

Universidad de La Habana
Facultad de Matemática y Computación



Sintetizador de Voz en Español con Voces Cubanas

Autor:

Sandra Martos Llanes

Tutores:

Katy Castillo Rosado

Flavio Reyes

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Fecha

github.com/username/repo

Dedicación

Agradecimientos

Agradecimientos

Opinión del tutor

Opiniones de los tutores

Resumen

Resumen en español

Abstract

Resumen en inglés

Índice general

| | |
|--|-----------|
| Introducción | 1 |
| 1. Estado del Arte | 6 |
| 1.1. Sistemas de dos etapas | 6 |
| 1.1.1. Modelos TTS | 6 |
| 1.1.2. VoCoders | 12 |
| 1.2. Modelos de extremo a extremo | 13 |
| 1.2.1. VITS | 13 |
| 2. Propuesta | 14 |
| 2.1. Creación de la base de datos [14] | 16 |
| 2.2. Reentrenando para el ajuste de Tacotron-DDC | 17 |
| 2.2.1. Fine-Tuning | 17 |
| 2.3. Entrenamiento de modelo VITS desde cero | 18 |
| 3. Detalles de Implementación y Experimentos | 19 |
| 3.1. Creación de base de datos con voces cubanas. | 19 |
| Conclusiones | 20 |
| Recomendaciones | 21 |
| Bibliografía | 22 |

Índice de figuras

| | | |
|------|---|----|
| 2.1. | Modelo TTS Tacotron2-DDC entrenado en Español | 14 |
| 2.2. | Modelo TTS Tacotron2-DDC entrenado en Inglés | 14 |
| 2.3. | Modelo TTS Fast-Pitch entrenado en Inglés | 15 |
| 2.4. | Modelo TTS Glow-TTS entrenado en Inglés | 15 |

Ejemplos de código

Introducción

La era de la informática conversacional está cambiando la forma en la que interactuamos con nuestros dispositivos: el Asistente de Google busca en Internet y lee las instrucciones sobre cómo preparar un pastel, Siri nos guía en la búsqueda de un lugar desconocido, las líneas automatizadas de servicio al cliente operan sin necesidad de esperas o botones. Uno de los primeros problemas que alguien se plantearía es la comunicación con los dispositivos, y es que no es posible determinar de qué forma va a interactuar una persona con estos. Se pueden hacer aproximaciones y suposiciones, pero cada persona puede decidir operar de una forma diferente, y sería imposible que un actor de voz grabase infinitas respuestas a las ramas de conversación que pueden surgir de una simple pregunta. En este punto entra la síntesis de voz, si en vez de unas grabaciones, el dispositivo pudiese sintetizar una voz humana, podría responder y hablar con la persona, entablando una especie de conversación.

La síntesis de voz es la producción artificial del habla humana, se han diseñado diferentes sistemas para este propósito, llamados sintetizadores de voz y pueden ser implementados tanto en hardware como en software.

Un sistema texto a voz (*text to speech*, TTS) o sintetizador de voz toma como entrada texto y produce voz audible como salida. Este sistema de conversión, se compone básicamente por dos componentes: un modelo que predice generalmente en forma de espectrograma la mejor pronunciación posible de cualquier texto dado, y un codificador de voz que, a partir del espectrograma anterior produce ondas sonoras de voz.

El texto a voz es un campo particularmente disciplinario, que requiere un conocimiento detallado en una variedad de ciencias. Si se quisiera construir un sistema TTS desde cero, se tendrían que estudiar los siguientes temas:

- Lingüística, el estudio científico del lenguaje. Para sintetizar un habla coherente, los sistemas TTS necesitan reconocer cómo un hablante humano pronuncia el lenguaje escrito; esto requiere conocimientos de lingüística hasta el nivel del fonema: las unidades de sonido que combinadas producen el habla, como el sonido /t/ en tierra. Para lograr un TTS verdaderamente realista, el sistema

también necesita predecir la prosodia apropiada, que incluye elementos del habla más allá del fonema, como acentos, pausas y entonación.

- Procesamiento de señales de audio, creación y manipulación de representaciones digitales de sonido. Las señales de audio (habla) son representaciones electrónicas de ondas sonoras. La señal de voz se representa digitalmente como una secuencia de números. En el contexto de TTS se utilizan diferentes representaciones de características que describen la señal del habla, lo que hace posible entrenar diversos modelos para generar una nueva voz.
- Inteligencia artificial, especialmente aprendizaje profundo, un tipo de aprendizaje automático que utiliza una arquitectura informática llamada red neuronal profunda o DNN por sus siglas en inglés. Una red neuronal es un modelo computacional inspirado en el cerebro humano. Está formado por redes complejas de procesadores, cada uno de los cuales realiza una serie de operaciones antes de enviar su salida a otro procesador. Una DNN capacitada aprende la mejor vía de procesamiento para lograr resultados precisos. Este modelo tiene una gran potencia informática, lo que lo hace ideal para manejar la gran cantidad de variables necesarias para la síntesis de voz de alta calidad.

