

Vocês não precisam mais pagar por geradores de vozes pessoal, porque agora podem gerar vozes como a minha, a de Papai Noel, e de graça no próprio computador.

O que vocês acabaram de ouvir é um áudio gerado por um modelo de ar que tá chamando muita atenção no mercado que rodou inteiramente aqui da minha máquina.

O nome dele é Cocoro Texto Speitch e ele foi publicado há poucos meses atrás, mas já aparece como top um aqui na categoria modelos de tex to speech do Hugin Face.

E ele chamou atenção por dois motivos.

Em primeiro lugar, ele é um modelo muito bom.

Então, esse aqui é um ranking que avalia diferentes tipos de modelos de geração de de áudio, de voz com IA.

São os chamados TTS, Tex to speitch.

E reparem que o cocoro ele aparece aqui em oitavo colocado, só que todos os outros modelos que estão anteriormente são modelos proprietários fechados, que a gente não tem acesso ao código fonte.

Então ele é o melhor modelo gratuito do mundo no momento.

Mas a parte mais legal dele tá aqui, ó.

Ele é um modelo pequeno, ele tem apenas, entre aspas, né, 82 milhões de parâmetros.

E caso vocês não tenham familiaridade, isso aqui é considerado pouco, tá?

Tão pouco que eu consigo rodar o modelo no meu próprio MacBook sem nenhum problema.

Se eu colocasse isso aqui numa GPU, então ele ia voar.

E o modelo também foi treinado para gerar vozes em diferentes línguas.

Então, a gente tem a inglês americano, inglês britânico, japonês, mandarim, espanhol, português aqui embaixo.

E a gente também tem acesso a diferentes vozes.

Então, por exemplo, esse áudio que foi gerado aqui são os do português brasileiro com a voz do Santa Claus, do Papai Noel, tá?

Mas a gente também tem vozes femininas como essa.

Oi, eu sou a Dora e esta é minha voz ou Alex.

Oi, eu sou o Alex e esta é minha voz.

E Oi, eu sou o Alex e esta é minha voz.

E nesse vídeo aqui eu vou ensinar vocês a como utilizar o modelo na própria máquina e também instalar essa outra aplicação aqui, o Cocoro Fest API, onde a gente consegue colocar o modelo dentro de um docum servidor qualquer no mundo para ter acesso ao nosso próprio serviço de inferência de geração de voz.

Então bora lá.

Então vou começar aqui com um projeto vazio e eu quero que vocês já instalem o UV, que é um gerenciador de pacotes do Python que tá muito popular ultimamente.

Eu já tenho ele instalado, não vou rodar.

E aí, após isso, a gente vai dar um `UV init` para ele criar aqui alguns arquivos básicos.

Vou apagar esse `main.file`.

Depois eu vou instalar algumas bibliotecas.

Então eu quero colocar o cocoro, Sound File, Open AI e o Numpie.

O UV.

Faço isso tudo muito rapidamente.

E eu vou começar com um projeto bem básico, aonde eu vou passar uma string de texto, ele vai gerar um arquivinho de áudio para eu rodar aqui na minha máquina.

Vou chamar aqui de cocoro Basic.

Então a gente vai começar importando três das bibliotecas que a gente instalou.

Então, `from cocoro` eu importo `k pipeline`, que é uma pipeline para geração de áudio, o `sound file` para poder gerar os arquivos de áudio pra gente e o `numpalhar` com esses dados internamente para converter do que o cocoro gerar aqui para esse arquivo de áudio.

Primeira coisa, eu vou colocar qual que é a língua que nós vamos utilizar.

Eu vou deixar para vocês aqui um pequena pequeno exemplo com outras línguas que vocês poderiam usar.

aqui da própria documentação do cocouro que vocês poderiam encontrar aqui vindo no cocouro, clicar em GitHub, está aqui.

Eu, no caso, como eu vou gerar áudios em português, eu coloquei aqui o P, mas se quisesse gerar em inglês, A de American, F de francês e assim por diante.

E eu vou criar uma instância dessa classe que a gente puxou, que é o pipeline.

Vai ser igual K pipeline com esse language code aqui, tá?

É isso que a gente vai utilizar para fazer as transcrições.

Vou colocar o meu texto aqui, Papai Noel, tá?

