

▼ **Processamento de Linguagem Natural [2023.Q3]**

Prof. Alexandre Donizeti Alves

▼ **ATIVIDADE PRÁTICA 03 [Tutorial sobre uma API]**

A **ATIVIDADE PRÁTICA 03** deve ser feita utilizando o **Google Colab** com uma conta sua vinculada ao Gmail. O link do seu notebook, armazenado no Google Drive, além do link de um repositório no GitHub e os principais resultados da atividade, devem ser enviados usando o seguinte formulário:

<https://forms.gle/nPsWnUuE5GCeXaiU8> (AINDA EM CONSTRUÇÃO)

IMPORTANTE: A submissão deve ser feita APENAS POR UM INTEGRANTE DA EQUIPE até o dia 30/10 (segunda-feira - até às 23h59). Por favor, lembre-se de dar permissão de ACESSO IRRESTRITO para o professor da disciplina de PLN.

▼ **EQUIPE**

POR FAVOR, PREENCHER OS INTEGRANDES DA SUA EQUIPE:

Integrante 01:

Por favor, informe o seu nome completo e RA: Pedro Victor Marcelino Jordão Motta RA: 11201921599

Integrante 02:

Por favor, informe o seu nome completo e RA: Luccas Vinicius de Faveri Tortorelli Cardoso 11201920991

▼ **API**

QUAL A API SELECIONADA POR SUA EQUIPE?

API: Whisper (OpenAI)

Site oficial: <https://platform.openai.com/docs/guides/speech-to-text>

Link para a documentação oficial: <https://platform.openai.com/docs/api-reference/audio/createTranslation>

▼ DESCRIÇÃO

Implementar um notebook no Google Colab para demonstrar passo a passo como usar uma determinada **API** em Python, ou seja, criar um **TUTORIAL** destacando desde a instalação e configuração da **API** até as principais finalidades. O **TUTORIAL** deve ser o mais completo possível, com diversos exemplos de uso e permitindo ser facilmente reproduzido.

A **API** deve ser selecionada na seguinte planilha:

https://docs.google.com/spreadsheets/d/1-Q1szJ3UmoE2_3LtcRQyqid5fPlcnpsR3XAPnoxLj2o/edit?usp=sharing

IMPORTANTE: É obrigatório usar o e-mail da UFABC.

DICA: Por favor, insira o seu nome ou da sua equipe na ordem definida na planilha. A escolha da **API** deve ser feita de acordo com a lista disponibilizada na planilha. Cada **API** só pode ser escolhida por, no **MÁXIMO**, 2 equipes.

ROTEIRO PARA APRESENTAÇÃO DE UMA API

IMPORTANTE: consulta feita no ChatGPT

Como escrever um tutorial sobre uma API em Python usando o Google Colab com foco nas etapas essenciais. Aqui está um guia rápido:

- **Passo 01: Introdução**

- Apresente a API que você está explicando.

- Explique por que é útil ou relevante.

- **Passo 02: Configuração no Google Colab**

- Explique como os leitores podem abrir um novo notebook no Google Colab.

- Mostre como importar bibliotecas necessárias.

- Instrua sobre como instalar quaisquer bibliotecas adicionais, se necessário.

- **Passo 03: Autenticação (se aplicável)**

- Descreva como configurar a autenticação, caso a API exija.

Forneça orientações sobre a obtenção de chaves de API ou tokens.

- **Passo 04: Utilização Básica da API**

Demonstre como fazer uma solicitação simples à API.

Mostre como processar a resposta da API.

- **Passo 05: Exemplos Práticos**

Forneça exemplos práticos e úteis que os leitores podem seguir.

Mostre como aplicar a API em cenários do mundo real.

- **Passo 06: Considerações**

Recapitule os principais pontos do tutorial.

Incentive os leitores a explorar mais a API e experimentar por conta própria.

- **Passo 07: Publicação**

Considere compartilhar seu tutorial em plataformas como Medium, GitHub (obrigatório) ou seu blog pessoal.

- **Passo 08: Promoção**

Compartilhe seu tutorial nas redes sociais, grupos de desenvolvedores e fóruns relevantes.

IMPORTANTE: Lembre-se de manter o tutorial curto e direto ao ponto. Use exemplos de código eficazes e forneça informações práticas que os leitores possam seguir facilmente. O Google Colab é uma ótima plataforma para compartilhar tutoriais, pois permite que os leitores executem o código diretamente no navegador.

▼ CRITÉRIOS DE AVALIAÇÃO

As 3 equipes que **criarem os melhores tutoriais** terão o peso diminuído na AVALIAÇÃO (Prova Escrita) em **25%** (caindo de 40 para 30).

IMPORTANTE: a diminuição no peso da AVALIAÇÃO será aplicado para todos os membros da equipe.