Tipos de tecnologías TTS

Hasta la actualidad se han desarrollado variadas tecnologías TTS, las cuales operan de maneras distintas y se basan en diferentes ideas. Entre las más dominantes se encuentran:

1. Síntesis de formantes y síntesis articulatoria:

Los primeros sistemas TTS empleaban tecnologías basadas en reglas, como la síntesis de formantes y la síntesis articulatoria, que logró un resultado similar a través de estrategias ligeramente diferentes. A partir de una grabación realizada a un hablante, se extrajeron características acústicas de este: formantes, cualidades definitorias de los sonidos del habla, en síntesis de formantes; y forma de articulación (nasal, oclusiva, vocal, etc.) en síntesis articulatoria. Luego, se programarían reglas que recrearan estos parámetros con una señal de audio digital. Este TTS era bastante robótico; y estos enfoques necesariamente abstraen gran parte de la variación que se encontrará en el habla humana, cosas como la variación de tono, porque solo permiten a los programadores escribir reglas para unos pocos parámetros a la vez.

2. Síntesis de difonos:

El próximo gran desarrollo en el campo TTS se llama síntesis de difonos, se inició en la década de 1970 y todavía era de uso popular durante los últimos

años del siglo XX. La síntesis de difonos crea un habla de máquina mediante la combinación de difonos, combinaciones de fonemas de una sola unidad, y las transiciones de un fonema al siguiente; es decir, no solo la /t/ en la palabra tierra sino la /t/ más la mitad del siguiente sonido /i/.

La tecnología TTS de síntesis de difonos incluye también modelos que predicen la duración y el tono de cada difono para la entrada dada. Estos dos sistemas se superponen el uno sobre el otro, primero conecta las señales de difono y luego procesa la señal para corregir el tono y la duración. El resultado es un discurso sintético con un sonido más natural que el que crea la síntesis de formantes, pero aún está lejos de ser perfecto, y tiene pocas ventajas sobre cualquier otro acercamiento más que su tamaño.

3. Síntesis de selección de unidades:

La síntesis de selección de unidades constituye un enfoque ideal para los motores TTS de bajo impacto en la actualidad. Cuando la síntesis de difonos añadió duración y el tono apropiado a través de un segundo sistema de procesamiento, la síntesis de selección de unidad omite ese paso: se inicia con una gran base de datos grabados del habla, alrededor de 20 horas o más, y selecciona los fragmentos de sonido que ya tienen la duración y el tono. La síntesis de selección de unidades proporciona un habla similar a la humana sin mucha modificación de la señal, pero sigue siendo identificablemente artificial. Probablemente el audio de salida de la mejor selección de unidades sea indistinguible de las voces humanas reales, especialmente en contextos con sistemas TTS. Sin embargo, un mayor naturalidad requiere de bases de datos de selección de unidades muy grandes, en algunos sistemas llegando a ser de gigabytes de datos grabados, representando docenas de horas de voz.

4. Síntesis neuronal:

La tecnología de las redes neuronales profundas es la que impulsa los avances actuales en el campo TTS, y es clave para la obtención de resultados mucho más realistas. Al igual que sus predecesores, el TTS neuronal comienza con grabaciones de voz; la otra entrada es texto, el guión escrito que su locutor de origen utilizó para crear esas grabaciones. Alimenta estas entradas en una red neuronal profunda y aprenderá el mejor mapeo posible entre un bit de texto y las características acústicas asociadas.

Una vez que el modelo esté entrenado, podrá predecir sonido realista para nuevos textos: con un modelo TTS neuronal entrenado, junto con un codificador de voz entrenado con los mismos datos, el sistema puede producir un habla que es notablemente similar a la del locutor de origen cuando se expone a prácticamente cualquier texto nuevo. Esa similitud entre la fuente y la salida es la

razón por la que el TTS neuronal a veces se denomina “clonación de voz”.

Hay todo tipo de trucos de procesamiento de señales que pueden ser utilizados para alterar la voz sintética resultante y no se asemeje al locutor fuente. La investigación actual está conduciendo a voces TTS que hablan con expresión emocional, voces únicas en varios idiomas y una calidad de audio cada vez más realista.

Entre algunos de los poderosos casos de uso de TTS en el mundo actual se encuentran:

- Sistemas de respuesta de voz interactiva conversacional (IVR), como en los centros de llamadas de servicio al cliente.
- Aplicaciones de comercio de voz, como comprar en un dispositivo Amazon Alexa.
- Herramientas de navegación y guía por voz, como aplicaciones de mapas GPS.
- Dispositivos domésticos inteligentes y otras herramientas de Internet de las cosas (IoT) habilitadas por voz.
- Asistentes virtuales independientes como Siri de Apple.
- Soluciones de publicidad y marketing experiencial, como anuncios de voz interactivos en servicios de transmisión de música.
- Desarrollo de videojuegos.
- Videos de marketing y formación de la empresa que permiten a los creadores cambiar las voces en off sin identificar al locutor.

Problemática

Existe un conjunto de sistemas TTS que se enmarcan bajo una licencia de software libre:

- Festival y Festvox
- Plataforma MaryTTS
- Sistema TTS
- SV2TTS

- Mozilla-TTS
- COQUI -TTS

A partir del estudio parcial de las plataformas de código abierto utilizadas para el desarrollo de conversores de texto a voz, fue posible comprobar que la mayoría se encuentran basadas en la síntesis neuronal teniendo en cuenta que alcanza mejores resultados. Estas plataformas brindan modelos previamente entrenados para idiomas específicos como inglés, francés, alemán, etc. Muy pocas presentan un modelo en español, y las que lo hacen solo poseen uno, con acento de voz española o voz neutra.

