_BOT

| № | Предусловие | Действие | Ожидаемый результат | Тест пройден? |
|---|------------------|--|--|---------------|
| 1 | Чат-бот запущен. | Пользователь через мессенджер Telegram отправляет релевантный вопрос. | Чат-бот отвечает пользователю, отправляя ответ на заданный вопрос. | Да |
| 2 | Чат-бот запущен. | Пользователь через мессенджер Telegram отправляет нерелевантный вопрос. | Чат-бот отвечает пользователю, сообщением о не корректности вопроса. | Да |
| 3 | Чат-бот запущен. | Пользователь через мессенджер Telegram, отправляет картинку или стикер. | Чат-бот не отвечает пользователю. | Да |
| 4 | Чат-бот запущен. | Пользователь через мессенджер Telegram, отправляет видео или видеосообщение. | Чат-бот не отвечает пользователю. | Да |
| 5 | Чат-бот запущен. | Пользователь через мессенджер Telegram, отправляет аудио или аудиосообщение. | Чат-бот не отвечает пользователю. | Да |



ПРИМЕРЫ РАБОТЫ ЧАТ-БОТА



@RVISION_TIP_QA_BOT

Что такое платформа TIP? 18:29 ✓✓

TIP - это платформа для создания и распространения информационных бюллетеней об угрозах и уязвимостях. TIP позволяет создавать информационные бюллетени на основе постоянно обновляемых данных об угрозах, уязвимостях и их применимости к какой-либо атаке, актуальной на текущий момент. TIP может экспортировать и формировать наборы (дайджесты) за период времени. Для работы с бюллетенями доступны функции API системы.

18:29

Какие есть виды бюллетеней? 18:45 ✓✓

Бюллетени - это данные, которые фид предоставляет другим фидам для анализа и оценки. Бюллетени могут быть в виде таблиц, диаграмм, графиков и т.д. В таблице Urls представлены все виды бюллетеней, которые используются в R-Vision TIP.

18:45

Привет! 18:42 ✓✓

Некорректный вопрос 18:42



ОСНОВНЫЕ РЕЗУЛЬТАТЫ



@RVISION_TIP_QA_BOT

1. Проведен анализ предметной области и обзор существующих решений
2. Спроектирован и реализован чат-бот
3. Проведен ряд экспериментов для подбора оптимальных параметров системы
4. Проведено тестирование разработанного чат-бота



ВЫЧИСЛЕНИЕ БЛИЗОСТИ ДВУХ ВЕКТОРОВ



@RVISION_TIP_QA_BOT

Для вычисления схожести используется косинусное сходство:

$$\textit{Similarity} = \frac{\vec{E}_1 \cdot \vec{E}_2}{\|\vec{E}_1\| \cdot \|\vec{E}_2\|}$$

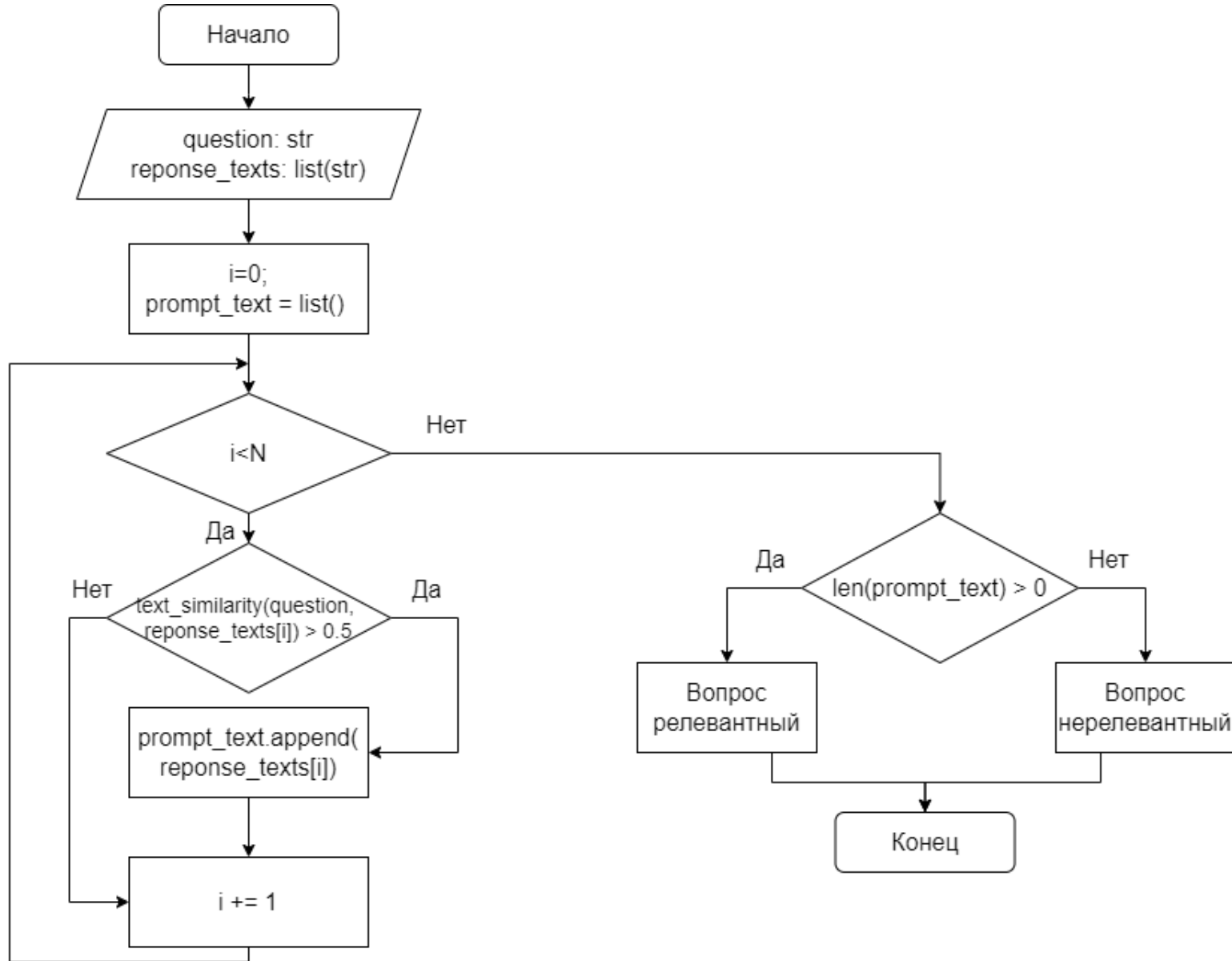
где \vec{E}_1 – векторное представление вопроса;