▼ TUTORIAL

INFORMAR O NOME DA API AQUI

Whisper (OpenAI)

SUGESTÃO: Usar o **ROTEIRO** para apresentação de uma **API**

▼ INTRODUÇÃO

Descrição da API

Whisper é um sistema de reconhecimento automático de fala (ASR) treinado em 680.000 horas de dados supervisionados multilíngues e multitarefa coletados da web. Mostramos que o uso de um conjunto de dados tão grande e diversificado leva a uma robustez aprimorada a sotaques, ruído de fundo e linguagem técnica. Além disso, permite a transcrição em vários idiomas, bem como a tradução desses idiomas para o inglês.

O Whisper realiza a tarefa de speech-to-text, além da possibilidade de já realizar a tradução da transcrição para o idioma inglês.

O modelo suporta diversas linguagens. Segue a lista: Afrikaans, Arabic, Armenian, Azerbaijani, Belarusian, Bosnian, Bulgarian, Catalan, Chinese, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, Galician, German, Greek, Hebrew, Hindi, Hungarian, Icelandic, Indonesian, Italian, Japanese, Kannada, Kazakh, Korean, Latvian, Lithuanian, Macedonian, Malay, Marathi, Maori, Nepali, Norwegian, Persian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swahili, Swedish, Tagalog, Tamil, Thai, Turkish, Ukrainian, Urdu, Vietnamese, and Welsh.

Importancia

Os modelos de transcrição de áudio desempenham um papel essencial em diversos contextos. Primeiramente, eles promovem a acessibilidade, tornando conteúdo de áudio compreensível para pessoas com deficiência auditiva. Além disso, facilitam a pesquisa e recuperação de informações em gravações longas, bem como a indexação de vastas bibliotecas de áudio. A transcrição automática também é valiosa para análise de dados, permitindo a compreensão de discursos públicos, mineração de informações em call centers e muito mais.

▼ API

A API de fala para texto fornece dois pontos de extremidade, transcrições e traduções, baseados em nosso modelo Whisper de código aberto de última geração. Eles podem ser

usados para:

Transcrever áudio para qualquer idioma em que o áudio esteja. Traduzir e transcrever o áudio para inglês.

Exemplo de código para cada ponto de extremidade, usando a biblioteca da OpenAI:

```
# transcript = openai.Audio.transcribe("whisper-1", file="openai.mp3")
```

```
# transcript = openai.Audio.translate(model="whisper-1", file="openai.mp3")
```

Parâmetros da de transcrever

- `file`: Objeto do arquivo de áudio (não o nome do arquivo) a ser transcrito, em um dos seguintes formatos: flac, mp3, mp4, mpeg, mpga, m4a, ogg, wav ou webm **(obrigatório)**.
- `model`: ID do modelo a ser usado. Atualmente, apenas `whisper-1` está disponível **(obrigatório)**.
- `language`: Idioma do áudio de entrada. Fornecer o idioma de entrada no formato ISO-639-1 melhorará a precisão e a latência **(opcional)**.
- `prompt`: Texto opcional para guiar o estilo do modelo ou continuar um segmento de áudio anterior. O prompt deve corresponder ao idioma do áudio **(opcional)**.
- `response_format`: Formato da saída da transcrição, em uma das seguintes opções: json, text, srt, verbose_json ou vtt. O padrão é `json` **(opcional)**.
- `temperature`: Temperatura de amostragem, entre 0 e 1. Valores mais altos, como 0,8, tornarão a saída mais aleatória, enquanto valores mais baixos, como 0,2, a tornarão mais focada e determinística. Se definido como 0, o modelo usará a probabilidade de log para aumentar automaticamente a temperatura até que certos limites sejam atingidos **(opcional)**.

Retorno:

O texto transcrito.

É importante notar que os parâmetros `file` e `model` são obrigatórios, enquanto os demais são opcionais.

Parâmetros da API de traduzir

- `file`: Objeto do arquivo de áudio (não o nome do arquivo) a ser traduzido e transcrito, em um dos seguintes formatos: flac, mp3, mp4, mpeg, mpga, m4a, ogg, wav ou webm **(obrigatório)**.
- `model`: ID do modelo a ser usado. Atualmente, apenas `whisper-1` está disponível **(obrigatório)**.

- `prompt` : Texto opcional para guiar o estilo do modelo ou continuar um segmento de áudio anterior. O prompt deve estar em inglês (**opcional**).
- `response_format` : Formato da saída da transcrição, em uma das seguintes opções: json, text, srt, verbose_json ou vtt. O padrão é json (**opcional**).
- `temperature` : Temperatura de amostragem, entre 0 e 1. Valores mais altos, como 0,8, tornarão a saída mais aleatória, enquanto valores mais baixos, como 0,2, a tornarão mais focada e determinística. Se definido como 0, o modelo usará a probabilidade de log para aumentar automaticamente a temperatura até que certos limites sejam atingidos (**opcional**).

