

Робототехника и комплексная автоматизация (РК)

Системы автоматизированного проектирования (РК6)

Гулько Никита Макарович

PK6-81Б

Преддипломная

ООО «ЮБС»

(Подпись, дата)

И.О. Фамилия

(Подпись, дата)

И.О. Фамилия

Оценка

Москва, 2023 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
ОСНОВНАЯ ЧАСТЬ	5
Теоретическая часть.....	5
Создание документа со знаниями	5
Алгоритм работы.....	6
Реализация	7
Демонстрация работы программы	9
ЗАКЛЮЧЕНИЕ	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	14
ПРИЛОЖЕНИЕ	15
Приложение 1. Программная реализация загрузки и представления данных в векторном виде	15
Приложение 2. Программная реализация взаимодействия с загруженной языковой моделью.....	18

ВВЕДЕНИЕ

В современном мире банки сталкиваются с растущей потребностью обеспечить клиентам мгновенный доступ к информации и услугам, особенно в сфере обслуживания через интернет. Однако, несмотря на все преимущества технологического прогресса, вопрос безопасности и конфиденциальности остается одной из основных проблем. Многие клиенты все еще опасаются использовать онлайн-сервисы из-за возможности утечки их личных данных или несанкционированного доступа к ним.

В этой работе представляется полностью локальная реализация банковского чат-бота, который позволяет клиентам задавать вопросы чат-боту, а самому виртуальному ассистенту работать без необходимости подключения к интернету. В решении использованы возможности LLM (Large Language Model) для создания интеллектуального ассистента, работающего исключительно внутри клиентской среды выполнения. Это означает, что все данные остаются строго конфиденциальными и никогда не покидают устройство пользователя.

Цель практики: поиск и реализация локального решения задачи построения банковского чат-бота с обученной языковой моделью.

Для выполнения поставленной цели необходимо решение следующих задач:

- выбрать архитектуру решения;
- найти готовую языковую модель.

ОСНОВНАЯ ЧАСТЬ

Теоретическая часть

Для решения поставленного задания было решено использовать технологию генерации человеческой речи предобученной языковой моделью на основе встроенного контекста по причине того, что это не требует поиска готовой к дообучению языковой модели и траты времени на это дообучение. Реализация такой программы была разбита на две части: поиск языковой модели, способной генерировать естественную речь и модели эмбедингов для получения наиболее релевантных по смыслу отрывков из векторной базы данных. Языковая модель была выбрана на основе технических характеристик рабочего компьютера. В качестве реализации программного решения был выбрал язык программирования Python.

Создание документа со знаниями

Для базы знаний был создан текстовый документ. Было собрано 470 строк банковских данных, взятых с сайта банка СберБанк. Все данные были переведены на английский язык из-за возникновения сложностей с работой на русском языке, так как происходит иное разбиение на токены. Небольшой отрывок получившегося документа показан на рисунке 1.

Where can I open a deposit or account? Deposits and accounts can be opened in your personal account and the Sberbank Online mobile application or web version. 2. Call the bank office. Choose a method that is convenient for you: 1. Via Sberbank Online - mobile application or web version. 2. Call the bank office. 3. At the bank office.

Where is it more profitable to open deposits - in the office or online? In Sberbank Online, deposit rates are higher than in the bank office. It will take about 3 minutes to make a deposit: just transfer money from a card or account - it's safe, and you don't have to go to the bank.

How to choose a deposit? To choose the right investment, decide on the goal you want to achieve and the actions you want to take. Deposits have different parameters: currency, term, the possibility of replenishment and withdrawal. The more transactions you make, the lower the profitability.

Can I open several deposits or accounts in my name at once? Yes, you can open any number of deposits or accounts. However, you can open 1 Savings Account and 1 Active Age Account.

Can I open 'Homeclick' or 'Good Start' deposits in the name of another person? Yes, but these deposits must be available not only to you, but also to another person. If a deposit is available to you, but not to another person, you will not be able to open it. For example, a 'Homeclick' deposit cannot be opened in the name of a client who has not made a real estate sale transaction transaction, even if you have made such a transaction.

