Использование ChatGPT API в приложениях

Понять как работают Large language Models (LLM)

# Приветствие

Всем привет.

Меня зовут Константин Волошенко, я бизнес - аналитик из Санкт-Петербурга.

Одно из моих увлечений — нейронные сети и искусственный интелект.

Сегодня я расскажу о том, как вы можете интегрировать ChatGPT в свое приложение.

Вступление

# Я представлю обзор того, что такое ChatGPT и как она работает, а затем объясню, как вы можете использовать API ChatGPT для включения его в свое программное обеспечение.

# Что такое и как работает ChatGPT

ChatGPT — это языковая модель, разработанная OpenAI. Она использует метод, называемый Generative Pre-trained Transformer (GPT) для генерации текстовых ответов, которые напоминают человеческую речь. Эта модель была обучена на большом количестве текста из Интернета, что позволило ей изучить шаблоны, грамматику и даже некоторое понимание различных тем. Она работает в диалоговом режиме, когда пользователи могут вводить вопросы, а ChatGPT генерирует соответствующий ответ.

# Шаг 1. Простой запрос. Использование ChatGPT API в приложениях

Let me explain how you can interact with GPT through the API. First, we will send a message to the model and ask for a response. ChatGPT uses the concept of "roles" to represent different participants in the conversation. Each message in the messages list is like a dictionary with two parts: role and content. Role can be one of three values: "system", "user", or "assistant", which tells us who is speaking the message. Content contains the actual message. In a typical chat, there are three main roles:

Как взаимодействовать с ChatGPT через API? ChatGPT использует концепцию "ролей" для различных участников в диалоге. Каждое сообщение в списке messages является словарем, имеющим два поля: role и content. Role может принимать одно из трех значений: "system", "user" или "assistant", указывая, кто именно "говорит" сообщение. Content содержит само сообщение.

В стандартном случае, три основные роли могут быть включены в чат:

* "**system**": Системная роль используется для установки контекста диалога. Это как общая инструкция для модели, которая определяет её поведение во время диалога. Сообщения от "system" обычно помещаются в начале диалога и могут содержать информацию, как модель должна вести себя в роли ассистента.
* "**user**": Это роль для человека- пользователя, который ведет диалог с моделью. Ваши запросы к модели будут выставлены под этой ролью.
* "**assistant**": Это роль для модели AI, которая отвечает на запросы от пользователя.

См. более подробно здесь:

<https://platform.openai.com/docs/guides/gpt/chat-completions-ap>

# Шаг 2. Промт и База Знаний

Чтобы модель смогла адекватно отвечать на вопросы клиентов о вашем предприятии, необходимо подготовить Промт и Базу знаний.

## Промт

При создании промптов (инструкций) для модели ChatGPT есть несколько принципов, которые могут помочь вам получить лучшие результаты:

* **Конкретизация**: Чем конкретнее ваш промпт, тем более конкретный ответ вы получите. Ваши инструкции должны быть понятными и конкретными. Если инструкции слишком общие или неоднозначные, модель может не понять, что вы от нее хотите.
* Роль **system**: Использование роли **system** для указания поведения модели может быть очень полезным. Например, вы можете указать в system: "Вы - помощник, который специализируется на истории искусства и всегда стремится предоставить подробные и точные ответы". Это сообщение задает контекст для модели. Роль **system** позволяет установить задачу или дать инструкции модели. Это сообщение видит модель, но оно не считается частью диалога с пользователем.

## База знаний

Структура вашей Базы Знаний играет жизненно важную роль в предоставлении ответов на запросы пользователей. Основной целью структуры базы знаний является обеспечение ее эффективного функционирования.

Вот несколько ключевых принципов, которые следует учесть при формировании структуры базы знаний:

* **Категоризация и классификация**: База знаний должна быть разделена на ясные и логически организованные блоки, соответствующие различным областям знаний или типам вопросов клиентов. Например, это могут быть разделы, связанные с продуктами, услугами, поддержкой, вопросами оплаты и т.д. В каждом блоке могут быть подблоки для дополнительной детализации. Это позволит LangChain легко найти релевантные чанки, соответствующие запросу.
* **Иерархия**: База знаний может быть организована в виде иерархической структуры, где более общие темы располагаются на верхнем уровне, а более специфические вопросы и ответы находятся на более низких уровнях. Так проще контролировать наполненность базы знаний нужной информацией и обновлять ее при необходимости.