Retorno:

O texto traduzido e transcrito.

É importante notar que os parâmetros `file` e `model` são obrigatórios, enquanto os demais são opcionais.

OBS: O sistema de prompt do Whisper é muito mais limitado do que os modelos outros modelos de linguagem e fornece apenas controle limitado sobre a transcrição do áudio gerado.

Corrigir palavras ou siglas específicas que o modelo muitas vezes reconhece erroneamente no áudio. Por exemplo, o prompt a seguir melhora a transcrição das palavras DALL·E e GPT-3, que foram anteriormente reconhecidas como "PIB 3" e "DALI".

▼ CONFIGURAÇÃO

Para a realização da tarefa, é necessário instalar a biblioteca da OpenAI. Através da seguinte célula:

```
# inserir código
!pip install openai
```

```
Collecting openai
  Downloading openai-0.28.1-py3-none-any.whl (76 kB)
    77.0/77.0 kB 1.8 MB/s et
Requirement already satisfied: requests>=2.20 in /usr/local/lib/python3.10/di
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/pyt
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.1
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.1
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /usr/local/lib/
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/di
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/
```

```
Installing collected packages: openai
```

```
ERROR: pip's dependency resolver does not currently take into account all the  
llmx 0.0.15a0 requires cohere, which is not installed.
```

```
llmx 0.0.15a0 requires tiktoken, which is not installed.
```

```
Successfully installed openai-0.28.1
```

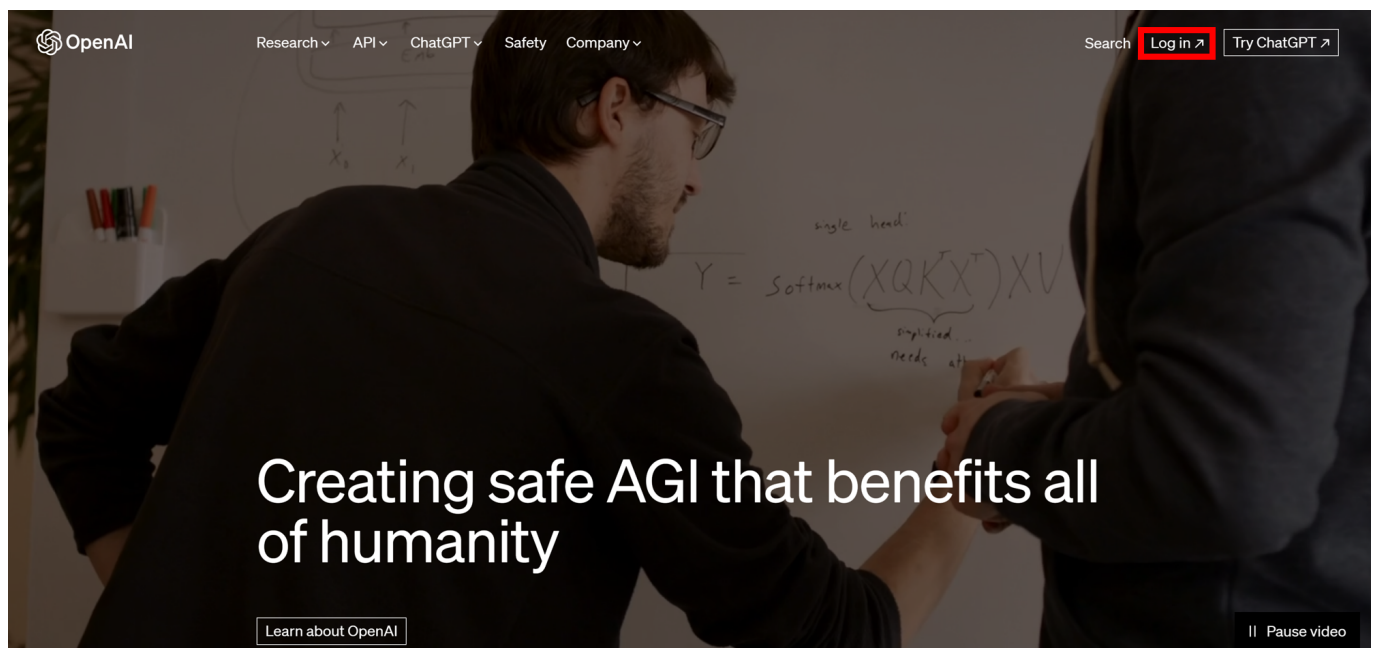
Caso, tiver um alerta de biblioteca faltando, será necessário repetir o processo de instalação alterando para o nome da biblioteca

▼ AUTENTICAÇÃO (SE APLICÁVEL)

Para essa tarefa é necessário ter a API-KEY da OpenAI, portanto, é necessário entrar no site:

<https://openai.com/>

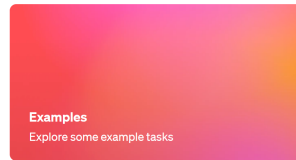
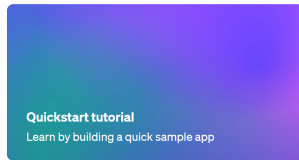
Realizar o cadastro ou login.