Is it possible to open a deposit for several people at once, for example, a family one? The deposit is opened only for one person. You can appoint an attorney to manage the deposit at the bank office for free: your loved ones will be able to receive money and account statements, open and close it and transfer money to other accounts. The trusted person does not have to come to draw up a power of attorney, but you need a passport or identity document.

In what currency can I open a deposit or account? Depends on the deposit or account. SberVklad, SberVklad Prime, Manage+, Savings Account can only be opened in rubles. If you want an account in another currency, use the Savings Account, or open the 'CNV' account.

What is interest capitalization? During capitalization, interest accrued for the past period is added to the principal amount and also accrues interest. This is often referred to as compound interest.

What is the minimum deposit balance? The minimum deposit balance is the minimum amount that must be kept on your deposit during the term of the deposit.

What benefits are there for pensioners when opening a deposit? Women over 55 and men over 60 can open an Online Account with a maximum rate on it is available from 1000 rubles to everyone who receives a pension in Sberbank. It can be replenished and withdrawn. If I am a citizen of another state, but I temporarily live in Russia and I want to make a deposit. What documents do I have to provide? A passport of a foreign citizen, a temporary residence permit or a residence permit.

Is it safe to keep money on a deposit opened online? We care about the safety of your money: we monitor their movement around the clock for emerging threats. Most importantly, do not share your password with anyone.

When opening a deposit online, you can additionally connect an SMS notification to keep abreast of all operations on accounts: replenishment, withdrawal, interest accrual. In Sberbank Online, it is easy to limit the visibility of the deposit: after that, only you will see your deposit in your personal account on application, at an ATM.

Рисунок 1 – Часть подготовленных данных

Алгоритм работы

Алгоритм работы следующий:

1. Импорт библиотек: Программа начинает свою работу с импорта необходимых библиотек, таких как LangChain, Transformers и Chroma. Эти библиотеки обеспечивают функциональность анализа документов, создания вложений и работу с готовыми языковыми моделями.

2. Анализ документа: первый запуск программы начинается с анализа прикрепленных документов. Программа использует библиотеку LangChain для анализа входящих документов. LangChain обрабатывает текстовые данные и разделяет их на токены (слова, фразы и символы), выполняет поиск шаблонов и извлекает ключевые слова. Результаты анализа сохраняются в памяти для дальнейшей обработки.

3. Создание вложений: далее программа использует модель эмбедингов для создания локальных вложений на основе обработанных текстовых данных.

4. Создание хранилища векторов: с использованием библиотеки Chroma программа создает локальное хранилище векторов, где сохраняются полученные вложения. Хранилище векторов служит в качестве базы данных, в которой векторные представления связаны с соответствующими контекстами из документов.

5. Загрузка языковой модели: программа загружает локально предварительно обученную языковую модель при помощи библиотеки Transformer. Эта модель будет использоваться для понимания вопросов пользователей и генерации ответов.

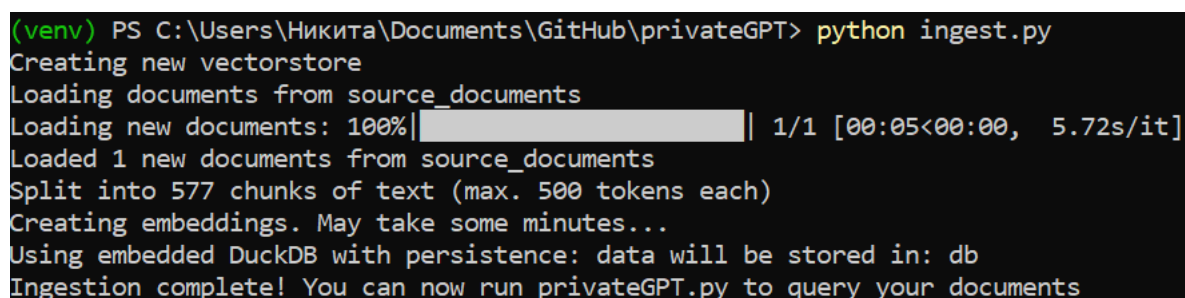
6. Поиск контекста: программа использует алгоритм для нахождения наиболее релевантного к запросу пользователя контекста из документов. Это позволяет языковой модели генерировать правдивый ответ на естественном языке.

