
**SISTEMA EN TIEMPO REAL DE PREVENCION Y MITIGACION DE INCENDIOS**

**Estudiantes:**
Rúa Echalar Juan Manuel **Ing. Ciencias de la Computación.**

**Docente:** Carlos Walter Pacheco Lora

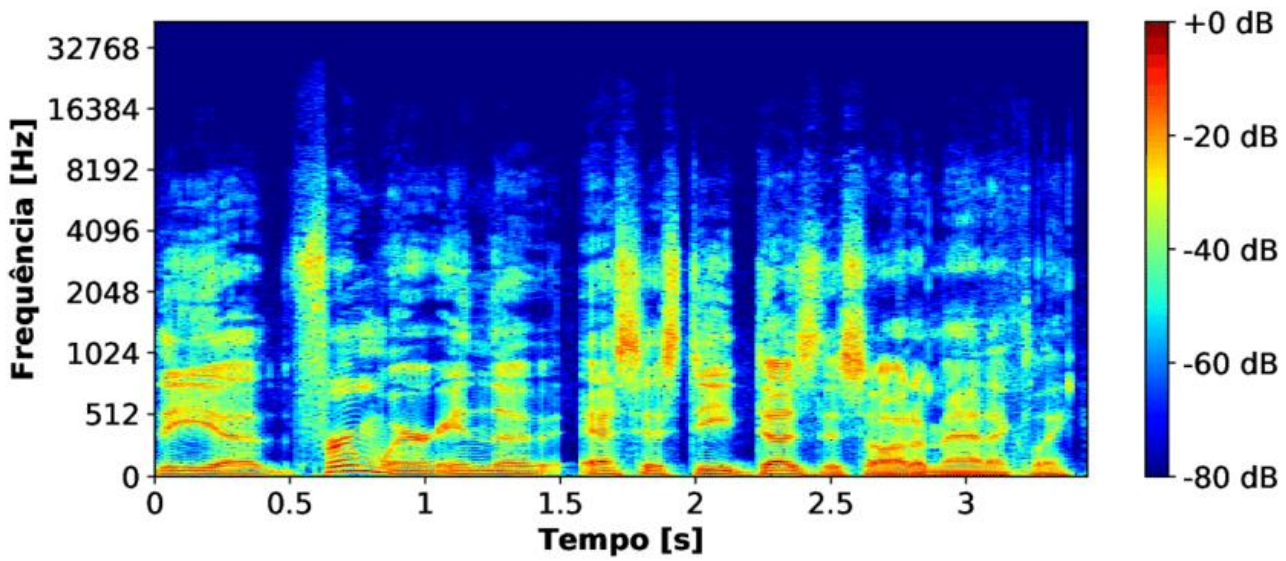
**Materia:** Inteligencia Artificial II

**Semestre:** 02/2024

**1. Introducción**

El objetivo de un modelo TTS es convertir texto en audio utilizando varias etapas que transforman la entrada textual en ondas sonoras reproducibles. A continuación, se describe cada componente y proceso involucrado.

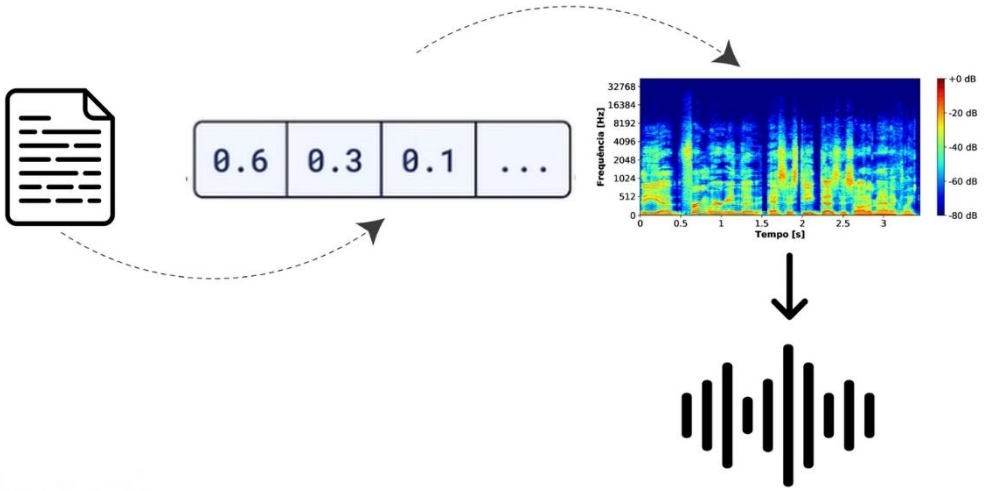
**2. Representación del Mel-Espectrograma**