Após se cadastrar, você será encaminhado para a seguinte página, e clicará em Upgrade (Localizado canto superior direito)

Welcome to the OpenAI platform

Start with the basics



Build an application



GPT
Learn how to generate text and call functions



GPT best practices
Learn best practices for building with GPT models



Embeddings
Learn how to search, classify, and compare text



Speech to text
Learn how to turn audio into text

Será redirecionado para outra página, que mostrar a quantidade de créditos que contém e clicará em Api Keys.

ORGANIZATION

Personal

Settings

Usage

Rate limits

Members

Billing

Overview

Payment methods

Billing history

Usage limits

Preferences

USER

Settings

API keys

Billing overview

Free trial

Credit remaining

\$5.00

Add payment details

View usage



Note: This does not reflect the status of your ChatGPT account.



Payment methods
Add or change payment method



Billing history
View past and current invoices



Usage limits
Set monthly spend limits



Preferences
Manage company information



Pricing
View pricing and FAQs

Você irá criar uma nova secret key para o serviço

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED ⓘ
Teste	sk-...1AhA	25 de out. de 2023	Never
<div>+ Create new secret key</div>			

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

Escolherá um nome para ela e clicará em "Create secret key":

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED ⓘ
Teste	sk-...1AhA	25 de out. de 2023	Never
<div>+ Create new secret key</div>			

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

Create new secret key

Name Optional

Whisper-Model

Cancel

Create secret key

Será criada nossa secret key, que é equivalente a APIKEY, ela será usada para a autenticação.

OBS: Necessário já copia-la, pois se necessitar dela posteriormente, é necessário criar uma nova

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

Create new secret key

Please save this secret key somewhere safe and accessible. For security reasons, **you won't be able to view it again** through your OpenAI account. If you lose this secret key, you'll need to generate a new one.

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

Done

Default organization

If you belong to multiple organizations, you can select a default organization to use when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

```
import openai
apiKey= "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"
openai.api_key=apiKey
```

▼ UTILIZAÇÃO BÁSICA

Para utilização básica da API, é necessário subir um arquivo de no máximo 25 mb, podendo ter as seguintes extensões: mp3, mp4, mpeg, mpga, m4a, wav webm.

Para realizar qualquer tarefa, vamos primeiro realizar o upload de um arquivo que tenha suporte

```
from google.colab import files
```

Escolher Arquivos Nenhum arquivo escolhido Upload widget is only available when the cell has been executed. Saving É de Mais de 8000 (Official).mp4 to É de Mais de 8000 (Official).mp4

'É de Mais de 8000 (Official).mp4'

'FIGHTING GAME Please, tell me what is the fighting power of Kakarot? It's more than 8000! More than 8000? This must be a mistake! This device must be broken!

- **Criar legendas para vídeos:** O Whisper pode ser usado para criar legendas para vídeos, o que pode tornar o conteúdo acessível a pessoas com deficiência auditiva, já que ele

devolve o tempo da fala.

- **Automatizar a transcrição de áudio:** O Whisper pode ser usado para automatizar a transcrição de áudio, o que pode economizar tempo e recursos.
- **Aprimorar a tradução automática:** O Whisper pode ser usado para aprimorar a tradução automática, o que pode melhorar a precisão e a fluência das traduções.
- **Desenvolver novos produtos e serviços:** O Whisper pode ser usado para desenvolver novos produtos e serviços, como assistentes virtuais, chatbots e aplicativos de tradução.

```
as audio_file= open(f"/content/{name}", "rb")
transcribeVerbose = openai.Audio.transcribe("whisper-1", audio_file, response_format="text")
```

transcribeVerbose

```
<OpenAIObject at 0x786148f38e00> JSON: {
  "task": "transcribe",
  "language": "portuguese",
  "duration": 20.3799991607666,
  "text": "Por favor, diga logo, qual \u00e9 o poder de luta do Kakarot? H\u00e9 de mais de oito mil! Mais de oito mil? Isso deve ser um engano, esse aparelho deve estar quebrado!",
  "segments": [
    {
      "id": 0,
      "seek": 0,
      "start": 0.0,
      "end": 12.0,
      "text": " Por favor, diga logo, qual \u00e9 o poder de luta do Kakarot?",
      "tokens": [
        50364,
        5269,
        2294,
        11,
        2528,
        64,
        9699,
        11,
        4101,
        1136,
        277,
        8152,
        368,
        287,
        12093,
        360,
        36775,
        289,
        310,
        30,
        50964
      ],
      "temperature": 0.0,
      "avg_logprob": -0.41848912835121155,
```

```

    "compression_ratio": 1.2519084215164185,
    "no_speech_prob": 0.1207631379365921
  },
  {
    "id": 1,
    "seek": 0,
    "start": 12.0,
    "end": 13.0,
    "text": " H\u00e3?",
    "tokens": [
      50964,
      389,
      1046,
      30,
      51014
    ],
    "temperature": 0.0,
    "text": " H\u00e3? 0.44040043005101455"
  }
]