7. Генерация ответа: на основе найденного контекста языковая модель генерирует ответ на заданный вопрос.

Реализация

Проект реализован в виде трех файлов и директории с документами, в которых хранятся данные. Первые два файла – это программы на языке программирования Python. В первом файле `ingest.py` реализована логика для подготовки векторного пространства для дальнейшего поиска в нем наиболее релевантной информации. Вторым файлом `privateGPT.py` отвечает за основную часть работы программы, а именно взаимодействие с загруженной языковой моделью. Третий файл называется `.env`. В нем собраны основные глобальные переменные, которые используются в описанных программах.

Во время первого запуска файла `ingest.py` необходимо подключение к сети Интернет, так как выполняется загрузка указанной эмбединговой модели. При дальнейших активациях этого файла не требуется подключение к сети Интернет. После загрузки модели или проверки ее наличия, создается директория `database`, куда помещается обработанный документ из директории `source_documents` в векторном виде. Реализация не имеет ограничений по количеству документов. Все загруженные знания будут собраны в локальной базе данных вложений. Во время загрузки никакие данные не покидают локальную среду (рис. 2).



```
(venv) PS C:\Users\Никита\Documents\GitHub\privateGPT> python ingest.py
Creating new vectorstore
Loading documents from source_documents
Loading new documents: 100% |████████████████████| 1/1 [00:05<00:00, 5.72s/it]
Loaded 1 new documents from source_documents
Split into 577 chunks of text (max. 500 tokens each)
Creating embeddings. May take some minutes...
Using embedded DuckDB with persistence: data will be stored in: db
Ingestion complete! You can now run privateGPT.py to query your documents
```

Рисунок 2 – Загрузка нового документа в векторную базу знаний

При запуске файла `privateGPT.py` будет показан этап активации языковой модели (рис. 3) и выведена строка для ввода запросов.


```
(venv) PS C:\Users\Никита\Documents\GitHub\privateGPT> python privateGPT.py
Using embedded DuckDB with persistence: data will be stored in: db
llama.cpp: loading model from models/ggml-vic7b-q4_0.bin
llama_model_load_internal: format      = ggjt v2 (latest)
llama_model_load_internal: n_vocab    = 32000
llama_model_load_internal: n_ctx      = 1000
llama_model_load_internal: n_embd     = 4096
llama_model_load_internal: n_mult     = 256
llama_model_load_internal: n_head     = 32
llama_model_load_internal: n_layer    = 32
llama_model_load_internal: n_rot      = 128
llama_model_load_internal: ftype      = 2 (mostly Q4_0)
llama_model_load_internal: n_ff       = 11008
llama_model_load_internal: n_parts    = 1
llama_model_load_internal: model size = 7B
llama_model_load_internal: ggml ctx size = 72.75 KB
llama_model_load_internal: mem required = 5809.34 MB (+ 1026.00 MB per state)
llama_init_from_file: kv self size = 500.00 MB
AVX = 1 | AVX2 = 1 | AVX512 = 0 | AVX512_VBMI = 0 | AVX512_VNNI = 0 | FMA = 1 | N
D = 0 | BLAS = 0 | SSE3 = 1 | VSX = 0 |
Enter a query:
```