DATYS, es una empresa de desarrollo de software, que como parte de sus soluciones requiere un sintetizador de voz en español con acento propio de nuestro país; esto formará parte de un proyecto que consiste en el desarrollo del primer asistente virtual cubano.

Objetivo

Objetivo General

Desarrollar un sintetizador de texto a voz con voces cubanas.

Objetivos específicos

1. Analizar en profundidad los métodos y plataformas existentes para la síntesis de voz, principalmente las que trabajan sobre la síntesis neuronal, y seleccionar el más adecuado a utilizar para las aplicaciones de DATYS.
2. Diseñar y conformar una base de datos en español, con con voces cubanas para el entrenamiento de los modelos y muestras de voz que se desea generar.
3. Reentrenar el modelo basado en síntesis neuronal seleccionado, para su ajuste al estilo de voz cubana.
4. Evaluar.

Capítulo 1

Estado del Arte

Las redes neuronales han sido las encargadas de los avances actuales en el campo TTS, por tanto esta será la línea seguida en la investigación.

Coqui TTS[1] es una biblioteca para la generación avanzada de texto a voz. Se basa en las últimas investigaciones y se diseñó para lograr el mejor equilibrio entre la facilidad de entrenamiento, la velocidad y la calidad. Coqui viene con modelos preentrenados, herramientas para medir la calidad del conjunto de datos y ya se utiliza en más de 20 idiomas para productos y proyectos de investigación.

Para lograr la transformación de texto a voz, Coqui sigue dos enfoques distintos. El primero es un sistema de dos etapas: una combinación de dos modelos, uno para convertir de texto a espectrograma de mel, y luego otro de espectrograma a onda sonora; y el segundo es un modelo de extremo a extremo.

1.1. Sistemas de dos etapas

El paradigma predominante en la conversión de texto a voz es la síntesis en dos etapas: primero, producir espectrogramas a escala de mel a partir del texto y, luego, las ondas de sonido reales con un modelo de codificador de voz(VoCoder). La representación acústica de bajo nivel: espectrograma de mel, es la utilizada como nexo entre las dos componentes.

1.1.1. Modelos TTS

FastSpeech

FastSpeech, una novedosa red de avance basada en Transformer para generar espectrogramas de mel en paralelo para TTS; toma como entrada una secuencia de

texto(fonema) y genera espectrogramas de mel de forma no autorregresiva. Adopta una red feed-forward basada en la autoatención en Transformer y convolución de 1D.

El modelo resuelve problemas existentes en otros modelos TTS de la siguiente forma:

- A través de la generación de espectrogramas de mel paralelos, FastSpeech acelera enormemente el proceso de síntesis.
- El predictor de duración de fonemas asegura alineaciones estrictas entre un fonema y sus espectrogramas, lo que es muy diferente de las alineaciones de atención automáticas y suaves en los modelos autorregresivos. Por lo tanto, FastSpeech evita los problemas de propagación de errores y alineaciones de atención incorrectas, lo que reduce la proporción de palabras omitidas y palabras repetidas.
- El regulador de longitud puede ajustar fácilmente la velocidad de la voz alargando o acortando la duración del fonema para determinar la duración de los espectrogramas de mel generados, y también puede controlar parte de la prosodia añadiendo pausas entre fonemas adyacentes.

Arquitectura

Feed-Forward Transformer:

La arquitectura para Fast Speech es una estructura de avance basada en la autoatención en Transformer y la convolución de 1D; se nombra esta estructura como Feed-Forward Transformer (FFT). Feed-Forward Transformer apila múltiples bloques FFT para la transformación de fonema a espectrograma de mel, con N bloques en el lado del fonema y N bloques en el lado del espectrograma de mel, con un regulador de longitud en el medio para cerrar la brecha de longitud entre el fonema y la secuencia del espectrograma de mel. Cada bloque FFT consta de una red convolucional de 1D y de autoatención. La red de autoatención consiste en una atención de múltiples cabezas para extraer la información de posición cruzada.

Regulador de longitud:

El regulador de longitud se utiliza para resolver el problema de la discordancia de longitud entre el fonema y la secuencia del espectrograma en el transformador de avance, así como para controlar la velocidad de la voz y parte de la prosodia. La longitud de una secuencia de fonemas suele ser menor que la de su secuencia de espectrograma de mel, y cada fonema corresponde a varios espectrogramas de mel.

Predictor de duración:

La predicción de la duración de los fonemas es importante para el regulador de longitud. El predictor consta de una red convolucional de 1D de 2 capas con activación ReLU, cada una seguida de la normalización de la capa y la capa de exclusión, y una capa lineal adicional para generar un escalar, que es exactamente la duración prevista del fonema.

Experimentación

El entrenamiento de FastSpeech y de todos los modelos TTS se realiza sobre un conjunto de datos que contiene clips de audios con sus correspondientes transcripciones de texto. Específicamente para FastSpeech, se divide al azar el conjunto de datos en 3 conjuntos: muestras para entrenamiento, muestras para validación y muestras para las pruebas.

