Creating Telegram Bots with ChatGPT:

Integration of Telegram bot and ChatGPT

# Prompt

You are a technical writer. Rewrite this text briefly in good technical English. The text is for public speaking. Here is the text:

Hello every one. My name is Konstantin Voloshenko and I’m a BA from St. Petersburg.

One of my hobbies is Neural Networks.

Today I will talk about Neural Networks from three sides: how it looks for a manager, for an analyst and for a developer.

You can see the Agenda.

In addition to the story, today there will be a demonstration of a simple Neural Network solution and everyone will receive a link to the source code of the examples.

# Using ChatGPT API in Applications

# Calling ChatGPT via API. Roles. Influence of the instruction on the response of the model.

## Step 1: Simple Request

Let's see how to interact with GPT through the API.

First, we will pass a message to the model and ask it to generate a response.

ChatGPT uses the concept of "roles" for different participants in the dialogue. Each message in the messages list is a dictionary with two fields: role and content. Role can take one of three values: "system", "user", or "assistant", indicating who is "speaking" the message. Content contains the actual message.

In the standard case, three main roles can be included in the chat:

* "system": The system role is used to set the context of the dialogue. This is like a general instruction for the model, defining its behavior during the dialogue. Messages from "system" are usually placed at the beginning of the dialogue and may contain information on how the model should behave as an assistant.
* "user": This role is for the human user who is engaging in the dialogue with the model. Your requests to the model will be set under this role.
* "assistant": This role is for the AI model that responds to user queries.

When creating a chat with the model, you provide a list of messages. Each message in this list is a dictionary containing two elements: "role" and "content". "Role" indicates the role (mentioned above: "system", "user", or "assistant"), and "content" contains the actual text of the message.

## Step 2: Prompt and Knowledge base

## Prompt

When creating prompts for the ChatGPT model, there are several principles that can help you achieve better results:

* **Specificity:** The more specific your prompt, the more specific answer you will receive. Your instructions should be clear and specific. If the instructions are too general or ambiguous, the model may not understand what you want from it.
* **System message**: Using a system message to indicate the model's behavior can be very useful. For example, you can specify in the system message: "You are an assistant specializing in art history, always striving to provide detailed and accurate answers." This message sets the context for the model. A system message allows you to set tasks or give instructions to the model. The model sees this message, but it is not considered part of the dialogue with the user.
* **Instructions in the user message:** A brief instruction can also be placed in the user message. For example: "Answer the customer's question based on the information provided to you."

## Knowledge base

The structure of the company's knowledge base, which is designed to provide answers to client questions in a text format, is a key aspect of its effective functioning. Remember that chunks most relevant to the user's query will be used for answering questions. Thus, the knowledge base structure should be organized, logical, and convenient for information retrieval in order to ensure maximum usefulness and completeness of information in the chunks presented for analysis. Here are several key principles to consider when forming the knowledge base structure:

* Categorization and classification: The knowledge base should be divided into clear and logically organized blocks corresponding to different knowledge areas or types of client questions. For example, these could be sections related to products, services, support, payment issues, etc. Each block can have sub-blocks for further detail. This will allow LangChain to easily find relevant chunks corresponding to the query.
* Hierarchy: The knowledge base can be organized in a hierarchical structure where broader topics are located at the top level and more specific questions and answers are found at lower levels. This makes it easier to control the relevance of information in the knowledge base and update it as needed.

## Step 3: TG bot and all together

Telegram bot and all together

The diagram illustrates the process of integrating a Telegram Bot and ChatGPT to provide responses to user questions. The diagram shows the flow of information and interactions between different components.

1. The Application downloads two files from Google Drive: one file contains prompt, and the other contains a knowledge base.
2. The Application splits the knowledge base into chunks using a CharacterTextSplitter component.
3. The Application creates an index database and converts the knowledge base fragments into embeddings.
4. The diagram shows a group labeled "async text(update, context) # TG bot function for text messages" which represents the asynchronous process of handling text messages in the Telegram Bot. Within this group, the following steps occur:

* The User sends a question to the Telegram Bot.
* The Application receives the user's question.
* The Application searches for relevant chunks in the knowledge base based on the user's question.
* The Application sends a request to ChatGPT, including the prompt, message content (question and relevant chunks), and a temperature parameter controlling the randomness of the response.
* ChatGPT generates a response and sends it back to the Application.

1. The response from ChatGPT is then sent back