SISTEMAS COMPUTACIONAIS AVANÇADOS (SISTCA) ADVANCED COMPUTING SYSTEMS

Degree in **T**elecommunications and **I**nformatics **E**ngineering

Instituto **S**uperior de **E**ngenharia do **P**orto, **P**olitécnico do **P**orto

Lab Classes Script:
Introdução ao Overleaf e LaTeX

Version	Date	Authors	Update information
V1.0	April 2024	Patrícia Sousa, Carlos Alves, José Leal, Tiago Ribeiro	Original version

Table 1: Version Control

Contents

List of Tables

List of Figures

1 Introduction

1.1 Context

In this article we will present an API developed by openAI, one of the leading organisations in artificial intelligence (AI) research and development. Launched in 2020, it represents a significant milestone in accessing advanced AI technology. It offers an accessible and simplified interface for developers to integrate AI capabilities into a variety of applications, from text analysis to content generation.

The OpenAI API is based on recent advances in machine learning, in particular, deep neural network architectures, such as GPT (Generative Pre-trained Transformer) language models. These models have revolutionised the way machines understand and generate natural language, enabling tasks such as automatic translation, text summarising, question answering, and content generation.

GPT is trained with a vast amount of textual data from the internet, allowing it to capture the nuances and complexities of human language. Through the pre-training process, the model learns to represent knowledge in a general way, without the need for specific training for a particular task.

The OpenAI API leverages these pre-trained language models, providing a simple and effective interface for developers to take advantage of their capabilities. This means users can easily integrate advanced AI capabilities into their applications, without needing to understand complex implementation details or model training.

Additionally, it is designed with a focus on security and ethics, including measures to prevent malicious or harmful use of AI technology. This reflects OpenAI's commitment to promoting the responsible development of artificial intelligence and ensuring that its benefits are extremely accessible and used for the good of society.

1.2 Motivation

One of the main motivations for choosing this topic lies in the possibility of structured presenting a language (and a platform) accessible to all students, providing them with the opportunity to work and apply it in the future, as it is an increasingly used topic day to day. Whether in the SISTCA curricular unit or in other contexts, the OpenAI API offers a versatile and robust environment to explore the potential of artificial intelligence.

When choosing this topic, the need to familiarise yourself not only with the fundamental concepts of artificial intelligence, but also with the practical tools for its implementation, is recognised. This required not only mastering theoretical principles, but also exploring accessible programming languages and platforms such as the OpenAI API.

1.3 Objectives

The main objective of this article is to explore the various features offered by the OpenAI API and learn how to integrate them, promoting a comprehensive and practical understanding of the platform's capabilities, providing users with a solid foundation to explore and use the following available tools (2.2):

- **Chat Completions**, to understand the generic implementation of chat GPT based assistant;
- **Assistants API + Tools** To create a program that provides the user with media suggestions such as books, films or TV shows;
- **Embeddings** Learn how to represent words, sentences, paragraphs, or entire documents in a continuous vector space;
- **Vision** The use of Vision functionality to extract information from an image.
- **Image Generation/DALL-E** Learn how to generate AI created images.
- **TTS (Text to Speech)** Turn strings into a robotised speech using AI;
- **Whisper** Capture an audio and translate it to different languages;
- **Moderation** Explore OpenAI's Moderation functionality to detect inappropriate content.

In addition, two practical exercises will be developed to apply the acquired knowledge and a challenge will be proposed to test the reader's understanding and ability to use the OpenAI API.

1.4 Document Structure

This tutorial is organised into six chapters with the following structure: Introduction, Theoretical, Setup/Installation, Tutorial/Functionality, Exercises and a Challenge.

In the Introduction, some concepts about the OpenAI API will be presented that will be deepened throughout the script.

The theoretical part, will introduce the State-of-the-art and detail the API's Features.

Then, the third chapter focuses on Setup/Installation, explaining how to create an open AI account and set up a development environment.

In the Tutorial/Functionality part, a detailed tutorial of the available features will be provided.

After all these topics, two exercises will be developed and their corresponding resolution will be presented and a final challenge to test the knowledge acquired by the users.

2 Theoretical (scientific/technological background)

Before diving into specific AI models it's important to have a general understanding of how AI generally works.

Most AI models are based on broader **Large language models (LLMs)** these are algorithms that have been trained on vast amounts of data with the purpose of understanding and generating human like text. LLMs make use of the **transformer** type of architecture, a deep learning technique that, in short, represents text via numerical representations known as tokens and gives them different weights so as to be able to contextualise words and find similarities [1].