El modelo FastSpeech puede casi coincidir con el modelo autoregresivo Transformer TTS en términos de calidad de voz, acelera la generación de espectrograma mel por 270x y la síntesis de voz de extremo a extremo por 38x, casi se elimina el problema de saltar y repetir palabras, y puede ajustar la velocidad de voz (0.5x-1.5x) sin problemas[2].

FastPitch

FastPitch es un modelo TTS feed-forward completamente paralelo basado en FastSpeech, condicionado por contornos de frecuencia fundamentales. El modelo predice contornos de tono durante la inferencia. Al alterar estas predicciones, el discurso generado puede ser más expresivo, coincidir mejor con la semántica del enunciado y, al final, ser más atractivo para el oyente.

Los modelos paralelos pueden sintetizar órdenes de magnitud de espectrogramas de mel más rápido que los autorregresivos, ya sea basándose en alineaciones externas o alineándose ellos mismos. El condicionamiento en la frecuencia fundamental también mejora la convergencia y elimina la necesidad de destilar el conocimiento de los objetivos del espectrograma de mel utilizados en FastSpeech.

Arquitectura

Se basa en FastSpeech y se compone principalmente de dos pilas de transformadores alimentados hacia adelante, feed-forward(FFTr) . El primero opera en la resolución de los tokens de entrada, el segundo en la resolución de los cuadros de salida. La primera pila de FFTr produce la representación oculta h . La representación oculta, se

usa para hacer predicciones sobre la duración y el tono promedio de cada caracter con una red neuronal convolucional de 1D. A continuación, el tono se proyecta para que coincida con la dimensionalidad de la representación oculta y se suma a h . La suma se muestrea discretamente y se pasa a la salida FFTr, que produce la secuencia de espectrograma mel de salida[3].

Experimentación

Para el entrenamiento y la experimentación los parámetros del modelo siguen principalmente FastSpeech.

Tacotron

Tacotron es un modelo TTS de tipo secuencia a secuencia con un paradigma de atención. Este modelo toma caracteres como entrada y devuelve un espectrograma sin procesar usando técnicas para mejorar un modelo *vanilla seq2seq*. Dado un par <texto,audio>, Tacotron puede ser entrenado desde cero con una inicialización aleatoria, y no requiere alineación a nivel de fonema.

Arquitectura

La columna vertebral de Tacotron es un modelo seq2seq con atención, que toma caracteres como entrada, y devuelve el correspondiente espectrograma sin procesar, para luego pasarlo al modelo o algoritmo que sintetiza la voz. En el centro de todo esto se encuentra un codificador, un decodificador basado en atención y una red de post procesamiento.

Tacotron se basa en cuadros, o frames en inglés, por lo que la inferencia es sustancialmente más rápida que los métodos autorregresivos a nivel de muestra. A diferencia de otros trabajos más antiguos, Tacotron no necesita características lingüísticas diseñadas a mano ni componentes complejos como un alineador de Modelo Markov oculto(HMM). Este modelo realiza una normalización de texto simple[4].

Tacotron 2

Tacotron 2 es similar al anteriormente mencionado Tacotron; es una red recurrente de predicción de características, de tipo secuencia a secuencia con atención, que mapea incrustaciones(embeddings) de caracteres en espectrogramas a escala de mel.

Un espectrograma de frecuencia de mel está relacionado con el espectrograma de frecuencia lineal, es decir, la magnitud de la transformada de Fourier de tiempo corto (STFT). Se obtiene aplicando una transformada no lineal al eje de frecuencia de la

STFT, inspirado en respuestas calificadas por el sistema auditivo humano, y resume el contenido de frecuencia con menos dimensiones.

El uso de una escala de frecuencia auditiva de este tipo tiene el efecto de enfatizar detalles en frecuencias más bajas, que son fundamentales para la inteligibilidad del habla, al mismo tiempo que se resta importancia a los detalles de alta frecuencia, que están dominados por ráfagas de ruido y generalmente no necesitan ser modelados con alta fidelidad. Debido a estas propiedades, las características derivadas de la escala de mel se han utilizado como representación base para el reconocimiento de voz durante muchas décadas.

Para Tacotron2 los espectrogramas de mel se calculan a través de una transformada de Fourier de tiempo corto (STFT) utilizando un tamaño de cuadro de 50 ms, 12,5 ms de salto de cuadro y una función de ventana de Hann. Se transforma la magnitud de la STFT a la escala de mel usando un banco de filtros de 80 canales mel que abarca de 125 Hz a 7,6 kHz, seguido de una compresión de registro de rango dinámico. Antes de la compresión de registros, las magnitudes de salida del banco de filtros se recortan a un valor mínimo de 0,01 para limitar el rango dinámico en el dominio logarítmico.[5]

La red del modelo en cuestión está compuesta por un codificador y un decodificador con atención. El codificador convierte una secuencia de caracteres en una representación oculta que alimenta al decodificador para predecir un espectrograma.

