Universidad Intercontinental de la Empresa

Asignatura

Sistemas Interactivos Inteligentes

Práctica 3

Unidad V

TTS

Profesor: David Rivas Villar

Contenido

1 Objetivo de la actividad

En esta práctica los estudiantes, trabajando individualmente, deberán usar modelos TTS (en concreto, la parte de modelos acústicos) para realizar zero-shot voice cloning.

Zero-shot voice cloning permite imitar la voz de una persona a partir de tan solo unos segundos de audio, sin necesidad de realizar costosos entrenamientos o fine-tuning del modelo. Así, a partir del breve ejemplo original, el model genera una voz sintética capaz de aproximarse a las características de la voz original.

Los alumnos deberán escoger un audio que contenga una voz y, mediante modelos de TTS, recrearla diciendo otras palabras totalmente diferentes. Los alumnos deben, como mínimo, comparar dos modelos acústicos diferentes dentro de pipelines TTS, tanto de manera personal y subjetiva como con métricas, siendo un requisito imprescindible usar, al menos, una métrica.

Para esta práctica no se entrega un entorno Docker debido a la libertad que tienen los alumnos para seleccionar modelos y/o métricas.

La entrega será, igual que en las anteriores, en formato Docker y estará acompañada, además del código fuente y desarrollos pertinentes, del dataset de imágenes y descripciones, así como de una **memoria**. En esta los alumnos detallarán las decisiones de implementación y harán un análisis de los resultados obtenidos.

2 Resultados de aprendizaje relacionados

RA02	Explicar los principios, beneficios y desafíos asociados al diseño de sistemas de interacción multimodal.
RA03	Analizar e implementar arquitecturas de software y hardware que permiten la integración de múltiples modalidades de interacción en base a los tipos de entradas y salidas requeridas para cada problema
RA08	Utilizar herramientas y bibliotecas de software especializadas en reconocimiento de emociones, análisis facial, síntesis de voz, etc.

3 Descripción de la actividad

Usando un snippet de audio, los alumnos deberán clonar esa voz, haciendo que diga palabras, frases, etc. no presentes en el audio original.

3.1 Selección de la voz

Es importante que en la selección del audio que contenga la voz se preste atención a cuestiones éticas. Es por esto que los alumnos deberán usar la voz de alguien que la haya cedido para este u otros propósitos similares. Esto es, no deberán usar la voz de profesores, compañeros, amigos, etc. si estos no están de acuerdo en este uso. Podrán usar su propia voz, snippets de datasets o cualquier otra fuente de voz que no presente problemas morales o éticos.

Adicionalmente, el audio usado como plantilla y el audio generado deberán contener audio adecuado para el contexto educativo de una clase.

3.2 Modelos

Los alumnos son libres de escoger los modelos acústicos y/o los vocoders que prefieran para crear su pipeline TTS. No obstante, es de destacar que el foco de esta práctica está en la parte acústica (texto a representación del sonido) por lo que no se aconseja a los alumnos perder el tiempo intentando encontrar el mejor vocoder para su caso de uso ya que, para estas tareas, su impacto en el pipeline TTS es significativamente menor que el del modelo acústico.

Se da total libertad a los alumnos para escoger el modelo que prefieran, siempre y cuando permita zero-shot voice cloning. No se aceptarán entrenamientos ni fine-tunings, ya que no son el foco de la prática. Algunos ejemplos de librerías usables son: coqui-TTS (o forks), GPTSoVITS, Index-TTS, Tortoise-TTS, etc.

3.3 Evaluación

Los alumnos deberán evaluar el rendimiento de las diferentes modelos probadas mediante métricas objetivas y valoraciones subjetivas.

En primer lugar, se deberán seleccionar y usar, al menos, **una métrica**. Esta métrica deberá ser adecuadas para la tarea objetivo, zero-shot voice cloning, por lo que deberá medir la similitud entre las voces del audio de referencia y de los audios generados. Los alumnos deberán realizar un análisis de resultados del modelo o modelos probados con las diferentes configuraciones probadas basándose en estas métricas.

Adicionalmente, los alumnos también pueden usar métricas cualitativas, basándose en su *impresión* de los resultados de los modelos.

Para esta tarea se pueden usar múltiples librerías, algunos ejemplos son: Resemblyzer, SpeechBrain, torchaudio, etc.

3.4 Memoria

Como siempre los alumnos deberán realizar una memoria. En este caso, la memoria deberá contener información sobre por qué se ha seleccionado el modelo, pruebas o cambios realizados con el (si aplica) y resultados obtenidos con el mismo en las diferentes pruebas, evaluados con al menos 2 métricas.

Es importante recalcar que cualquier trabajo hecho, pero no reflejado de manera adecuada en la memoria con el análisis debido, **no se tendrá en cuenta**.

Los alumnos deberán explicar todas las decisiones, métricas y resultados obtenidos durante la realización de la práctica, teniendo en cuenta, por ejemplo, el rendimiento (a nivel computacional y temporal) de los diferentes modelos ya que, como sabemos, en TTS la latencia puede ser decisiva.

La memoria tendrá una longitud acotada, no pudiendo superar en ningún caso las 3000 palabras ni ser más corta de 500. Esta memoria podrá contener el soporte audiovisual escogido por el alumno tales como fotos o incluso vídeos (mediante enlaces, por ejemplo). La memoria es parte imprescindible del trabajo, en caso de no entregarse o entregarse de manera deficiente, la práctica será suspensa.

4 Entrega y evaluación

Se habilitará un **repositorio de entrega** con fecha límite a las **23:59:00 del viernes 21 de noviembre**. En dicho repositorio deberán subirse los archivos correspondientes o, en su defecto, un **enlace a un repositorio tipo Git** que contenga la totalidad del software y la memoria del proyecto.

Las **entregas fuera de plazo** y/o los **commits realizados después de la fecha límite** serán motivo de **suspenso automático** en la práctica.

Del mismo modo, se considerará motivo de suspenso el **no haber realizado la práctica**, incluyendo casos de **plagio, copia de repositorios ajenos o uso indebido de herramientas de IA**.

Como se ha indicado previamente, la **memoria** es un elemento **fundamental** de la práctica; por tanto, una entrega incompleta, plagiada o de calidad insuficiente será considerada **no apta**.

La evaluación tendrá en cuenta, además de la calidad del código y de la memoria, especialmente el análisis realizado de los resultados y so comparación.

El **formato de entrega** es libre, siempre que se cumplan las siguientes condiciones:

- Debe incluir **todo lo necesario para ejecutar el código**.
- Debe incluir la **memoria del proyecto**.
- Debe incorporar un **Dockerfile** que permita instalar las dependencias y ejecutar el código.
- Debe incluir una **receta de Makefile** (especificada en el archivo *README* o en la memoria) para construir la imagen de Docker y ejecutar.

4.1 Criterios de evaluación

Criterio	Ponderación	Descripción
Código e implementación	40%	El alumno ha creado código adecuado para la ejecución de las pruebas necesarias.
Memoria y justificación técnica	60%	Argumentación de decisiones (en caso de ser necesario), claridad expositiva, coherencia técnica y análisis adecuado y profundo de los resultados, adaptados a las temáticas presentadas en el dataset.