Рисунок 3 – Инициализация языковой модели при запуске программы

Демонстрация работы программы

Демонстрация работы программы произведена на нескольких тестовых запросах. Включен режим показа дополнительного контекста, на который опирается языковая модель при генерации ответа на вопросы.

```

> Question:
Hello! tell me about deposit named SberDeposit

> Answer:
Sure! SberDeposit is a type of deposit offered by Sberbank, which allows customers to

> source_documents\sber_en.txt:
SberDeposit and SberDeposit Prime have their own advantages - the client can periodically
rest will be charged on a large amount. This will increase the benefits and accumulate e
count until the expiration date. The interest rate is up to 7.2% per annum. To open a de
0 rubles. You can place them for a period from 1 month to 3 years. The maximum rate is

> source_documents\sber_en.txt:
Social. This is a deposit for orphans, children without parental care, veterans and inva
deposit will be 1.95%, the term is 3 years, you can deposit an amount starting from 1 ru
ible to withdraw the accrued interest.
Depending on which deposits in Sberbank you are interested in, you can issue them at the
banking.

> source_documents\sber_en.txt:
of the deposit term. If the accrual of interest falls on a non-working day or holiday, t
day. For SberDeposit and SberDeposit Prime opened after March 3, 2022, interest is accru
of the deposit is a weekend or a holiday, then the accrual of interest is transferred t
decide to close the deposit on that day, you get money with interest. You can read

```

Рисунок 4 – Ответ чат-бота на входящий запрос

```

> Question:
What happens if there is not enough money on the debit card on the day of the automatic transfer?

> Answer:
If there is not enough money on the debit card on the day of the automatic transfer, you will receive an SMS or a push that the transfer
cannot be completed. The next attempt will be repeated the next day. If even then there is not enough money on the card, the transfer will
not be executed. A new attempt will be repeated in the next period according to the schedule.

> source_documents\sber_en.txt:
Autotransfers. What happens if there is not enough money on the debit card on the day of the automatic transfer? You will receive an SMS o
r a push that the transfer cannot be completed. The next attempt will be repeated the next day. If even then there is not enough money on
the card, the transfer will not be executed. A new attempt will be repeated in the next period according to the schedule.

> source_documents\sber_en.txt:
"Automatic transfers. Why is an automatic transfer not performed? An automatic transfer may not be performed for the following reasons: 1)
there is not enough money on the card to which the service is connected; 2) the debit or receipt card is blocked or its validity period h
as expired; 3) you canceled the operation; 4) on the date of execution of the automatic transfer, the limit of debit transactions on the c
ard has been exceeded."

> source_documents\sber_en.txt:
Autotransfers. How do automatic transfers work? How it works: 1) Connect the service. It is enough to know the card number. Set the frequen
cy: once a week, once a month, once a quarter or once a year. 2) Control operations. Every time on the eve of the transfer, you will rece
ive an SMS with a reminder. You will also find the cancellation code there - if you want to cancel the transfer, send it in the reply. You
can change the amount if you wish. 3) Stay informed. As soon as the money is credited

> source_documents\sber_en.txt:
Credits. How to pay in installments if the payment date falls on a weekend? If your card has the required amount, it will be debited autom
atically on the day off. If the amount is incomplete, attempts to write off will continue. The deadline for depositing money is the first
business day. Until then, no fines or penalties will be charged.

```

Рисунок 5 – Ответ чат-бота на входящий запрос

```

> Question:
What is credit card?

> Answer:
A credit card is a form of credit that allows you to pay for purchases in stores and online around the world, has an interest-free period of up to 3-4 months during which you do not pay interest to the bank, but requires not only a passport, but also proof of permanent income.

> source_documents\sber_en.txt:
How does a credit card work? How a credit card works. Credit cards, when used correctly, are a convenient financial tool. They allow you to pay for purchases in stores and online around the world, have a grace period of up to 3-4 months, during which you do not pay interest to the bank. Unlike a consumer loan, the card does not have to be constantly 'opened' again. Just pay off your grace period debt and use your card to pay for new purchases - the amount of available credit is automatically

> source_documents\sber_en.txt:
Credit cards. What is a mandatory credit card payment? A credit card is a form of credit that has an interest-free period. In order for the interest-free period to be valid for all 120 days, you need to make a mandatory payment every month - 2% of the debt. This amount is used to pay off the total debt. Example. On November 12, you paid with a credit card for a purchase worth 10,000 rubles. The credit card interest-free period always starts on the 1st of each month, not on the day of purchase.

> source_documents\sber_en.txt:
Credit cards. How is a credit card different from a debit card? On a debit card, we use our own money, and on a credit card, we use bank money. You can transfer money and withdraw cash from a credit card only with a commission, and from a debit card - without a commission. The credit card has an interest free period. This is when you use the bank's money for some time and do not pay interest. This is very convenient when you do not have enough personal money for purchases. For example, Sberbank

> source_documents\sber_en.txt:
a credit card requires not only a passport, but also proof of permanent income. you spent 50 thousand at that time - interest will be charged on this amount; if on October 22 you spent another 50 thousand - this is the debt of the new reporting period, it can be repaid without interest until November 22. Unlike a debit card, a credit card requires not only a passport, but also proof of permanent income.