Los caracteres de entrada se representan utilizando una incrustación de caracteres 512-dimensional, que se pasan a través de una pila de 3 capas convolucionales, cada una de las cuales contiene 512 filtros con organización 5×1 , es decir, donde cada filtro abarca 5 caracteres, seguido de la normalización por lotes y activaciones de ReLU[5]. Como en Tacotron, estas capas convolucionales modelan el contexto a largo plazo en la secuencia de caracteres de entrada. La salida de la capa convolucional final se pasa a una sola capa bidireccional LSTM que contiene 512 unidades (256 en cada dirección) para generar las características codificadas.

La salida del codificador es consumida por una red de atención que resume la secuencia codificada completa como un vector de contexto de longitud fija para cada paso de salida del decodificador. Se usa la atención sensible a la ubicación de [6], que extiende el mecanismo de atención aditiva [7] para usar pesos de atención acumulativos de anteriores pasos de tiempo del decodificador como una funcionalidad adicional. Esto anima al modelo a seguir adelante consistentemente a través de la entrada, mitigando los posibles modos de falla donde algunas subsecuencias son repetidas o ignoradas por el decodificador. Las probabilidades de atención se calculan después de proyectar las entradas y funciones de localización a representaciones ocultas de 128 dimensiones. Las funcionalidades de localización se calculan utilizando 32 filtros de convolución 1-D de longitud 31[5].

El decodificador es una red neuronal autorregresiva recurrente que predice un espectrograma de mel a partir de la secuencia de entrada codificada un fotograma a la vez. La predicción del paso de tiempo anterior es primero pasado a través de una pequeña red previa que contiene 2 capas completamente conectadas de 256 unidades ReLU ocultas. La pre-red actuando como un cuello de botella de información es esencial para el aprendizaje de la atención.

La salida de la pre-red y el vector de contexto de atención se concatenan y pasan a través de una pila de 2 capas *Long Short-Term Memory (LSTM)* unidireccionales con 1024 unidades. La concatenación de la salida LSTM y el vector de contexto de atención se proyecta a través de una transformación lineal para predecir el cuadro de espectrograma objetivo. Finalmente, se pasa el espectrograma de mel predicho a través de una post-red convolucional de 5 capas que predice un residuo a agregar a la predicción para mejorar la reconstrucción general. Cada capa post-net está compuesta por 512 filtros con forma 5×1 con lote normalización, seguida de activaciones de tanh en todas las capas excepto en la final[5].

En paralelo a la predicción de cuadros de espectrograma, la concatenación de salida del decodificador LSTM y el contexto de atención se proyectan a un escalar y pasa a través de una activación sigmoidea para predecir la probabilidad de que la secuencia de salida se haya completado. La predicción de este “token de parada” se usa durante la inferencia para permitir que el modelo determine dinámicamente cuándo terminar la generación en lugar de generar siempre por una duración fija. Específicamente, la generación se completa en el primer fotograma para el que esta probabilidad supera un umbral de 0,5.

Las capas convolucionales en la red se regularizan usando abandono(dropout) con probabilidad 0.5, y las capas LSTM son regularizadas usando desconexión(zoneout) con probabilidad 0.1. Para introducir la variación de salida en tiempo de inferencia, se aplica dropout con probabilidad 0.5 solo a capas en la pre-red del decodificador autorregresivo[5].

En contraste con el Tacotron original, este modelo utiliza una construcción más simple, usando LSTM vainilla y capas convolucionales en el codificador y decodificador en lugar de pilas CBHG y capas recurrentes GRU. No se usa un "factor de reducción", es decir, cada paso del decodificador corresponde a un único cuadro de espectrograma.

Este sistema puede ser entrenado directamente desde un conjunto de datos sin depender de una compleja ingeniería de características, y logra calidad de sonido de última generación cercana a la del habla humana natural. Los resultados de Tacotron 2, constituyen un paso de avance sobre Tacotron y otros sistemas previos, sin embargo dejan aún espacio para mejoras.

1.1.2. VoCoders

Los VoCoders neuronales basados en redes neuronales profundas pueden generar voces similares a las humanas, en lugar de utilizar los tradicionales métodos que contienen artefactos audibles[8][9][10].

La línea principal de la investigación se basa en los modelos TTS, como los antes expuestos, sin embargo como no es posible sintetizar voz sin un VoCoder, y luego de varias pruebas realizadas, se concluye que los más adecuados para el objetivo principal son los siguientes:

UnivNet

La mayoría de los codificadores de voz neuronales emplean espectrogramas de mel de banda limitada para generar formas de onda. UnivNet, es un codificador neural de voz que sintetiza formas de onda de alta fidelidad en tiempo real.

Usando espectrogramas de mel de banda completa como entrada, se espera generar señales de alta resolución agregando un discriminador que emplea espectrogramas de múltiples resoluciones como entrada. En una evaluación de un conjunto de datos que contiene información sobre cientos de ponentes, UnivNet obtuvo los mejores resultados positivos, objetivos y subjetivos, entre los modelos que competían. Estos resultados, incluida la mejor puntuación subjetiva en la conversión texto a voz, demuestran el potencial para una rápida adaptación a nuevos hablantes sin necesidad de entrenamiento desde cero[11].

