20 DE NOVIEMBRE DE 2025

PRÁCTICA 04

SANTIAGO SOUTO ORTEGA
UIE

Tabla de contenido

Introducción

Este proyecto implementa y evalúa dos modelos de síntesis de voz (Text-to-Speech) capaces de clonar voces a partir de muestras de audio breves mediante la técnica de zero-shot voice cloning. Esta técnica permite imitar la voz de una persona a partir de tan solo 3-10 segundos de audio, sin necesidad de realizar costosos entrenamientos o fine-tuning del modelo específicamente con esa voz. El objetivo principal es comparar diferentes TTS y evaluar su capacidad para reproducir el habla de una persona a partir de un audio de referencia, utilizando métricas objetivas para realizar una comparación fundamentada.

Modelos usados

Para este proyecto se eligieron dos modelos que representan diferentes aproximaciones al problema de síntesis de voz.

XTTS

XTTS v2, basado en el modelo ***tts_models/multilingual/multi-dataset/xtts_v2***, utiliza una arquitectura Transformer que combina un encoder de texto para procesar la entrada de texto, un encoder de audio que extrae características del audio de referencia, un decoder autoregresivo que genera el espectrograma mel, y un vocoder HiFi-GAN que convierte el espectrograma en audio de forma de onda. Este modelo es multilingüe, entrenado en 16 idiomas incluyendo, y destaca por su capacidad de zero-shot voice cloning.

YOURTTS

YourTTS, implementado como ***tts_models/multilingual/multi-dataset/your_tts***, utiliza la arquitectura VITS (Variational Inference TTS), que es un modelo end-to-end basado en un encoder de texto con mecanismo de atención, un posterior encoder que procesa el espectrograma mel del audio de referencia, un generador de flujo normalizado para modelar la distribución de audio, un discriminador para entrenamiento adversarial, y un vocoder integrado basado en HiFi-GAN. Este modelo es completamente end-to-end y destaca por su generación rápida debido a su arquitectura de un solo paso, buena calidad de audio con menor latencia, y flexibilidad para generar variaciones del mismo texto.

Elección

La justificación para seleccionar estos dos modelos específicos se fundamenta en varios criterios técnicos importantes. De todos los modelos TTS disponibles que pueden funcionar únicamente con CPU, estos dos presentaron la mejor compatibilidad de dependencias, siendo los que mostraron menos errores durante la instalación y configuración de sus librerías. Muchos otros modelos como Bark, StyleTTS2, o VALL-E X presentaban conflictos de versiones de PyTorch, o requerían dependencias descontinuadas.

Problemas

Durante el desarrollo del proyecto se encontraron varios desafíos técnicos significativos que afectaron el proceso de implementación y requirieron soluciones específicas.

Espacio

El problema más importante fue la gestión de las imágenes Docker, que resultaron extremadamente pesadas causando serios problemas de espacio en disco. La causa de este problema tiene múltiples componentes: las dependencias científicas son extremadamente pesadas, con PyTorch ocupando aproximadamente 800 MB, NumPy, SciPy y scikit-learn añadiendo otros 500 MB adicionales, bibliotecas de audio como librosa, soundfile y audioread sumando 200 MB más, y TTS con sus dependencias alcanzando 1.5 GB, para un total de más de 3 GB solo en dependencias Python. A esto se suma el peso de los modelos pre-entrenados, donde XTTS v2 ocupa 1.8 GB incluyendo encoder, decoder y vocoder, YourTTS 500 MB, y Resemblyzer para extracción de embeddings 100 MB, totalizando aproximadamente 2.4 GB en modelos. La imagen base de Docker con Ubuntu y Python añade otros 400 MB, las herramientas del sistema como ffmpeg y sox otros 300 MB, y las capas intermedias generadas durante la construcción ocupan espacio adicional. El resultado final fue una imagen de aproximadamente 7-8 GB por imagen completa.

Este tamaño excesivo tuvo un impacto en el desarrollo: el disco disponible se llenó haciendo imposible construir nuevas versiones de la imagen o ejecutar contenedores por falta de espacio. Cada build tomaba entre 20 y 30 minutos, lo que dificultaba enormemente la iteración rápida durante el desarrollo y hacía muy complicado probar diferentes configuraciones. Cada cambio requería reconstruir la imagen completa y no se podía mantener múltiples versiones para comparación.

La solución implementada incluyó una limpieza completa del sistema Docker mediante la detención y eliminación de todos los contenedores, la eliminación de todas las imágenes, y una limpieza profunda del sistema usando *docker system prune -a --volumes*, lo que liberó varios GB de espacio ocupado por imágenes antiguas y capas no utilizadas. Finalmente se adoptó un enfoque de desarrollo mixto utilizando desarrollo local con entorno virtual de Python para iteración rápida, reservando Docker solo para validación final.

Dependencias

El segundo problema importante fue la gestión de dependencias, específicamente encontrar modelos TTS compatibles entre sí que funcionaran sin errores en un entorno exclusivamente CPU. Se encontraron conflictos de versiones de PyTorch donde algunos modelos requerían versiones específicas incompatibles entre sí, diferencias entre versión CPU y GPU de PyTorch, y problemas con torchaudio y torchvision. Algunos modelos dependían de bibliotecas ya no mantenidas y había

problemas con versiones antiguas de Python donde algunos modelos requerían Python 3.7 mientras otros necesitaban 3.10 o superior.

Resultados

Para evaluar objetivamente la calidad de los modelos se implementó Speaker Similarity, que permite comparar numéricamente qué tan parecida es la voz sintética a la voz original.