One example of a transformer based model is the **Generative pre-trained transformer (GPT)**, an AI model that has been pre-trained on large sets of data via the use of the transformer architecture for general purpose tasks. Furthermore, these models can be fine tuned to achieve greater performance in more specific tasks, such is the case of image generation models [2].

Lastly, in order to make use of these models it's essential to understand the concept of prompts. That is the name given to the textual inputs given to the model. These inputs are then broken down into the aforementioned tokens via a process we call tokenization, which facilitates the use of the models language structure.

2.1 State-of-the-art

AI, Artificial Intelligence, refers to a simulation of human intelligence in machines programmed to mimic human cognitive processes and actions. The concept of AI is not new, it has been around since the mid-20th century, but in the last few years there have been significant advances for AI. With these advances, due to its potential, AI has become increasingly important across various industries such as healthcare, finance, manufacturing, education, amongst others. Nowadays, one of the most famous companies developing AI products is OpenAI, which we are going to base our work on. However, just like any other business, OpenAI has its competitors. Some of them currently only dispose of a chat bot, and even that, at the time, are not available in Portugal. Listed below are some companies in the AI business.

AI	**OpenAI[3]**	**X[4]**	**Anthropic[5]**	**Deepmind[6]**	**Cohere[7]**
ChatBot	ChatGPT	Grok	Claude	Gemini	Coral
API	Yes	No	Yes	Yes	Yes
Inputs	Text, Audio, Image		Text	Text, Image	Text
Outputs	Text, Audio, Image		Text	Text	Text
ChatBot and API Availability in Portugal	Both	None	API Only	ChatBot Only	Both

Table 2: AI Companies.

2.2 Features

OpenAI's API offers a vast array of cutting-edge AI models based on deep learning and natural language processing techniques. These models have been trained on vast datasets and fine tuned to fulfil a variety of tasks such as text and image generation, audio and text conversions, amongst other things.

2.2.1 GPT

The GPT (Generative Pre-trained Transformer) series is OpenAI's main set of large language models. These models are trained to understand and generate natural language text based on contextual inputs so as to better communicate with humans. The most widespread version, GPT-3.5 currently serves as the model that powers the free version of ChatGPT. Its understanding of human language allows for coherent conversations which makes it a suitable chat bot. GPT-4 improves upon its predecessor with a smarter and more knowledgeable model that provides greater accuracy across various tasks. In particular, GPT-4 introduces Vision as a new feature, that enables it to process image inputs, making it useful in a wider range of applications.

Function calling

The GPT based models are capable of calling previously specified functions in response to user actions or prompts by calling external APIs to retrieve data or to automate procedures like sending an email or extracting and sorting data from a document.

Assistants

The AI Assistants functionality leverages the use of the GPT models alongside function calling and other tools like file retrieval and code interpreter to allow users to create custom assistants that fulfil more specific tasks based on the provided instructions.

2.2.2 Other models

DALL-E

The DALL-E model is capable of generating images from natural-language text descriptions, as well as modifying existing ones by feeding the model instructions from a text prompt.

TTS

The Text-To-Speech model is capable of converting text into a natural sounding speech. Note that it currently only supports the English language.

Whisper

Whisper does the opposite of the TTS model: it takes an audio input and then transcribes it into text. Unlike text to speech, Whisper is capable of understanding multiple languages, as such it can be used to identify the input language and translate the contents of the speech into English.

Embeddings

Text embeddings are vectorial representations of strings of text, such as words or phrases. By comparing two or more vectors we can infer their similarity. This mechanism is highly useful in applications such as search engines or product recommendations due to it's ability of evaluating similarity between text strings.

Moderation

OpenAI's Moderation model is designed to verify if a certain piece of text includes any content that could be classified as hateful, violent, sexual, harmful or otherwise inappropriate. Whilst OpenAI's own use of the model aims to ensure that content complies with their usage policies [3], this model is suitable for any application that aims to ensure a safe digital environment.

3 Setup/Installation

3.1 Creating an OpenAI Account

In this section,we will guide you through the process of setting up an OpenAI account. Whether you're a developer, researcher, or simply curious about AI, having your own account opens the door to the vast possibilities of artificial intelligence.

Firstly, navigate to the OpenAI website [3] to create or log into your account.