```

Рисунок 6 – Ответ чат-бота на входящий запрос

```

> Question:
What is the percent of SberDeposit?

> Answer:
I don't know.

> source_documents\sber_en.txt:
Contributions for everyone. SberVklad and SberVklad Prime. Replenishment, Without withdrawal, yield per year up to 7.2%, deposit amount from 100,000 ?, deposit term from 1 month. Even more profitable with a SberPrime+ subscription
Contributions for everyone. Savings account. Replenishment, Withdrawal, yield per year up to 6.8%, any account amount, account term is indefinite. Save and manage your money freely.

> source_documents\sber_en.txt:
"List of deposits. What deposits do we have? Deposits in our bank: 1) Best% Online: Rate 9.50%, Term 30 - 1095 days, Amount from 100,000 ?; 2) Savings account Active age: Rate 6.80%, Term from 30 days, Amount from 1,000 ?; 3) Savings account: Rate 6.80%, Term 30 - 1460 days, Amount 3,000-1 million ?; 4) SberDeposit Online: Rate 6.20%, Term 30 - 1095 days. , Amount from 100,000 ?; 5) SberDeposit: Rate 4.95%, Term 30 - 1095 days, Amount from 100,000 ?; 6) Manage + Online: Rate 4.77%, Term 91 - 365

> source_documents\sber_en.txt:
confirming the right to inheritance and your passport of a citizen of the Russian Federation to the bank office and fill out an application for compensation there. For more information, please contact the bank staff.
Rating of the most profitable Sberbank deposits in 2023. Best% Online - rate 9.50% per annum. SberVklad Online - the rate is 7.20% per annum. Pension Plus - rate of 3.67% per annum. Savings account - rate up to 6.80% per annum. SberVklad Online - rate up to 6.20% per annum

> source_documents\sber_en.txt:
SberDeposit and SberDeposit Prime have their own advantages - the client can periodically replenish the account, as a result of which interest will be charged on a large amount. This will increase the benefits and accumulate enough funds. You cannot withdraw money from the account until the expiration date. The interest rate is up to 7.2% per annum. To open a deposit account, you need to deposit at least 100,000 rubles. You can place them for a period from 1 month to 3 years. The maximum rate is