Speaker Similarity mide el grado de similitud entre dos voces desde la perspectiva de las características del hablante, evaluando específicamente si la voz sintética generada "suena como" la voz de referencia original, independientemente de qué palabras se estén diciendo.

La métrica se basa en el concepto de voice embeddings o representaciones vectoriales de la voz. Estos embeddings capturan características espectrales únicas de cada persona. La propiedad fundamental es que dos audios de la misma persona hablando diferentes palabras deberían tener embeddings muy similares, mientras que audios de personas diferentes deberían tener embeddings diferentes.

Una vez obtenidos los dos embeddings, uno del audio original y otro del sintético, se calcula qué tan similares son usando la similitud coseno. Si los vectores apuntan en la misma dirección la similitud es 1 indicando que son idénticos, si los vectores son perpendiculares la similitud es 0 indicando que no están relacionados, y si los vectores apuntan en direcciones opuestas la similitud es -1 indicando que son opuestos. En el contexto de embeddings de voz, una similitud cercana a 1.0 indica que las voces suenan casi idénticas, cercana a 0.5 que son medianamente similares, y cercana a 0.0 o negativa que son muy diferentes.

Para la evaluación se seleccionaron dos audios de referencia con características diferentes para evaluar el rendimiento de los modelos en distintas condiciones acústicas. El primer audio es un anuncio de avión, grabado en ambiente ruidoso con reverberación y presencia de ruido de fondo ambiental típico de una cabina de avión. Este audio fue seleccionado específicamente para evaluar la resistencia al ruido de los modelos.

El segundo audio es una cita del *Club de los Poetas Muertos* que representa una voz limpia y clara sin ruido de fondo significativo, con audio de calidad profesional con entonación expresiva y emocional. Este audio representa el caso ideal para evaluar el rendimiento máximo de los modelos, permite medir la calidad de síntesis en condiciones óptimas, sirve como línea base para comparar con el audio ruidoso, y evalúa la capacidad de los modelos para capturar matices emocionales y de entonación.

Métricas

Audio base: Anuncio avión

```
============================================================
EVALUACIÓN DE SPEAKER SIMILARITY
============================================================
Audio original: data/inference_voice_plane_announcement.wav
Audio sintético: outputs/yourtts/yourtts_output_plane_announcement.wav

Calculando Speaker Similarity (Resemblyzer)...
Cargando VoiceEncoder de Resemblyzer...
Loaded the voice encoder model on cpu in 0.04 seconds.
  Speaker Similarity: 0.5765

============================================================
RESUMEN
============================================================
Speaker Similarity: 0.5765 (objetivo: >0.8)
============================================================

============================================================
EVALUACIÓN DE SPEAKER SIMILARITY
============================================================
Audio original: data/inference_voice_plane_announcement.wav
Audio sintético: outputs/xtts/xtts_output_plane_announcement.wav

Calculando Speaker Similarity (Resemblyzer)...
Cargando VoiceEncoder de Resemblyzer...
Loaded the voice encoder model on cpu in 0.05 seconds.
  Speaker Similarity: 0.8490

============================================================
RESUMEN
============================================================
Speaker Similarity: 0.8490 (objetivo: >0.8)
```

Audio base: Club Poetas Muertos

```
============================================================
EVALUACIÓN DE SPEAKER SIMILARITY
============================================================
Audio original: data/inference_voice_poeta.wav
Audio sintético: outputs/yourtts/yourtts_output_cita_Armstrong.wav

Calculando Speaker Similarity (Resemblyzer)...
Cargando VoiceEncoder de Resemblyzer...
Loaded the voice encoder model on cpu in 0.05 seconds.
  Speaker Similarity: 0.7920

============================================================
RESUMEN
============================================================
Speaker Similarity: 0.7920 (objetivo: >0.8)
============================================================

============================================================
EVALUACIÓN DE SPEAKER SIMILARITY
============================================================
Audio original: data/inference_voice_poeta.wav
Audio sintético: outputs/xtts/xtts_output_cita_Armstrong.wav

Calculando Speaker Similarity (Resemblyzer)...
Cargando VoiceEncoder de Resemblyzer...
Loaded the voice encoder model on cpu in 0.05 seconds.
  Speaker Similarity: 0.8460

============================================================
RESUMEN
============================================================
Speaker Similarity: 0.8460 (objetivo: >0.8)
============================================================
```

Conclusiones

XTTS v2 demostró un rendimiento consistentemente superior en ambas condiciones de prueba, alcanzando valores de Speaker Similarity superiores a 0.84 tanto en el audio con ruido ambiental como en el audio limpio, cumpliendo y superando el objetivo establecido de 0.80 que indica una clonación excelente de la voz. YourTTS mostró un rendimiento inferior en ambos escenarios, siendo especialmente notable su baja similitud en el audio con ruido ambiental donde obtuvo solo 0.5765, lo que indica mayor sensibilidad al ruido de fondo y menor capacidad para extraer características vocales relevantes en condiciones adversas.

La diferencia de rendimiento entre ambos modelos fue más pronunciada en condiciones adversas con ruido, donde XTTS v2 demostró una robustez significativamente mayor para extraer características vocales relevantes ignorando el ruido de fondo. Incluso en condiciones óptimas con audio limpio, XTTS v2 mantuvo una ventaja considerable con una diferencia de aproximadamente 0.054 puntos en la métrica de similitud, lo que se traduce en una clonación notablemente más precisa y fiel a la voz original.

Referencias

- Audio: We Read And write.wav por ajwphotographic, usado bajo Licencia Creative Commons Attribution 3.0.
- Audio: Plane Flight Safety Announcement (part 1).wav por ajwphotographic, usado bajo Licencia Creative Commons Attribution 3.0.