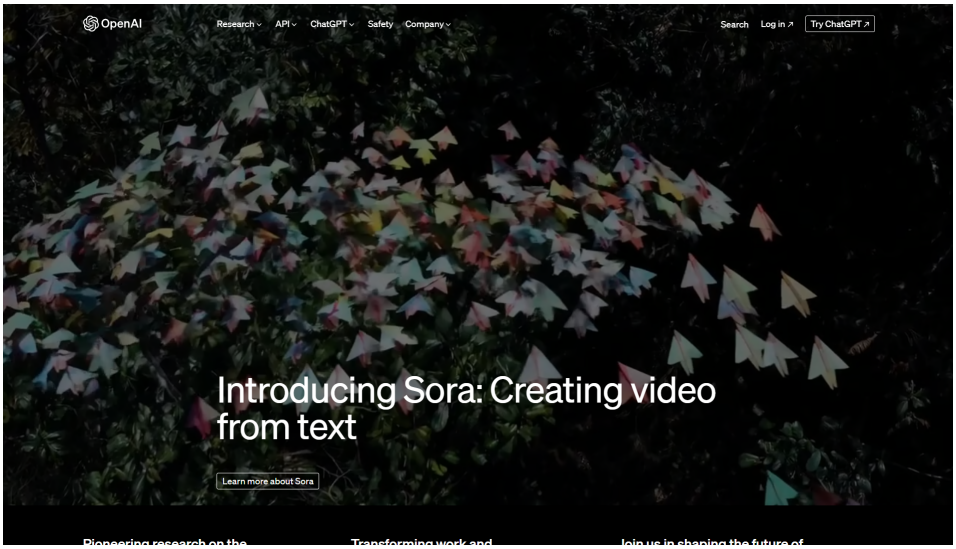

Figure 1: OpenAI Website Access

Upon logging in with your email, you will be presented with two options: ChatGPT and API. Select the API option to access the documentation.

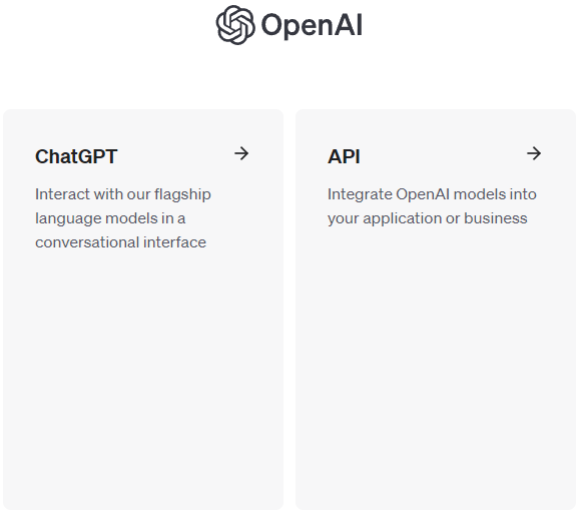

Figure 2: Select the API Option.

Congratulations! You have now successfully created an operational OpenAI account.

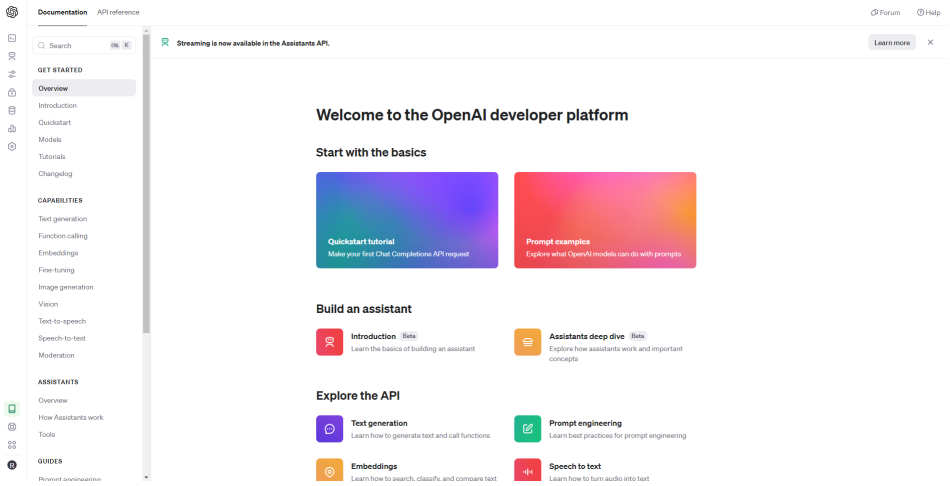

Figure 3: OpenAI Account Creation Confirmation.