```

Рисунок 7 – Неудачный результат генерации ответа из-за не точной формулировки запроса

```

> Question:
What is rate of deposit Best% Online?

> Answer:
The rate of deposit Best% Online is 9.50%.

> source_documents\sber_en.txt:
is best to navigate by the key rate of the Central Bank. This is the percentage at which the Central Bank lends to banks. If you are offered a deposit rate higher than the key one, the matter may be unclear.

> source_documents\sber_en.txt:
"List of deposits. What deposits do we have? Deposits in our bank: 1) Best% Online: Rate 9.50%, Term 30 - 1095 days, Amount from 100,000 ?; 2) Savings account Active age: Rate 6.80%, Term from 30 days, Amount from 1,000 ?; 3) Savings account: Rate 6.80%, Term 30 - 1460 days, Amount 3,000-1 million ?; 4) SberDeposit Online: Rate 6.20%, Term 30 - 1095 days. , Amount from 100,000 ?; 5) SberDeposit: Rate 4.95%, Term 30 - 1095 days, Amount from 100,000 ?; 6) Manage + Online: Rate 4.77%, Term 91 - 365

> source_documents\sber_en.txt:
the online application. Then you choose from the list of offers the option that you need and specify your parameters: the amount and the term of its placement. The interest rate will be calculated automatically. Do not forget to indicate the account or card from which funds will be debited for placement on deposit. After completing all these steps, click 'Continue', ticking the box to agree to the terms. This operation must also be confirmed with a one-time password from SMS. After that, an

> source_documents\sber_en.txt:
When choosing a deposit, you need to remember that additional conditions may be hidden behind high rates. For example, a requirement to invest several million at once or spend a hundred thousand rubles a month on a bank card. And too high a deposit rate is a common sign of an unscrupulous or problematic financial organization. It can be just scammers or a bank that is breathing its last and is trying to quickly attract depositors' money. Here it is best to navigate by the key rate of the Central

```

Рисунок 8 – Удачный результат генерации ответа при верной формулировке запроса

ЗАКЛЮЧЕНИЕ

В данной работе была представлена полностью локальная реализация банковского чат-бота без необходимости обращения в интернет. Используя мощность и возможности современных библиотек и инструментов, таких как LangChain, Transformers, Chroma и загруженных моделей, которые находятся в открытом доступе, был разработан проект чата с ботом, который позволяет клиентам задавать вопросы и получать на них правильные ответы, обеспечивая полную конфиденциальность данных.

В зависимости от системных характеристик рабочей станции можно использовать различные модели языкового моделирования, от размера которых будет зависеть качество формулирования естественной речи.

Результаты данной работы демонстрируют перспективы и возможности полностью локальной реализации банковского чат-бота. Этот подход открывает новые горизонты в сфере обслуживания клиентов, обеспечивая высокую степень безопасности. Одним из ключевых аспектов этой работы является применимость такого подхода не только в банковской сфере, но и в других отраслях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Руководство по языку программирования Python [Электронный ресурс] // URL: <https://metanit.com/python/tutorial/> (дата обращения: 25.05.2023);
2. Hugging Face. The AI community [Электронный ресурс] // URL: <https://huggingface.co/> (дата обращения: 25.05.2023);
3. How to create a private ChatGPT with your own data? [Электронный ресурс] // URL: <https://medium.com/@imicknl/how-to-create-a-private-chatgpt-with-your-own-data-15754e6378a1> (дата обращения: 25.05.2023).