Vocês podem colocar o texto que vocês quiserem gerar.

E aqui eu vou rodar esse nosso pipeline, passando o nosso texto como parâmetro e escolhendo uma das vozes.

Para acessar as vozes, vocês podem vir nesse link aqui da própria documentação do Cocoro Norgin Face, vi aqui embaixo, clicar em voices, vai abrir essa página aqui e a gente tem vários exemplos.

Eu vou pegar, começar aqui com a do Santa Claus, que seria o nosso PM Santa.

Isso aqui, galera, é um gerador, tá, do Python.

Então, a gente precisa iterar esse cara para ele ir gerando esses trechos de códigos um a um.

Que que acontece?

O cocoro ele vai rodar linha a linha de maneira separada.

Então ele vai quebrar isso aqui, tá?

Vai rodar essa linha em primeiro.

Quando ele encontrar uma quebra de linha, ele gera a segunda para que a gente consiga processando os textos mais rapidamente.

Mas eu vou pedir para ele processar tudo e todo esse arquivo final eu vou colocar dentro de um arquivo de áudio e salvar.

Como que a gente faz isso?

Muito simples.

Vou criar uma uma lista chamado audio chunks vazia e eu vou colocando os nossos dados aqui dentro.

E para cada vez que eu for iterar sobre esse gerador aqui, ele vai me devolver três elementos.

A gente não precisa se preocupar com os dois primeiros, que são informações sobre o áudio, mas o áudio em si, os valores da do áudio gerado, tá nesse terceiro que ele vai gerar um GS, um PS e um áudio.

Então, para cada um desses em generator, eu quero vir aqui no audio chunks e dar uma pendu áudio.

Simples assim.

Por fim, eu vou pegar o meu áudio completo e concatenar, juntar todos eles e utilizar a biblioteca Sound File para poder criar esse arquivo de áudio completo e utilizar esse conjunto de dados aqui para rodar.

Vamos ver isso na prática funcionando.

Então, a gente viria aqui embaixo, daria UV run cocoro basic.

Ele está processando já.

Aqui faltou o sample rates da do sound file, que é o parâmetro aqui embaixo da taxa de amostragem na hora de geração de dados.

A gente pode utilizar 24.000.

Vou rodar de novo.

A primeira vez que vocês rodam leva um tempinho porque ele tem que baixar os modelos pra máquina local, para esse repositório aqui.

Eita lá, gerou o meu áudio completo.

Vamos ouvir.

Vocês não precisam mais pagar por geradores de vozes pessoal, porque agora podem gerar vozes como a minha, a de Papai Noel.

Funcionou.

E galera, a gente decidiu também criar um repositório aqui no GitHub para vocês, contendo todo um guia mais completo com o projeto que eu apresentei aqui.

Mais um exemplo de como poder fazer o streaming em tempo real do áudio, ao invés de salvar isso no arquivo ponovoov, seria nesse segundo arquivo aqui, tá?

E eu vou deixar o link desse repositório também na descrição desse vídeo para vocês clicarem e darem uma olhadinha, caso precisem de mais, tenham mais alguma dúvida.

Próximo exemplo, eu vou mostrar para vocês agora como é que a gente roda o cocoro fasts.

Que que seria isso?

Os caras desenvolveram um projeto aonde eles colocaram esse modelo dentro de uma de um Docker, tá?

E esse Docker vocês podem colocar em qualquer servidor do mundo, inclusive máquinas que de repente tenham acesso à GPU para montar o teu próprio serviço de inferência e de geração de áudio com o cocoro.

Então isso é muito útil caso vocês de repente ten um computador ruim ou queiram construir um sistema mais robusto, tá?

Para poder fazer essas inferências em tempo real, não querem mais gastar com geradores de áudio como Labs ou da Open AI.

E qual que é a grande vantagem?

Eles colocaram não só dentro de um Docker, mas também dentro de uma de uma de uma API, defesta API, e mimetizaram a forma como a própria Open AI faz para gerar o seu áudio.

Então, quem de repente já tá trabalhando gerando áudios com a Open AI e quiser trocar para essa nova instância, o processo é bem simples, ó.

Basta instanciar aqui a Open AI, colocar o servidor que vocês estão utilizando e rodar alguma coisa.