\vec{E}_2 – векторное представление текста документации.

АЛГОРИТМ ОТБОРА ВОПРОСОВ



@RVISION_TIP_QA_BOT





ЭКСПЕРИМЕНТЫ ПО ОРГАНИЗАЦИИ ДАННЫХ ДЛЯ АРХИТЕКТУРЫ RAG



@RVISION_TIP_QA_BOT

Все тексты документации объединены и разделены на небольшие части chunk

| Длина chunk | Перекрытие chunk | Rouge1 | Rouge2 | RougeL | Rouge Lsum | Blue |
|------------------------|-----------------------------|---------------|---------------|---------------|-----------------------|-------------|
| 500 | 150 | 0,1593 | 0,0806 | 0,1568 | 0,1569 | 0,0988 |
| 300 | 100 | 0,0827 | 0,0371 | 0,0791 | 0,0790 | 0,0786 |



ЭКСПЕРИМЕНТЫ ПО ОРГАНИЗАЦИИ ДАННЫХ ДЛЯ АРХИТЕКТУРЫ RAG



@RVISION_TIP_QA_BOT

Текст все тексты документации разделены на тематические разделы и главы.

| Предобработка | | Использование обобщения | Rouge1 | Rouge2 | RougeL | Rouge Lsum | Blue |
|--------------------------|-------------------|-------------------------|--------|--------|--------|------------|--------|
| для теста документации и | для теста вопроса | | | | | | |
| Да | Нет | Нет | 0,1513 | 0,0715 | 0,1490 | 0,1504 | 0,0779 |
| Да | Да | Нет | 0,1773 | 0,0754 | 0,1748 | 0,1761 | 0,0984 |
| Да | Да | Да | 0,1535 | 0,0684 | 0,1512 | 0,1508 | 0,0701 |



ROUGE-N МЕТРИКА



@RVISION_TIP_QA_BOT

Метрика ROUGE-N представляет собой обобщенную статистическую меру, выражающую какой процент лексических единиц (N-gram,- последовательностей из N лексем), входящих в состав ручного, построенного независимым экспертом, перевода, повторяется в автоматическом переводе

$$ROUGE - N = \frac{\sum_{S \in RefSum} \sum_{n-gram \in S} Count_{match}(n - gramm)}{\sum_{S \in RefSum} \sum_{n-gram \in S} Count(n - gramm)}$$



ROUGE-L МЕТРИКА



@RVISION_TIP_QA_BOT

$$R_{lcs} = \frac{\sum_{i=1}^v LCS(r_i, C)}{m}$$

$$P_{lcs} = \frac{\sum_{i=1}^u LCS(r_i, C)}{m}$$

$$ROUGE - L = \frac{(1 - \beta^2) \cdot R_{lcs} \cdot P_{lcs}}{R_{lcs} + \beta^2 \cdot P_{lcs}}$$

где $LCS(r_i, C)$ – длина наибольшей подпоследовательности между предложением ручного реферата r_i и всеми предложениями автоматического реферата C ,

v – количество предложений автоматического реферата (всего n слов),

u – количество предложений ручного реферата (всего m слов)

Полагается $\beta \rightarrow \infty$, таким образом, учитывается только R_{lcs} составляющая



BLUE МЕТРИКА



@RVISION_TIP_QA_BOT

$$P_n = \frac{\sum_{C \in (candidates)} \sum_{n_gram \in C} Count_{clip}(n_gram)}{\sum_{C' \in (candidates)} \sum_{n_gram' \in C'} Count_{clip}(n_gram')}$$

$$BR = \begin{cases} 1, & \text{если } c > r \\ e^{(1-\frac{r}{c})}, & \text{если } c \leq r \end{cases}$$

$$BLUE = BR \cdot \exp\left(\sum_{i=1}^n W_n \text{Log}(P_n)\right)$$

где c – длина результата модели,

r – эффективная длина,

P_n - среднее геометрическое модифицированных n -граммовых точностей