**Загрузка документов в GPT-агента: пошаговый процесс**

Для эффективной загрузки документов в GPT-агента (например, OpenAI, GPT API), необходимо подготовить данные в правильном формате и структурировать их для последующей обработки. Вот пошаговый процесс:

**I. Подготовка документов**

**1. Определите формат исходных данных**

* Документы могут быть в формате:
  + **TXT**: предпочтительный формат для обработки текстов.
  + **PDF/DOCX**: их нужно конвертировать в текстовые файлы.
  + **JSON**: используется для структурированных данных.

**2. Преобразуйте документы в текст**

* Используйте Python для извлечения текста из PDF/DOCX:

python

Копировать код

from pdfminer.high\_level import extract\_text

def pdf\_to\_text(pdf\_path, output\_txt\_path):

text = extract\_text(pdf\_path)

with open(output\_txt\_path, "w", encoding="utf-8") as file:

file.write(text)

pdf\_to\_text("document.pdf", "document.txt")

**3. Очистите текст**

* Удалите лишние элементы:
  + Номера страниц.
  + Колонтитулы.
  + Избыточные пробелы.

**4. Разбейте текст на логические блоки**

* Для больших документов разделите текст на части по темам или разделам.

**II. Структурирование данных**

**1. Определите категорию**

* Укажите, к какой категории относится документ:
  + Законодательство.
  + Методические рекомендации.
  + Политики и регламенты.

**2. Добавьте метаданные**

* Каждая запись должна включать:
  + Название документа.
  + Источник.
  + Дата публикации.
  + Версия (при необходимости).

Пример структуры:

json

Копировать код

{

"title": "Федеральный закон № 152-ФЗ",

"category": "Законодательство",

"content": "Текст закона...",

"source": "https://pravo.gov.ru",

"date": "27 июля 2006 г.",

"version": "1.0"

}

**III. Загрузка в GPT-агента**

**1. Использование API**

* OpenAI GPT-3/4 предоставляет интерфейс для передачи данных через API.

Пример кода для загрузки документа:

python

Копировать код

import openai

openai.api\_key = "YOUR\_API\_KEY"

def upload\_document(title, content):

response = openai.Embedding.create(

input=content,

model="text-embedding-ada-002"

)

return response

document\_title = "Федеральный закон № 152-ФЗ"

document\_content = open("document.txt", "r", encoding="utf-8").read()

embedding = upload\_document(document\_title, document\_content)

print("Документ успешно загружен!")

**2. Интеграция с локальной базой знаний**

* Сохраните документы в базе данных (например, PostgreSQL) и добавьте поиск.
* При запросе агент может обращаться к базе для предоставления данных.

Пример хранения в PostgreSQL:

python

Копировать код

import psycopg2

def save\_to\_database(title, content, category):

conn = psycopg2.connect(

dbname="gpt\_agent",

user="user",

password="password",

host="localhost",

port="5432"

)

cursor = conn.cursor()

cursor.execute(

"INSERT INTO documents (title, content, category) VALUES (%s, %s, %s)",

(title, content, category)

)

conn.commit()

conn.close()

save\_to\_database("Федеральный закон № 152-ФЗ", document\_content, "Законодательство")

**IV. Проверка и тестирование**

**1. Проверьте качество загрузки**

* Убедитесь, что текст корректно разделен и загружен.
* Проверьте доступность каждого документа через запросы к агенту.

**2. Тестовые запросы**

Попробуйте задать агенту вопросы по загруженным данным:

* "Что говорит ФЗ-152 о защите персональных данных?"
* "Какие требования указаны в Постановлении № 4088?"

**V. Обновление данных**

**1. Настройте автоматическую загрузку**

* Используйте cron или Планировщик задач Windows для регулярного обновления базы.

**2. Уведомление об изменениях**

* Добавьте функционал оповещения о новых версиях документов:
  + Email.
  + Telegram-бот.

**VI. Пример интеграции для использования**

**1. Запрос информации**

При запросе агент ищет нужные данные в базе и отвечает:

python

Копировать код

def query\_document(query):

# Пример поиска в базе данных

conn = psycopg2.connect(

dbname="gpt\_agent",

user="user",

password="password",

host="localhost",

port="5432"

)

cursor = conn.cursor()

cursor.execute(

"SELECT content FROM documents WHERE title ILIKE %s",

(f"%{query}%",)

)

result = cursor.fetchone()

conn.close()

return result[0] if result else "Документ не найден."

print(query\_document("ФЗ-152"))

**Итог**

Система будет:

1. **Автоматически загружать** нормативные акты в формате TXT/JSON.
2. **Структурировать данные** с метаданными для удобного поиска.
3. **Обеспечивать доступ** к информации через GPT-агента.