ПРИЛОЖЕНИЕ

Приложение 1. Программная реализация загрузки и представления

данных в векторном виде

```
#!/usr/bin/env python3
import os
import glob
from typing import List
from dotenv import load_dotenv
from multiprocessing import Pool
from tqdm import tqdm

from langchain.document_loaders import (
    CSVLoader,
    EverNoteLoader,
    PDFMinerLoader,
    TextLoader,
    UnstructuredEmailLoader,
    UnstructuredEPubLoader,
    UnstructuredHTMLLoader,
    UnstructuredMarkdownLoader,
    UnstructuredODTLoader,
    UnstructuredPowerPointLoader,
    UnstructuredWordDocumentLoader,
)

from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.vectorstores import Chroma
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.docstore.document import Document
from constants import CHROMA_SETTINGS

load_dotenv()

# Load environment variables
persist_directory = os.environ.get('PERSIST_DIRECTORY')
source_directory = os.environ.get('SOURCE_DIRECTORY', 'source_documents')
embeddings_model_name = os.environ.get('EMBEDDINGS_MODEL_NAME')
chunk_size = 500
chunk_overlap = 50

# Custom document loaders
class MyElmLoader(UnstructuredEmailLoader):
    """Wrapper to fallback to text/plain when default does not work"""

    def load(self) -> List[Document]:
        """Wrapper adding fallback for elm without html"""
        try:
            try:
                doc = UnstructuredEmailLoader.load(self)
            except ValueError as e:
                if 'text/html content not found in email' in str(e):
```

```

        # Try plain text
        self.unstructured_kwargs["content_source"]="text/plain"
        doc = UnstructuredEmailLoader.load(self)
    else:
        raise
except Exception as e:
    # Add file_path to exception message
    raise type(e)(f"{self.file_path}: {e}") from e

return doc

# Map file extensions to document loaders and their arguments
LOADER_MAPPING = {
    ".csv": (CSVLoader, {}),
    # ".docx": (Docx2txtLoader, {}),
    ".doc": (UnstructuredWordDocumentLoader, {}),
    ".docx": (UnstructuredWordDocumentLoader, {}),
    ".enex": (EverNoteLoader, {}),
    ".eml": (MyElmLoader, {}),
    ".epub": (UnstructuredEPubLoader, {}),
    ".html": (UnstructuredHTMLLoader, {}),
    ".md": (UnstructuredMarkdownLoader, {}),
    ".odt": (UnstructuredODTLoader, {}),
    ".pdf": (PDFMinerLoader, {}),
    ".ppt": (UnstructuredPowerPointLoader, {}),
    ".pptx": (UnstructuredPowerPointLoader, {}),
    ".txt": (TextLoader, {"encoding": "utf8"}),
    # Add more mappings for other file extensions and loaders as needed
}

def load_single_document(file_path: str) -> List[Document]:
    ext = "." + file_path.rsplit(".", 1)[-1]
    if ext in LOADER_MAPPING:
        loader_class, loader_args = LOADER_MAPPING[ext]
        loader = loader_class(file_path, **loader_args)
        return loader.load()

    raise ValueError(f"Unsupported file extension '{ext}'")

def load_documents(source_dir: str, ignored_files: List[str] = []) -> List[Document]:
    """
    Loads all documents from the source documents directory, ignoring specified files
    """
    all_files = []
    for ext in LOADER_MAPPING:
        all_files.extend(
            glob.glob(os.path.join(source_dir, f"**/*{ext}"), recursive=True)
        )
    filtered_files = [file_path for file_path in all_files if file_path not in
                      ignored_files]

    with Pool(processes=os.cpu_count()) as pool:
        results = []

```



```

        with tqdm(total=len(filtered_files), desc='Loading new documents', ncols=80)
as pbar:
    for i, docs in enumerate(pool.imap_unordered(load_single_document,
filtered_files)):
        results.extend(docs)
        pbar.update()

    return results

def process_documents(ignored_files: List[str] = []) -> List[Document]:
    """
    Load documents and split in chunks
    """
    print(f"Loading documents from {source_directory}")
    documents = load_documents(source_directory, ignored_files)
    if not documents:
        print("No new documents to load")
        exit(0)
    print(f"Loaded {len(documents)} new documents from {source_directory}")
    text_splitter = RecursiveCharacterTextSplitter(chunk_size=chunk_size,
chunk_overlap=chunk_overlap)
    texts = text_splitter.split_documents(documents)
    print(f"Split into {len(texts)} chunks of text (max. {chunk_size} tokens each)")
    return texts

def does_vectorstore_exist(persist_directory: str) -> bool:
    """
    Checks if vectorstore exists
    """
    if os.path.exists(os.path.join(persist_directory, 'index')):
        if os.path.exists(os.path.join(persist_directory, 'chroma-
collections.parquet')) and os.path.exists(os.path.join(persist_directory, 'chroma-
embeddings.parquet')):
            list_index_files = glob.glob(os.path.join(persist_directory,
'index/*.bin'))
            list_index_files += glob.glob(os.path.join(persist_directory,
'index/*.pkl'))
            # At least 3 documents are needed in a working vectorstore
            if len(list_index_files) > 3:
                return True
    return False

def main():
    # Create embeddings
    embeddings = HuggingFaceEmbeddings(model_name=embeddings_model_name)

    if does_vectorstore_exist(persist_directory):
        # Update and store locally vectorstore
        print(f"Appending to existing vectorstore at {persist_directory}")
        db = Chroma(persist_directory=persist_directory,
embedding_function=embeddings, client_settings=CHROMA_SETTINGS)
        collection = db.get()
        texts = process_documents([metadata['source'] for metadata in
collection['metadatas']])
        print(f"Creating embeddings. May take some minutes...")
        db.add_documents(texts)