# Step 3: TG bot and all together

Данная диаграмма описывает процесс интеграции Телеграм-бота и ChatGPT для предоставления ответов на вопросы пользователей.

1. Приложение загружает файл с Промтом и файл с Базой Знаний из Google Drive.
2. Приложение разбивает Базу Знаний на чанки с помощью CharacterTextSplitter.
3. Приложение создает индексную базу данных (Индексная DB) на основе чанков Базы Знаний.
4. Индексная база данных преобразует чанки Базы Знаний в эмбеддинги для дальнейшей обработки.
5. Далее асинхронно:.
   1. Пользователь отправляет вопрос Telegram Bot.
   2. Приложение получает вопрос пользователя.
   3. Приложение ищет соответствующие фрагменты в базе знаний на основе вопроса пользователя.
   4. Приложение отправляет запрос в ChatGPT, включая промт, содержимое сообщения (вопрос и соответствующие фрагменты) и параметр температуры, управляющий случайностью ответа.
   5. ChatGPT генерирует ответ и отправляет его обратно в приложение.
   6. В итоге ответ от ChatGPT отправляется обратно пользователю в Телеграмм.

## Bot question examples

who are you?

price?

any discounts?

can I bring alcohol?

What about smoking?

how r u

all your visitors are idiots

you are a fool

what games do you have

que juegos tienes

|  |  |
| --- | --- |
| Question | Type |
| who are you? | General |
| price? | Content |
| any discounts? | Content |
| can I bring alcohol? | Content |
| What about smoking? | Content |
| how r u | General |
| all your visitors are idiots | Stress resistance |
| you are a fool | Stress resistance |
| what games do you have | Content |
| que juegos tienes | Another language |

# Summary

Summary

1. The ChatGPT API is a crucial aspect of the integrated application.
2. It is necessary to have your own Knowledge Base.
3. Telegram bot integration is the easiest approach.
4. Apart from ChatGPT, there are other LLM options available, with the most well-known being the Llama models family.

# Additional materials

## Github reference to NC\_Attic branch

<https://github.com/kvoloshenko/LLMT_01/tree/NC_Attic>

## Structure of the .env file:

TOKEN = '???' # TG bot token

API\_KEY = '???' # Open AI API Key

## Links to openai documentation:

Chat Completions API:

<https://platform.openai.com/docs/guides/gpt/chat-completions-api>

Chat completions response format:

<https://platform.openai.com/docs/guides/gpt/chat-completions-response-format>

## Token Limit in GPT-3.5

see <https://www.scriptbyai.com/token-limit-openai-chatgpt/>

|  |  |
| --- | --- |
| Model | Max Tokens |
| gpt-3.5-turbo | 4086 |
| gpt-3.5-turbo-0613 | 4086 |
| gpt-3.5-turbo-16k | 16384 |

# Meeting info

1. Topic and Agenda

Using ChatGPT API in Applications

Understanding Large language Models (LLM)

* Overview
* Step 1: Simple Request
* Step 2: Prompt and Knowledge base
  + Prompt
  + Knowledge base
* Step 3: TG bot and all together
* Summary
* Q&A

2. Meeting description (not agenda!): In this meeting you will know about/how to....

On the example of ChatGPT you will learn how to integrate the Large Language Model (LLM) into your application.

3. Your NC experience (3-4 sentences)

Previous video <https://bass.netcracker.com/display/VID/Neural+Networks+Overview>

Konstantin Voloshenko, BA from St. Petersburg. One of my hobbies is Artificial Intelligence (AI) and Neural Networks (NN).

4. Follow up materials with links - link to the deep dive into the topic. Bass spaces, articles, useful resources.

My sources <https://github.com/kvoloshenko/LLMT_01/tree/NC_Attic>

Links to Open AI documentation:

Chat Completions API: <https://platform.openai.com/docs/guides/gpt/chat-completions-api>

Chat completions response format: <https://platform.openai.com/docs/guides/gpt/chat-completions-response-format>