3.2 Setting up Your Development Environment

Setting up a proper development environment is crucial for working efficiently with AI applications. Ensure that you have the necessary tools and libraries installed on your system. For most AI development tasks, Python is the recommended programming language due to its extensive ecosystem of AI and machine learning libraries. So, that is exactly what we are going to help you with in this sub-section.

3.2.1 Windows

The first step is accessing python's official website and downloading it. In case you are not sure if you have already installed Python in the past, just type "cmd" in your search-bar and then type "python". If you are having trouble installing, maybe try checking Python's beginners guide [8].

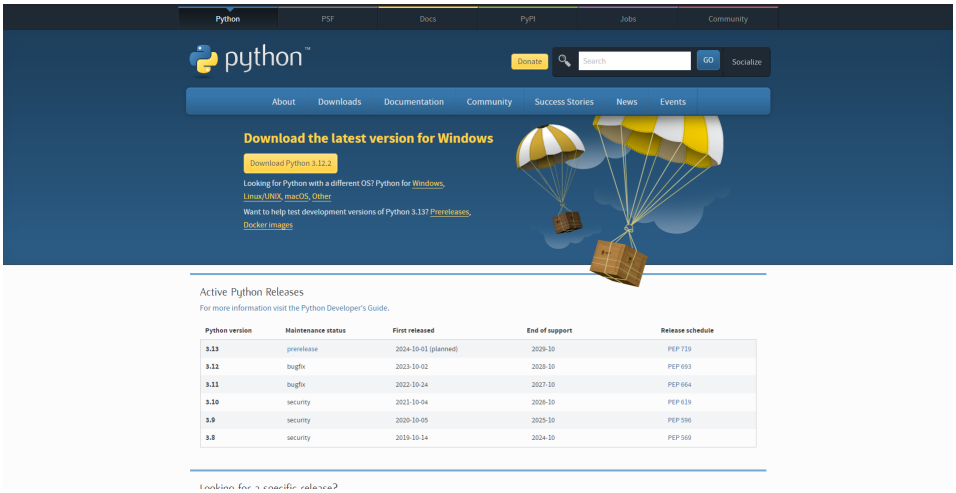

Figure 4: Download Python

Once installed, you are going to create a virtual environment, as it is good practise to avoid conflicts with other installed libraries.

Insert one of the following command in your command prompt:

```
'python -m venv openai-env'

'python3 -m venv openai-env'
```

Now, after creating the virtual environment, you need to activate it:

```
'openai-env\Scripts\aAvailable at: ctivate'
```

After this step you should be able to see "openai-env" to the left of the cursor input section.

Once you have Python installed and (optionally) set up a virtual environment, the OpenAI Python library can be installed. From the command prompt, run:

```
'pip install --upgrade openai'
```

Once this completes, running 'pip list' will show you the Python libraries you have installed in your current environment, which should confirm that the OpenAI Python library was successfully installed.

Now we are going to setup your API key. If you don't have an API key yet then you'll have to follow the instruction in API section

Open your command prompt and then insert the following command:

```
'setx OPENAI_API_KEY "your-api-key-here'
```

In order to make this key setup permanent, you ought to access Environment Variables, for that you just need to search for it in your windows search bar. Click on "New" and then set

`"OPENAI_API_KEY"`

as the variable name and your API key as the value.

3.2.2 Linux

Firstly, open your terminal and introduce the following command in order to download python:

In case you are working on Debian or Ubuntu:

```
'apt install python3 python3-dev'
```

In case you are working on Red Hat, CentOS, or Fedora:

```
'dnf install python3 python3-devel'
```

If you are having trouble installing, maybe try checking Python's beginners guide [8].

Once installed, you are going to create a virtual environment, as it is good practise to avoid conflicts with other installed libraries.

Insert one of the following command in your terminal:

```
'python -m venv openai-env'

'python3 -m venv openai-env'
```

If you can't use none of these commands above because of this error: "The virtual environment was not created successfully because ensurepip is not available" then, try using the following command (after this command you have to re-insert one of the commands above):

```
'sudo apt install python3.10-venv'
```

Now, after creating the virtual environment, you need to activate it:

```
'source openai-env/bin/activate'
```

After this step you should be able to see "openai-env" to the left of the cursor input section.

Once you have Python installed and (optionally) set up a virtual environment, the OpenAI Python library can be installed. From the terminal, run:

```
'pip install --upgrade openai'
```

Once this completes, running 'pip list' will show you the Python libraries you have installed in your current environment, which should confirm that the OpenAI Python library was successfully installed.