WaveGrad

WaveGrad es un modelo condicional para la generación de formas de onda que estima los gradientes de la densidad de datos. El modelo se basa en trabajos previos sobre emparejamiento de puntuaciones y modelos probabilísticos de difusión. Parte de una señal Gaussiana de ruido blanco e iterativamente refina la señal a través de un muestreador basado en gradientes, condicionado en el espectrograma de mel. WaveGrad ofrece una forma natural de intercambiar velocidad de referencia por calidad de la muestra ajustando el número de pasos de refinamiento, y cierra la brecha entre los modelos autorregresivos y no autorregresivos en términos de calidad de audio. El modelo puede generar muestras de audio de alta fidelidad usando como tan solo seis iteraciones. Los experimentos revelan que WaveGrad genera señales de audio de alta fidelidad, superando las líneas de base adversariales no autorregresivas y emparejando un fuertemente la línea de base autorregresiva basada en la probabilidad, utilizando menos operaciones secuenciales[12].

1.2. Modelos de extremo a extremo

1.2.1. VITS

Variational Inference with adversarial learning for end-to-end Text-to-Speech (VITS) es un método TTS de extremo a extremo en paralelo. Usando un Autocodificador Variacional se conectan los dos módulos de sistemas TTS: modelo acústico y VoCoder, a través de variables latentes para permitir el aprendizaje de extremo a extremo.

Para mejorar el poder expresivo del método con el fin de sintetizar formas de onda de voz de alta calidad, se aplican flujos de normalización a la distribución condicional previa y entrenamiento adversarial en el dominio de formas de ondas.

El modelo VITS se describe principalmente en tres etapas: una formulación condicional de Autocodificador variacional; estimación de alineación derivada de la inferencia variacional; y entrenamiento adversarial para mejorar la calidad de la síntesis.

(***No se si explicar cada parte)

Luego de un proceso de entrenamiento, experimentación y comparación con sistemas de dos etapas, utilizando los modelos preentrenados Tacotron2 y Glow-TTS como modelos de primer escenario e HiFi-GAN como modelo de segundo escenario, se comprueba que VITS obtiene un habla que suena más natural. Una evaluación humana subjetiva (puntuación de opinión media, o MOS) en LJ Speech, un conjunto de datos de un solo hablante, muestra que el modelo supera a los mejores sistemas TTS disponibles públicamente y logra un MOS comparable a la realidad del terreno.

Ha mostrado la capacidad de ampliarse a la síntesis de voz de múltiples hablantes, generando un discurso con diversos tonos y ritmos de acuerdo a diferentes identidades de hablantes. Esto demuestra que aprende y expresa varias características del habla en un contexto de extremo a extremo.

Capítulo 2

Propuesta

Con el objetivo de lograr una síntesis de voz satisfactoria y después de realizar un estudio de las tecnologías que se utilizan para este fin, se elige Coqui TTS como herramienta base. Coqui cuenta con una gran variedad de modelos preentrenados en más de 20 idiomas. El primer paso en el desarrollo del sintetizador fue instalar la biblioteca Coqui TTS, de acuerdo a las indicaciones del repositorio oficial[1]. Se realizó un estudio del comportamiento de combinaciones de parejas modelo TTS y VoCoder, para evaluar cuáles ofrecían mejores resultados:

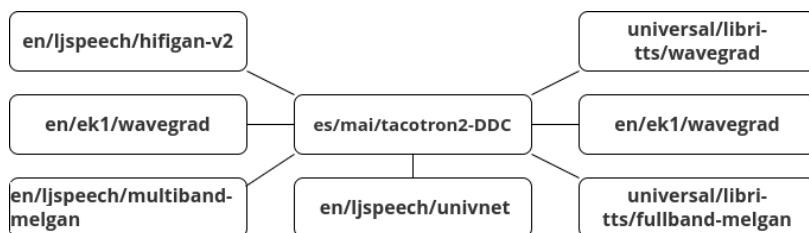


Figura 2.1: Modelo TTS Tacotron2-DDC entrenado en Español

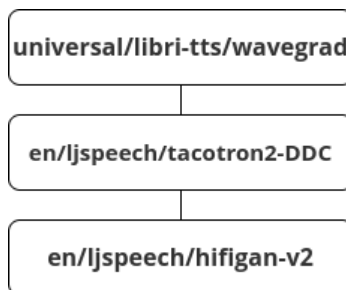


Figura 2.2: Modelo TTS Tacotron2-DDC entrenado en Inglés

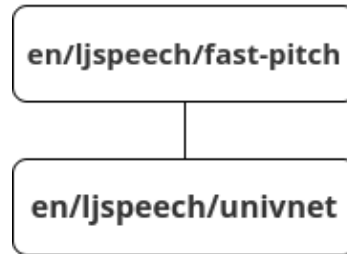


Figura 2.3: Modelo TTS Fast-Pitch entrenado en Inglés

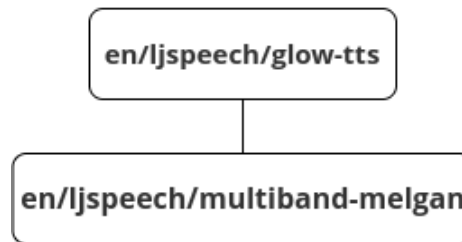


Figura 2.4: Modelo TTS Glow-TTS entrenado en Inglés

Los modelos **Tacotron2-DDC**, **Fast-pitch** y **Glow-tts** preentrenados en idioma inglés al combinarse con VoCoders, como se observa en las figuras (citar figuras) para un texto en español arroja distintos resultados, en algunos casos solo producen ruido mientras que los más satisfactorios, producen un discurso con una pronunciación propia de una persona de habla inglesa hablando español.