```

Aqui estão alguns cenários do mundo real em que o Whisper pode ser aplicado:

- **Educação:** O Whisper pode ser usado para transcrever aulas e palestras, o que pode ajudar os alunos a acompanhar o conteúdo e a criar anotações.
- **Saúde:** O Whisper pode ser usado para transcrever consultas médicas, o que pode ajudar os pacientes a se lembrar das instruções do médico.
- **Negócios:** O Whisper pode ser usado para transcrever reuniões, entrevistas e chamadas telefônicas, o que pode ajudar as empresas a melhorar a comunicação e a tomada de decisões.
- **Entretenimento:** O Whisper pode ser usado para transcrever podcasts, vídeos e outros conteúdos de áudio.

OBS: Podemos também melhorar a performance da transcrição, realizando uma simples pós-processamento usando GPT-4 ou GPT-3.5-Turbo para corrigir o texto transcrito.

▼ CONSIDERAÇÕES

O modelo presente na API, é um modelo OpenSource, disponibilizado no GitHub (<https://github.com/openai/whisper>). Podendo ser baixado para realizar o processamento do áudio de forma local, tendo a saída json_verbose, como saída default.

Além que rodando de forma local, existe outros modelos que estão disponiveis:

Size	Parameters	English-only model	Multilingual model	Required VRAM	Relative speed
tiny	39 M	<code>tiny.en</code>	<code>tiny</code>	~1 GB	~32x
base	74 M	<code>base.en</code>	<code>base</code>	~1 GB	~16x
small	244 M	<code>small.en</code>	<code>small</code>	~2 GB	~6x
medium	769 M	<code>medium.en</code>	<code>medium</code>	~5 GB	~2x
large	1550 M	N/A	<code>large</code>	~10 GB	1x

Segue o código de exemplo, com as instalações de bibliotecas e execução do modelo:

```
!pip install git+https://github.com/openai/whisper.git
```

```
Collecting git+https://github.com/openai/whisper.git
  Cloning https://github.com/openai/whisper.git to /tmp/pip-req-build-hc2l2jo
  Running command git clone --filter=blob:none --quiet https://github.com/openai/whisper.git
  Resolved https://github.com/openai/whisper.git to commit b38a1f20f4b23f3f30
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting triton==2.0.0 (from openai-whisper==20230918)
  Downloading triton-2.0.0-1-cp310-cp310-manylinux2014_x86_64.manylinux2_17_
  63.3/63.3 MB 9.8 MB/s et
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: more-itertools in /usr/local/lib/python3.10/di
Collecting tiktoken==0.3.3 (from openai-whisper==20230918)
  Downloading tiktoken-0.3.3-cp310-cp310-manylinux2_17_x86_64.manylinux2014_
  1.7/1.7 MB 70.4 MB/s eta
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.10/
Requirement already satisfied: requests>=2.26.0 in /usr/local/lib/python3.10/
Requirement already satisfied: cmake in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-pac
Collecting lit (from triton==2.0.0->openai-whisper==20230918)
  Downloading lit-17.0.3.tar.gz (154 kB)
  154.7/154.7 kB 15.9 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in /usr/local/lib/p
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packa
INFO: pip is looking at multiple versions of torch to determine which version
Collecting torch (from openai-whisper==20230918)
  Downloading torch-2.1.0-cp310-cp310-manylinux1_x86_64.whl (670.2 MB)
  670.2/670.2 MB 1.1 MB/s
```

```

Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch->openai-whisper==2023
  Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl
      23.7/23.7 MB 7.9 MB/s et
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch->openai-whisper==20
  Downloading nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.wh
      823.6/823.6 kB 34.1 MB/s
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch->openai-whisper==2023
  Downloading nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl
      14.1/14.1 MB 21.6 MB/s e
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch->openai-whisper==20230918)
  Downloading nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.
      731.7/731.7 MB 1.5 MB/s
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch->openai-whisper==20230918
  Downloading nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410
      410.6/410.6 MB 2.9 MB/s
Collecting nvidia-cufft-cu12==11.0.2.54 (from torch->openai-whisper==20230918
  Downloading nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl (121
      121.6/121.6 MB 8.4 MB/s

```

```
!sudo apt update && sudo apt install ffmpeg
```

```

Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease [3
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:6 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u.4.0+/ubuntu jammy InRe
Get:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Hit:8 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:9 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRe
Get:10 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x8
Hit:11 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:12 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
Get:15 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packag
Fetched 5,711 kB in 4s (1,522 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
29 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ffmpeg is already the newest version (7:4.4.2-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.