Now we are going to setup your API key. If you don't have an API key yet then you'll have to follow the instruction in API section

Go to OpenAI website and access the "API keys" section, there you are going to retrieve your API key or create one in case you do not already have.

Then, open your terminal and type the following command:

```
'export OPENAI_API_KEY='your-api-key-here''
```

To save just press Ctrl+O.

If you want to check it did get setup correctly, type

```
'echo $OPENAI_API_KEY'
```

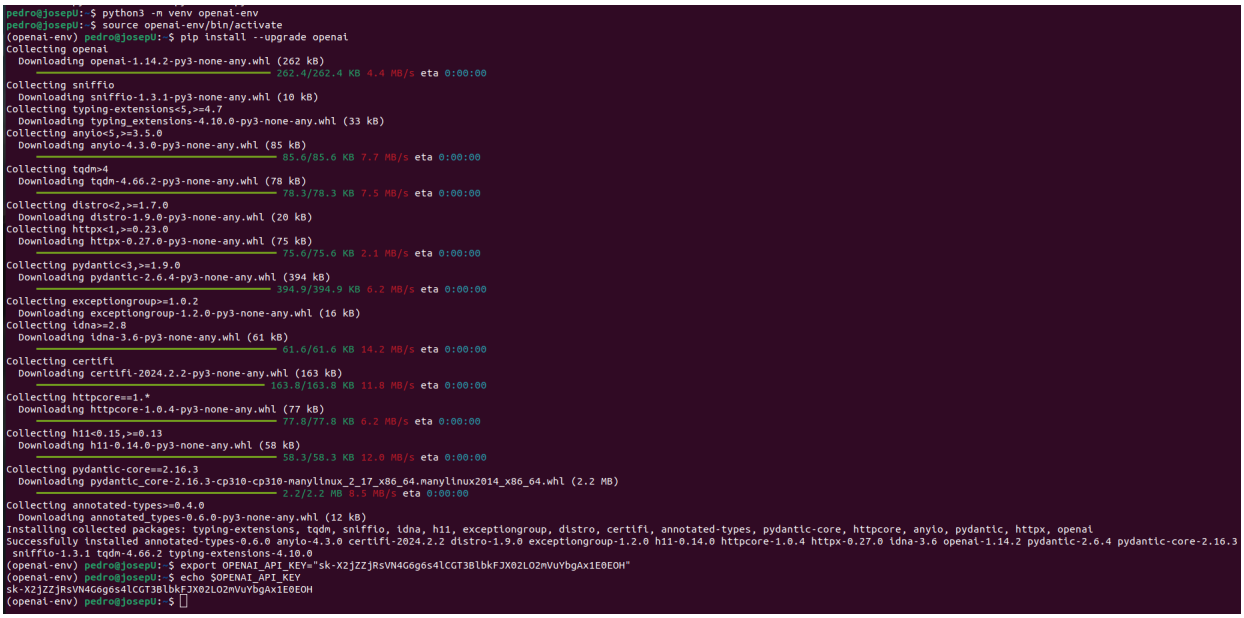

Figure 5: Setup Python

3.2.3 MacOS

Firstly install Brew, if not already installed:

```
'/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/instal
```

Now that you already have Brew, open your terminal and introduce the following command in order to download python:

```
'brew install python'
```

If you are having trouble installing, maybe try checking Python's beginners guide [8].

Once installed, you are going to create a virtual environment, as it is good practise to avoid conflicts with other installed libraries.

Insert one of the following command in your terminal:

```
'pip install virtualenv'
```

Create venv:

```
'virtualenv openai-env'
```

Now, after creating the virtual environment, you need to activate it:

```
'source openai-env/bin/activate'
```

After this step you should be able to see "openai-env" to the left of the cursor input section.

Once you have Python installed and (optionally) set up a virtual environment, the OpenAI Python library can be installed. From the terminal, run:

```
'pip install --upgrade openai'
```

Once this completes, running 'pip list' will show you the Python libraries you have installed in your current environment, which should confirm that the OpenAI Python library was successfully installed.

Now we are going to setup your API key. If you don't have an API key yet then you'll have to follow the instruction in API section

Go to OpenAI website and access the "API keys" section, there you are going to retrieve your API key or create one in case you do not already have.

Then, open your terminal and type the following command:

```
'export OPENAI_API_KEY='your-api-key-here''
```

To save just press Ctrl+O.