Por otro lado Coqui solo cuenta con un modelo preentrenado en español, el **Tacotron-DDC**, que está entrenado sobre la base de datos de M-AILABS[13]; este modelo fue probado junto a los VoCoders preentrenados como se muestra en la figura 2.1. Los mejores resultados fueron arrojados por las combinaciones de **Tacotron-DDC** con los modelos: Univnet entrenado sobre la base de datos Ljspeech en inglés, y Wavegrad entrenado sobre el conjunto de datos LibriTTS, aunque ambos presentan problemáticas como la mala pronunciación de la letra ñ.

De acuerdo con la documentación de Coqui, y con el objetivo de la investigación que consiste en desarrollar un sintetizador de voz en español con voz cubana, se consideraron tres enfoques:

1. Realizar un *fine-tuning* o reentrenamiento a un modelo preentrenado de [1], en este caso **Tacotron-DDC**, que es el único disponible en español.
2. Realizar el entrenando desde cero de un modelo; en este caso se podría entrenar cualquier modelo disponible. **Glow-TTS** y **VITS** son buenos candidatos pues en general producen resultados bastante satisfactorios.

3. La problemática del tamaño de la base de datos puede resolverse entrenando el modelo **YourTTS**, que según la literatura, es capaz de adaptarse a una nueva voz con muy poco tiempo de audio.

Para el desarrollo de cualquiera de los anteriores es imprescindible la construcción de un conjunto de datos que se adapten al modelo y a las necesidades de la investigación.

2.1. Creación de la base de datos [14]

Para entrenar un modelo TTS, se necesita un conjunto de datos con grabaciones de voz y transcripciones, en este caso fue una base de datos en español con voces cubanas. El discurso debe dividirse en clips de audio y cada clip necesita su transcripción correspondiente. Debe poseer un formato específico de forma tal que el cargador de datos de Coqui TTS sea capaz de cargarlos para utilizar en el entrenamiento.

La base de datos conformanda debe tener una buena cobertura del idioma en el que se desea entrenar el modelo. Debe cubrir la variedad fonémica, los sonidos excepcionales y las sílabas.

Qué hace a un buen Dataset?

- Debe cubrir una cantidad considerable de clips cortos y largos.
- Libre de errores. Se debe eliminar cualquier archivo incorrecto o roto.
- Para escuchar una voz con la mayor naturalidad posible con todas las diferencias de frecuencia y tono, por ejemplo, usando diferentes signos de puntuación.
- Es necesario que el *dataset* cubra una buena parte de fonemas, difonemas y, en algunos idiomas, trifenemas. Si la cobertura de fonemas es baja, el modelo puede tener dificultades para pronunciar nuevas palabras difíciles.

Los clips de audio poseen formato .wav y se organizan dentro de una carpeta de nombre *wavs* de la siguiente forma:

```
    /wavs
    | - audio1.wav
    | - audio1.wav
    | - audio2.wav
    | - audio3.wav
    ...
```

Las transcripciones se recogen dentro del archivo metadata.csv. Donde audio1, audio2, etc se refieren a los archivos audio1.wav, audio2.wav etc.

```
audio1|Esta es mi transcripción 1.  
audio2|Esta es mi transcripción 2.  
audio3|Esta es mi transcripción 3.  
audio4|Esta es mi transcripción 4.
```

El modelo sobre el que realizaremos el ajuste, está preentrenado sobre la base de datos en Español de *The M-AiLabs Speech Dataset*, por tanto utilizaremos la misma estructura de este en la conformación de la base de datos con voces cubanas. Finalmente queda la siguiente estructura:

```
MyDataset/by_book/female/[creador del dataset]/[nombre del hablante]  
|/wavs  
|metadata.csv
```

2.2. Reentrenando para el ajuste de Tacotron-DDC

Se propone desarrollar el primer enfoque: reentrenando(*fine-tuning* en inglés) el modelo **Tacotron-DDC** en español, pues trae ventajas tales como un aprendizaje más rápido, ya que un modelo preentrenado ya tiene aprendidas funcionalidades que son relevantes para la tarea de producir un discurso. Convergerá más rápido en el nuevo *dataset*, lo que reducirá el costo de entrenamiento y permitirá el proceso de experimentación más rápidamente. Además se pueden obtener buenos resultados con un conjunto de datos más pequeño.

Luego de tener entrenado el modelo acústico(tts) con una base de datos construida con voces cubanas, es posible que alguno de los VoCoders preentrenados disponibles produzca una salida con las características deseadas, en caso contrario se debería entrenar el VoCoder con los datos del *dataset* construido.

2.2.1. Fine-Tuning

Entrenar correctamente un modelo de aprendizaje profundo requiere generalmente de una gran base de datos y de un extenso entrenamiento.