```

```
! pip install jiwer
```

```

Collecting jiwer
  Downloading jiwer-3.0.3-py3-none-any.whl (21 kB)
Requirement already satisfied: click<9.0.0,>=8.1.3 in /usr/local/lib/python3.
Collecting rapidfuzz<4,>=3 (from jiwer)
  Downloading rapidfuzz-3.4.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014
      3.2/3.2 MB 11.0 MB/s eta
Installing collected packages: rapidfuzz, jiwer
Successfully installed jiwer-3.0.3 rapidfuzz-3.4.0

```

```

import torch
import os

# Initialize the device
device = "cuda" if torch.cuda.is_available() else "cpu"

import whisper

def transcribe_audio(audio_file, device):
    model = whisper.load_model("base", device=device)

    transcribe = model.transcribe(audio=audio_file)

    return transcribe

def translate(audio_file, device):
    model = whisper.load_model("base", device=device)

    transcribe = model.transcribe(audio=audio_file, task="translate") # Task é um pi

    return transcribe

transcribe_Whisper = transcribe_audio(f"/content/{name}", device)
transcribe_Whisper

/usr/local/lib/python3.10/dist-packages/whisper/transcribe.py:114: UserWarning:
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
{'text': ' O Favore, Figa logo qual é o poder de luta do Kakaroto? Ah, é demais de 8000! Mais de 8000, isso deve ser o engano, esse aparelho deve estar quebrado!',
 'segments': [{'id': 0,
  'seek': 0,
  'start': 0.0,
  'end': 12.0,
  'text': ' O Favore, Figa logo qual é o poder de luta do Kakaroto?',
  'tokens': [50364,
  422,
  479,
  706,
  418,
  11,
  479,
  9900,
  9699,
  4101,
  1136,
  277,
  8152,
  368,
  287,
  12093,
  360,
  36775,

```



```

289,
6738,
30,
50964],
'temperature': 0.0,
'avg_logprob': -0.47896498964543927,
'compression_ratio': 1.2258064516129032,
'no_speech_prob': 0.10448088496923447},
{'id': 1,
'seek': 0,
'start': 12.0,
'end': 15.0,
'text': ' Ah, é demais de 8000!',
'tokens': [50964, 2438, 11, 1136, 36879, 368, 1649, 1360, 0, 51114],
'temperature': 0.0,
'avg_logprob': -0.47896498964543927,
'compression_ratio': 1.2258064516129032,
'no_speech_prob': 0.10448088496923447},
{'id': 2,
'seek': 0,
'start': 15.0,
'end': 20.0,
'text': ' Mais de 8000, isso deve ser o engano, esse aparelho deve estar
quebrado!',
'tokens': [51114,
6313,
368,
1649,
1360,
11

```

```

translate_whisper = translate(f"/content/{name}", device)
translate_whisper

```

```

/usr/local/lib/python3.10/dist-packages/whisper/transcribe.py:114: UserWarning:
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
{'text': " TECHNEDIY Please, just a second What's the power of Kakarot? It's
plus AND EUR! More than thirty ... Patronus !! This should bemam This wall
should be broken",
'segments': [{'id': 0,
'seek': 0,
'start': 0.0,
'end': 2.16,
'text': ' TECHNEDIY',
'tokens': [50364, 19744, 5462, 15988, 3085, 56, 50472],
'temperature': 1.0,
'avg_logprob': -3.2959012349446613,
'compression_ratio': 1.162962962962963,
'no_speech_prob': 0.10448088496923447},
{'id': 1,
'seek': 0,
'start': 8.5,
'end': 9.18,
'text': ' Please, just a second',
'tokens': [50789, 2555, 11, 445, 257, 1150, 50823],
'temperature': 1.0,
'avg_logprob': -3.2959012349446613,
'compression_ratio': 1.162962962962963,
'no_speech_prob': 0.10448088496923447},
{'id': 2,

```

```

'seek': 0,
'start': 9.64,
'end': 11.24,
'text': " What's the power of Kakarot?",
'tokens': [50846, 708, 311, 264, 1347, 295, 36775, 289, 310, 30, 50926],
'temperature': 1.0,
'avg_logprob': -3.2959012349446613,
'compression_ratio': 1.162962962962963,
'no_speech_prob': 0.10448088496923447},
{'id': 3,
'seek': 0,
'start': 12.48,
'end': 14.68,
'text': " It's plus AND EUR!",
'tokens': [50988, 467, 311, 1804, 8093, 462, 7932, 0, 51098],
'temperature': 1.0,
'avg_logprob': -3.2959012349446613,
'compression_ratio': 1.162962962962963,
'no_speech_prob': 0.10448088496923447},
{'id': 4,
'seek': 0,
'start': 15.14,
'end': 16.12,
'text': ' More than thirty ...',
'tokens': [51121, 5048, 813, 11790, 1097, 51170],
'temperature': 1.0,
'avg_logprob': -3.2959012349446613,
'compression_ratio': 1.162962962962963,
'no_speech_prob': 0.10448088496923447},
{'id': 5,
'seek': 0,
'start': 16.56.