Eu vou copiar esse trecho de código aqui.

Vamos criar uma um novo aqui chamado cocoro festa api.p.

E eu vou querer agora subir em um outro servidor, numa máquina que eu já tenho.

Como vocês sabem, a gente tem uma máquina específica aqui na Zimov para rodar modelos de linguagem que tem acesso a uma 3090, mas vocês poderiam de repente alugar computadores, servidores no Google Cloud, on serviço de locação de de servidores com placas de vídeo para poder criar o teu próprio servidor que tenha uma capacidade de inferência mais rápida.

O processo vai ser o mesmo.

Então, no meu caso, eu vou entrar lá no nosso servidor, tá?

E o processo via o Cocoro Fest API também é muito simples.

Eu recomendo que vocês cliquem aqui nesse primeiro, ó, quick start.

E ele tem um passo a passo bem rápido aqui, tá?

De simplesmente ou vocês podem rodar isso aqui, caso estejam rodando numa máquina que tem acesso à CPU, ou esse esse segundo, caso vocês tenham uma alguma GPU da Nvidia, tá?

Então, no meu caso, eu tenho, tá?

Eu vou simplesmente vir aqui e rodar isso com sudo.

Essa minha máquina aqui é um Ubuntu server 24.04.

Ao rodar, ele pede meu password porque eu tô usando um sudo.

Ele vai aparecer, vai mostrar para vocês aqui qual que é a porta que ele tá hospedando isso, tá?

Então vai demorar um pouquinho, primeira vez vai demorar mais, vai baixar os modelos, tá?

E ele gerou, tá?

Ele me disse que ele subiu, primeiro lugar uma espécie de front end nessa localização aqui, tá?

que eu vou ter que trocar só o IP pro IP da máquina.

Ó, reparem que ele dá bem bonitinho aqui essa interface, onde eu posso sair gerando o áudio por aqui.

Isso aqui não tá rodando na minha máquina e sim nesse outro servidor, tá?

Posso vir aqui escolher vozes diferentes, posso aumentar a velocidade da fala.

Então, por exemplo, reparem isso, ó.

Vamos botar aqui o geral, por exemplo.

Posso aumentar a velocidade de geração da dela.

E a parte mais interessante é que a gente também pode mesclar vozes diferentes.

Então vamos colocar aqui o aloi.

Vamos botar misturado com a Bela com 0.5.

Ele vai fazer um blend duas vozes, gerar uma voz intermediária pra gente.

Então uma forma super útil da gente tá gerando de repente novas vozes.

Agora eu vou tirar, por exemplo, a loia.

Vamos botar a Bela 100%.

Então, reparem que ele pegou uma voz mais grossa, uma mais fina e ficou uma coisa intermediária, tá?

Isso é uma forma que a cocoro deixa a gente fazer.

Isso aqui também é possível dentro da própria API no Python.

Essa interface aqui é muito legal, dá para baixar os áudios por aqui também, mas não é nisso que eu tô interessado.

E sim nesse nesse serviço aqui, onde eu posso acessar o nosso sistema através do Python.

Então, reparem que aqui ele até colocou, ó, single ou multiple voice voice pack combo.

Ele tá misturando a voz da Sky com a Bela.

Colocou um Hello World.

Vamos rodar isso aqui e ver se funciona.

E a diferença é que aqui eu vou ter que colocar o IP da máquina que eu estou rodando, ó.

Então, se vocês esverem rodando uma máquina da da WS, do Google Cloud, troquem aqui pelo IP externo de lá e garantir tá up.

Vamos de novo dar um new run cocuro.

Agora festa API.

Vamos rodar.

Ele vai gerar aqui.

Ele vai falar inclusive.

Não, não falou.

Ele salvou um arquivo em output.

Vamos ver.

Hello world.

Então, foi só um hello world, só que rodou na minha máquina.

Então é isso.

Isso aqui é mais uma alternativa para vocês construírem agora.

aplicações que com gerando vozes, super útil para trabalhar com de repente assistente por voz ou qualquer outro projeto que vocês queiram embutir em algo que já existe no Python ou mesmo conectar com alguma LLM da voz a modelo de linguagem.

Espero que tenham gostado dos vídeos.

Qualquer dúvida, deixe nos comentários.

Materiais e links aqui na descrição.