If you want to check it did get setup correctly, type

```
'echo $OPENAI_API_KEY'
```

4 Tutorial/Functionality

Listing 1: OpenAI Client

```

from openai import OpenAI
# create an instance of OpenAI, the default construct gets the token from environment
    variable
# use client = OpenAI(api_key = {your_api_token}) if you want to use a different token
client = OpenAI()
```

4.1 Chat Completions

4.2 Assistants

4.3 Embeddings

Listing 2: Embeddings tutorial code

```
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

from openai import OpenAI
client = OpenAI()

# get the embeddings for the first string
response1 = client.embeddings.create(
    input="We are testing to see if this string has any similarities to another one.",
    model="text-embedding-3-small"
)

embeddings1 = response1.data[0].embedding

# get the embeddings for the second string
response2 = client.embeddings.create(
    input="We're experimenting to determine if this string bears resemblance to another.",
    model="text-embedding-3-small"
)

embeddings2 = response2.data[0].embedding

# convert the embeddings to numpy arrays
embeddings1 = np.array(embeddings1).reshape(1, -1)
embeddings2 = np.array(embeddings2).reshape(1, -1)

# computes the cosine similarity between the embeddings of the two strings
similarity_score = cosine_similarity(embeddings1, embeddings2)

final_score = float(format(similarity_score[0][0], ".2f"))

print(f"Similarity: {final_score}")

# determine the level of similarity
if final_score < 0.25:
    print("Totally Different")
elif final_score <= 0.5:
    print("Different")
elif final_score <= 0.75:
    print("Parcial Equal")
```

```
elif final_score <= 0.90:
    print("Almost Equal")
elif final_score <= 1:
    print("Equal")
else:
    print("Unknown value")
```

4.4 Vision

4.5 Image Generation/DALL-E

4.6 TTS

In this tutorial you will learn how to integrate text-to-speech feature from open-ai in your projects.

Available voices	**Output Formats**
Alloy	**mp3** - default format
Echo	**opus**
Fable	**aac**
Onyx	**flac**
Nova	**waw**
Shimmer	**pcm**

Table 3: TTS Information Table

Voice only changes the tone and the "person" who is speaking. TTS only produces english audio files.

Listing 3: TTS tutorial code

```
from pathlib import Path
from openai import OpenAI

client = OpenAI()


### steps needed to generate a random name for the file; this step is not necessarily
    required; it was only made to facilitate testing
import random

animal_names = ["Bunny", "Kitten", "Puppy", "Panda", "Koala", "Otter", "Fawn",
    "Chipmunk", "Hedgehog", "Lamb"]

first_name = random.choice(animal_names)
number = random.randint(1, 9999)

file_name = f"{first_name}{number}"
###

# create a path and a format to save the audio file
speech_file_path = Path(__file__).parent / f"{file_name}.mp3"

# send the api request, change the voice to one of the options available and the input to
    anything you want
response = client.audio.speech.create(
  model="tts-1",
  voice="shimmer",
```

```
25    input="Hey, I'm a student in Lincenciatura de Engenharia de Telecomunicacoes e
          Informatica in Instituto Superior de Engenharia do Porto, and I'm doing a tutorial
          in how to use open-a-i in my projects!"
26  )
27
28  # saves the audio file to specified path
29  response.stream_to_file(speech_file_path)
30
31  print(f"Audio saved under the name {file_name}, {speech_file_path}")
```

4.7 Whisper

4.8 Moderation

5 Exercises

5.1 Exercise A

5.2 Exercise B

6 Challenge

References

[1] IBM, "What is an ai model?," *IBM*. Accessed: 2024-03-28.

[2] R. Merritt, "What is a transformer model?," *NVIDIA Blog*, mar 2022. (Accessed: 2024-03-28.

[3] "Openai." Available at: https://openai.com/. Accessed: 2024-03-27.

[4] xAI, "xai grok." Available at: https://grok.x.ai/. (Accessed: 2024-03-27).

[5] Anthropic, "Anthropic home." Available at: https://www.anthropic.com/. (Accessed: 2024-03-27).

[6] Google, "Deepmind technologies." Available at: https://deepmind.google/technologies/. (Accessed: 2024-03-27).

[7] Cohere, "Cohere home." Available at: https://cohere.com/. (Accessed: 2024-03-27).

[8] "Python." Available at: https://wiki.python.org/. (Accessed: 2024-03-29).