```

Outra alternativa é a utilização do faster-whisper (<https://github.com/guillaumekln/faster-whisper>). Que é mesmo modelo de speech-to-text, entretanto, com otimizações em relação ao processamento e ocupação de memória, entretanto, o seu funcionamento ocorre de forma mais otimizada em GPU. Segue exemplo de código:

Size	Parameters	English-only model	Multilingual model	Required VRAM	Relative speed
tiny	39 M	<code>tiny.en</code>	<code>tiny</code>	~1 GB	~32x
base	74 M	<code>base.en</code>	<code>base</code>	~1 GB	~16x
small	244 M	<code>small.en</code>	<code>small</code>	~2 GB	~6x
medium	769 M	<code>medium.en</code>	<code>medium</code>	~5 GB	~2x
large	1550 M	N/A	<code>large</code>	~10 GB	1x

Segue um exemplo de código:

```
!pip install faster-whisper
```

```

Collecting faster-whisper
  Downloading faster_whisper-0.9.0-py3-none-any.whl (1.5 MB)
    _____ 1.5/1.5 MB 7.7 MB/s eta
Collecting av==10.* (from faster-whisper)
  Downloading av-10.0.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64
    _____ 31.0/31.0 MB 16.8 MB/s eta
Collecting ctranslate2<4,>=3.17 (from faster-whisper)
  Downloading ctranslate2-3.20.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2
    _____ 35.7/35.7 MB 7.0 MB/s eta
Collecting huggingface-hub>=0.13 (from faster-whisper)
  Downloading huggingface_hub-0.18.0-py3-none-any.whl (301 kB)
    _____ 302.0/302.0 kB 24.1 MB/s
Collecting tokenizers<0.15,>=0.13 (from faster-whisper)
  Downloading tokenizers-0.14.1-cp310-cp310-manylinux_2_17_x86_64.manylinux20
    _____ 3.8/3.8 MB 63.9 MB/s eta
Collecting onnxruntime<2,>=1.14 (from faster-whisper)
  Downloading onnxruntime-1.16.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2
    _____ 6.2/6.2 MB 72.4 MB/s eta
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: pyyaml<7,>=5.3 in /usr/local/lib/python3.10/di
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/p
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/d
Collecting coloredlogs (from onnxruntime<2,>=1.14->faster-whisper)
  Downloading coloredlogs-15.0.1-py2.py3-none-any.whl (46 kB)
    _____ 46.0/46.0 kB 4.9 MB/s eta
Requirement already satisfied: flatbuffers in /usr/local/lib/python3.10/dist-
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packag
Collecting huggingface-hub>=0.13 (from faster-whisper)
  Downloading huggingface_hub-0.17.3-py3-none-any.whl (295 kB)
    _____ 295.0/295.0 kB 27.4 MB/s
Collecting humanfriendly>=9.1 (from coloredlogs->onnxruntime<2,>=1.14->faster
  Downloading humanfriendly-10.0-py2.py3-none-any.whl (86 kB)
    _____ 86.8/86.8 kB 9.0 MB/s eta
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/pyt
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.1
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.1
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist
Installing collected packages: av, humanfriendly, ctranslate2, huggingface-hu
Successfully installed av-10.0.0 coloredlogs-15.0.1 ctranslate2-3.20.0 faster

```

```
from faster_whisper import WhisperModel
```

```
model_size = "large-v2"
```

```
# Run on GPU with FP16
```

```
model = WhisperModel(model_size, device=device)
```

```
segments, info = model.transcribe(f"/content/{name}", beam_size=5)
```