```

```

else:
    # Create and store locally vectorstore
    print("Creating new vectorstore")
    texts = process_documents()
    print(f"Creating embeddings. May take some minutes...")
    db = Chroma.from_documents(texts, embeddings,
persist_directory=persist_directory, client_settings=CHROMA_SETTINGS)
    db.persist()
    db = None

    print(f"Ingestion complete! You can now run privateGPT.py to query your
documents")

if __name__ == "__main__":
    main()

```

Приложение 2. Программная реализация взаимодействия с загруженной языковой моделью

```

#!/usr/bin/env python3
from dotenv import load_dotenv
from langchain.chains import RetrievalQA
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.callbacks.streaming_stdout import StreamingStdOutCallbackHandler
from langchain.vectorstores import Chroma
from langchain.llms import GPT4All, LlamaCpp
import os
import argparse

load_dotenv()

embeddings_model_name = os.environ.get("EMBEDDINGS_MODEL_NAME")
persist_directory = os.environ.get('PERSIST_DIRECTORY')

model_type = os.environ.get('MODEL_TYPE')
model_path = os.environ.get('MODEL_PATH')
model_n_ctx = os.environ.get('MODEL_N_CTX')
target_source_chunks = int(os.environ.get('TARGET_SOURCE_CHUNKS',4))

from constants import CHROMA_SETTINGS

def main():
    # Parse the command line arguments
    args = parse_arguments()
    embeddings = HuggingFaceEmbeddings(model_name=embeddings_model_name)
    db = Chroma(persist_directory=persist_directory, embedding_function=embeddings,
client_settings=CHROMA_SETTINGS)
    retriever = db.as_retriever(search_kwargs={"k": target_source_chunks})
    # activate/deactivate the streaming StdOut callback for LLMs
    callbacks = [] if args.mute_stream else [StreamingStdOutCallbackHandler()]
    # Prepare the LLM
    match model_type:
        case "LlamaCpp":
            llm = LlamaCpp(model_path=model_path, n_ctx=model_n_ctx,
callbacks=callbacks, verbose=False)
        case "GPT4All":
            llm = GPT4All(model=model_path, n_ctx=model_n_ctx, backend='gptj',
callbacks=callbacks, verbose=False)
        case _default:
            print(f"Model {model_type} not supported!")

```

```

        exit;
    qa = RetrievalQA.from_chain_type(llm=llm, chain_type="stuff",
retriever=retriever, return_source_documents= not args.hide_source)
    # Interactive questions and answers
    while True:
        query = input("\nEnter a query: ")
        if query == "exit":
            break

        # Get the answer from the chain
        res = qa(query)
        answer, docs = res['result'], [] if args.hide_source else
res['source_documents']

        # Print the result
        print("\n\n> Question:")
        print(query)
        print("\n> Answer:")
        print(answer)

        # Print the relevant sources used for the answer
        for document in docs:
            print("\n> " + document.metadata["source"] + ":")
            print(document.page_content)

def parse_arguments():
    parser = argparse.ArgumentParser(description='privateGPT: Ask questions to your
documents without an internet connection, '
                                     'using the power of LLMs.')
    parser.add_argument("--hide-source", "-S", action='store_true',
                        help='Use this flag to disable printing of source documents
used for answers.')
    parser.add_argument("--mute-stream", "-M",
                        action='store_true',
                        help='Use this flag to disable the streaming StdOut callback
for LLMs.')

    return parser.parse_args()

if __name__ == "__main__":
    main()

```