Si se dispone del material necesario y del tiempo para entrenar el algoritmo, estos requisitos no suponen ningún problema, pero, si la base de datos es pequeña o el modelo no se entrena lo suficiente, el aprendizaje podría no ser completo.

El *fine-tuning* es una forma de transferencia de aprendizaje, consiste en aprovechar la información que se ha aprendido de la resolución de un problema y utilizarla sobre otro distinto, pero que comparte ciertas características. Se usan los conocimientos adquiridos por una red convolucional para transmitirlos a una nueva red convolucional. Esta nueva red convolucional que se crea no tiene por qué modificar la red original y puede simplemente aprender de ella, sin embargo también es válido el caso donde no solo se modifica la red original, sino que se vuelve a entrenar para aprender más conceptos.

En la presente investigación, se utiliza el reentrenamiento, para a partir modelo previamente entrenado y realizar un nuevo entrenamienor para mejorar su rendimiento en un conjunto de datos diferente.

El proceso consiste en modificar la configuración original del modelo preentrenado seleccionado, es decir, establecer la base de datos a utilizar en el reentrenamiento, los detalles acústicos que reflejen las características del nuevo conjunto de datos, el nombre del nuevo modelo ajustado, la dirección donde se guardará el modelo reentrenado, entre otras cuestiones. Para la mayoría de los parámetros se tomaron las características del modelo original **Tacontron-DDC** en español.

2.3. Entrenamiento de modelo VITS desde cero

Existen varios modelos de texto a voz de extremo a extremo que permiten el entrenamiento en una sola etapa y el muestreo en paralelo, sin embargo, generalmente la calidad de la muestra no coincide con la de los sistemas TTS de dos etapas.

Se selecciona VITS porque es un método TTS paralelo de extremo a extremo que genera un sonido de audio más natural que los modelos actuales de dos etapas. Y de acuerdo a una evaluación humana subjetiva (puntuación de opinión media, o MOS), muestra que el modelo supera a los mejores sistemas TTS disponibles públicamente y logra un MOS comparable a la realidad del terreno.

Capítulo 3

Detalles de Implementación y Experimentos

3.1. Creación de base de datos con voces cubanas.

Cuban Voice Dataset

La base de datos está conformada por 160 clips de audio con sus respectivas transcripciones recogidas en el archivo `metadata.csv`. Cada clip tiene una duración de 2 a 15 segundos.

Utilizando **RNNoise**, la cual es una biblioteca basada en una red neuronal para la eliminación de ruido en grabaciones, obtenemos clips de audio libres de ruidos y con la frecuencia de muestreo deseada.

Conclusiones

Conclusiones

Recomendaciones

Recomendaciones

Bibliografía

- [1] Coqui. «Coqui TTS.» (), dirección: <https://github.com/coqui-ai/TTS> (vid. págs. 6, 14, 15).
- [2] Y. Ren y col., «Fastspeech: Fast, robust and controllable text to speech,» *Advances in Neural Information Processing Systems*, vol. 32, 2019 (vid. pág. 8).
- [3] A. Łańcucki, «Fastpitch: Parallel text-to-speech with pitch prediction,» en *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, págs. 6588-6592 (vid. pág. 9).
- [4] Y. Wang y col., «Tacotron: Towards end-to-end speech synthesis,» *arXiv preprint arXiv:1703.10135*, 2017 (vid. pág. 9).
- [5] J. Shen y col., «Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,» en *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2018, págs. 4779-4783 (vid. págs. 10, 11).
- [6] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho e Y. Bengio, «Attention-based models for speech recognition,» *Advances in neural information processing systems*, vol. 28, 2015 (vid. pág. 10).
- [7] D. Bahdanau, K. Cho e Y. Bengio, «Neural machine translation by jointly learning to align and translate,» *arXiv preprint arXiv:1409.0473*, 2014 (vid. pág. 10).
- [8] D. Griffin y J. Lim, «Signal estimation from modified short-time Fourier transform,» *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, n.º 2, págs. 236-243, 1984 (vid. pág. 12).
- [9] H. Kawahara, I. Masuda-Katsuse y A. De Cheveigne, «Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds,» *Speech communication*, vol. 27, n.º 3-4, págs. 187-207, 1999 (vid. pág. 12).

- [10] M. Morise, F. Yokomori y K. Ozawa, «WORLD: a vocoder-based high-quality speech synthesis system for real-time applications,» *IEICE TRANSACTIONS on Information and Systems*, vol. 99, n.º 7, págs. 1877-1884, 2016 (vid. pág. 12).
- [11] W. Jang, D. Lim, J. Yoon, B. Kim y J. Kim, «UnivNet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation,» *arXiv preprint arXiv:2106.07889*, 2021 (vid. pág. 12).
- [12] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi y W. Chan, «WaveGrad: Estimating gradients for waveform generation,» *arXiv preprint arXiv:2009.00713*, 2020 (vid. pág. 12).
- [13] I. Solak. «The M-AiLabs Speech Dataset.» (), dirección: <https://www.caito.de/2019/01/03/the-m-ailabs-speech-dataset/> (vid. pág. 15).
- [14] C. TTS. «Formatting Your Dataset.» (), dirección: https://tts.readthedocs.io/en/latest/formatting_your_dataset.html#formatting-your-dataset (vid. pág. 16).