segments

<generator object WhisperModel.generate_segments at 0x786068babf40>

info

```
TranscriptionInfo(language='pt', language_probability=0.9838305711746216,
duration=20.387125, duration_after_vad=20.387125, all_language_probs=[('pt',
0.9838305711746216), ('en', 0.005116687156260014), ('es',
0.003397029358893633), ('fr', 0.001094887382350862), ('it',
0.0009255387121811509), ('ja', 0.0008803860982879996), ('pl',
0.0005957882385700941), ('hu', 0.0005614966503344476), ('ru',
0.0004879726329818368), ('nn', 0.000324992899550125), ('la',
0.0003024408651981503), ('tr', 0.00024573944392614067), ('ko',
0.00022541568614542484), ('zh', 0.00019231077749282122), ('km',
0.00017023677355609834), ('jw', 0.0001362338225590065), ('nl',
0.0001293669774895534), ('ro', 0.00012149565736763179), ('de',
0.00012038576096529141), ('ar', 0.0001194217984448187), ('haw',
0.00011309508408885449), ('cs', 0.00010054223821498454), ('el',
8.512037311447784e-05), ('vi', 8.442654507234693e-05), ('id',
7.050685962894931e-05), ('tl', 5.810270886286162e-05), ('hi',
4.245387754053809e-05), ('th', 4.033182995044626e-05), ('ca',
3.8202932046260685e-05), ('sv', 3.8103633414721116e-05), ('cy',
3.672017192002386e-05), ('fi', 3.337522502988577e-05), ('ms',
3.186441972502507e-05), ('gl', 2.7722913728212006e-05), ('sn',
2.128709456883371e-05), ('br', 2.071251583402045e-05), ('uk',
2.045971450570505e-05), ('ur', 1.4761432794330176e-05), ('hr',
1.287070699618198e-05), ('mi', 1.1576297765714116e-05), ('fa',
1.1485871254990343e-05), ('te', 9.895236871670932e-06), ('he',
7.289057521120412e-06), ('lt', 5.906310434511397e-06), ('da',
5.5236564548977185e-06), ('bn', 5.5160021474875975e-06), ('bg',
5.368225629354129e-06), ('yo', 5.215517830947647e-06), ('no',
5.196556685405085e-06), ('eu', 5.0853464017563965e-06), ('oc',
4.628066562872846e-06), ('sk', 4.4267158045840915e-06), ('ta',
4.317301772971405e-06), ('bs', 3.7794582112837816e-06), ('si',
3.4356512514932547e-06), ('ml', 3.07831783175061e-06), ('sa',
2.9888738026784267e-06), ('pa', 2.746490281424485e-06), ('sl',
2.32317324844189e-06), ('sr', 2.269685865030624e-06), ('ht',
2.1150192424101988e-06), ('sd', 2.0870975276920944e-06), ('lv',
1.8883806660596747e-06), ('sw', 1.8450498373567825e-06), ('sq',
1.512754579380271e-06), ('ne', 1.4728225323779043e-06), ('my',
1.3163661378712277e-06), ('yi', 1.079234834833187e-06), ('hy',
8.032669143176463e-07), ('fo', 6.9449504280783e-07), ('mn',
4.46928936526092e-07), ('be', 4.203332650831726e-07), ('ln',
3.6327929819890414e-07), ('af', 3.3958983181037183e-07), ('is',
2.93251332550426e-07), ('az', 2.6718669232650427e-07), ('gu',
2.611666047869221e-07), ('kn', 2.4548657506784366e-07), ('bo',
2.3153000938691548e-07), ('lo', 2.1190659538206091e-07), ('kk',
1.8423638437070622e-07), ('ps', 1.7491777271061437e-07), ('et',
1.2513977765138407e-07), ('as', 1.0545988260446393e-07), ('su',
1.0342011336206269e-07), ('mk', 8.001112661304433e-08), ('mr',
7.883228647642682e-08), ('so', 5.4123777459835765e-08), ('mt',
4.627201022344707e-08), ('ka', 4.347798565618177e-08), ('ha',
2.8410466867967443e-08), ('am', 1.8256928413507012e-08), ('tg',
1.662502313592995e-08), ('tt', 9.058768846159637e-09), ('lb',
6.469788260687892e-09), ('uz', 8.310030996661055e-10), ('mg',
4.218682903012194e-10), ('tk', 1.100970530021339e-11), ('ba',
7.107260960315109e-12)],
transcription_options=TranscriptionOptions(beam_size=5, best_of=5,
```

```
patience=1, length_penalty=1, repetition_penalty=1, no_repeat_ngram_size=0,
log_prob_threshold=-1.0, no_speech_threshold=0.6,
compression_ratio_threshold=2.4, condition_on_previous_text=True,
prompt_reset_on_temperature=0.5, temperatures=[0.0, 0.2, 0.4, 0.6, 0.8, 1.0],
initial_prompt=None, prefix=None, suppress_blank=True, suppress_tokens=[-1],

print("Detected language '%s' with probability %f" % (info.language, info.language.

for segment in segments:
    print("[%2fs -> %2fs] %s" % (segment.start, segment.end, segment.text))

Detected language 'pt' with probability 0.983831
[0.00s -> 12.00s] Por favor, diga logo, qual é o poder de luta do Kakarot?
[12.00s -> 13.00s] Hã?
[13.00s -> 15.80s] É de mais de oito mil!
[15.80s -> 17.40s] Mais de oito mil?
[17.40s -> 20.28s] Isso deve ser um engano, esse aparelho deve estar quebrad
```

REFERÊNCIAS

Informar todas as referências usadas para a elaboração do **TUTORIAL**. Citar, por exemplo: vídeos no **YouTube**, artigos no **Medium** etc.

IMPORTANTE: considerar referências **RECENTES** (3 meses).