Если потребуется помощь с кодом или настройкой, готов детализировать! 🚀

4o

**Вы сказали:**

почему нельзя загрузить документ?

**ChatGPT сказал:**

ChatGPT

Если вы сталкиваетесь с проблемой загрузки документа в GPT-агента, это может быть связано с несколькими причинами. Рассмотрим возможные причины и пути их устранения.

**1. Ограничения размера текста**

**Причина:**

* GPT-модели, такие как GPT-4, имеют ограничения на длину текста, который можно обработать за один запрос. Например:
  + GPT-3.5: около 4,096 токенов.
  + GPT-4: до 8,192 токенов (или больше в некоторых версиях).

**Решение:**

* Разбейте текст документа на части, каждая из которых не превышает лимита токенов.

python

Копировать код

def split\_text\_into\_chunks(text, max\_tokens=3000):

words = text.split()

chunks = []

current\_chunk = []

current\_length = 0

for word in words:

current\_length += len(word) + 1 # Считаем длину слова + пробел

if current\_length > max\_tokens:

chunks.append(" ".join(current\_chunk))

current\_chunk = []

current\_length = len(word) + 1

current\_chunk.append(word)

chunks.append(" ".join(current\_chunk)) # Добавляем последний кусок

return chunks

**2. Ограничения формата**

**Причина:**

* Документ может быть в неподдерживаемом формате (например, PDF или DOCX).
* API GPT принимает только текст, JSON или аналогичные форматы.

**Решение:**

* Преобразуйте документ в текстовый формат (TXT):

python

Копировать код

from pdfminer.high\_level import extract\_text

def pdf\_to\_text(pdf\_path):

return extract\_text(pdf\_path)

text = pdf\_to\_text("document.pdf")

print(text)

**3. Проблемы с подключением к API**

**Причина:**

* Ошибка в конфигурации API, например:
  + Неверный API-ключ.
  + Проблемы с подключением к серверу.

**Решение:**

* Проверьте ваш API-ключ:

python

Копировать код

import openai

openai.api\_key = "YOUR\_API\_KEY"

* Убедитесь, что доступ к сети не ограничен, и проверьте, нет ли блокировок на уровне брандмауэра.

**4. Ограничения в базе данных**

**Причина:**

* Если используется база данных для хранения документов, она может не поддерживать загрузку больших текстов.
* Например, ограничение на размер записи в таблице.

**Решение:**

* Убедитесь, что поле в базе данных имеет тип TEXT или LONGTEXT (в PostgreSQL — TEXT, в MySQL — LONGTEXT).
* Для больших документов используйте их разбиение.

**5. Проблемы с обработкой текста**

**Причина:**

* Документ может содержать сложные или неподходящие для модели структуры:
  + Таблицы, графики.
  + Необработанный HTML-код.

**Решение:**

* Удалите нестандартные элементы или конвертируйте их в текст:

python

Копировать код

from bs4 import BeautifulSoup

def clean\_html(html\_content):

soup = BeautifulSoup(html\_content, "html.parser")

return soup.get\_text()

cleaned\_text = clean\_html("<h1>Заголовок</h1><p>Содержимое.</p>")

print(cleaned\_text)

**6. Ограничения на использование модели**

**Причина:**

* Некоторые модели или версии GPT (особенно бесплатные) могут иметь ограничения на загрузку и обучение на больших объемах данных.

**Решение:**

* Убедитесь, что вы используете подходящую модель (например, GPT-4).
* Если документ слишком большой, настройте свой сервер с библиотеками, такими как langchain для управления длинными текстами.

**7. Лимиты запросов к API**

**Причина:**

* Превышение лимита запросов, установленного для вашего аккаунта.

**Решение:**

* Проверьте ваш тарифный план на OpenAI или другом провайдере.
* Если лимит достигнут, подождите или обновите тариф.

**8. Ошибки кодировки**

**Причина:**

* Документ может содержать некорректные символы или быть в неподдерживаемой кодировке.

**Решение:**

* Перекодируйте текст в UTF-8:

python

Копировать код

with open("document.txt", "r", encoding="utf-8") as file:

text = file.read()

**9. Ошибки интеграции**

**Причина:**

* Неверная логика обработки текста в коде (например, пропуск важных частей).

**Решение:**

* Проверьте весь рабочий процесс:
  + Конвертация.
  + Очистка текста.
  + Загрузка в API.

**Диагностический чек-лист**

1. **Формат документа**:
   * Преобразован ли он в поддерживаемый текстовый формат?
2. **Размер текста**:
   * Уложился ли он в лимиты токенов?
3. **Настройка API**:
   * Правильно ли указан API-ключ?
4. **Текстовые ошибки**:
   * Нет ли в тексте некорректных символов?

Если вы хотите, могу помочь с диагностикой или исправлением проблемы