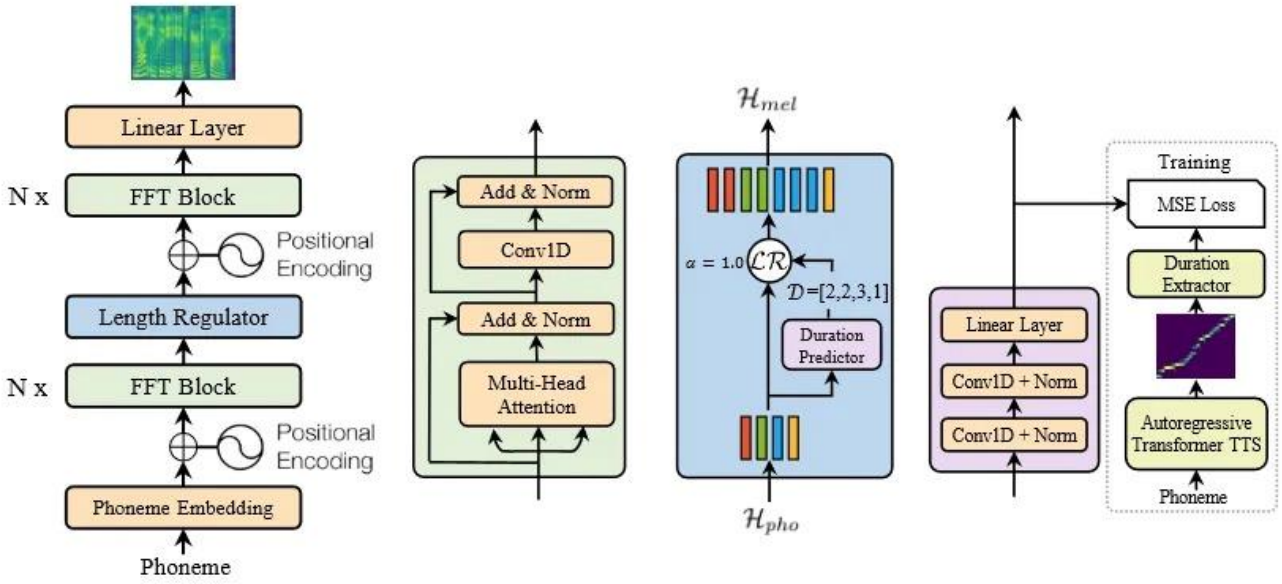



Un mel-espectrograma es una representación visual que captura la información acústica del audio. Sus componentes principales son:

1. **Eje Horizontal (Tiempo [s]):**
    - Representa la evolución temporal del audio. Ejemplo: un gráfico de 3.5 segundos.
2. **Eje Vertical (Frecuencia [Hz]):**
    - Representa las frecuencias en escala mel (logarítmica), que imita la percepción humana del sonido.
    - Rango típico: 512 Hz a 32,768 Hz.
3. **Colores (Intensidad en dB):**
    - Indican la energía o amplitud de las frecuencias:
        - **Rojo/Amarillo:** Mayor intensidad (sonidos fuertes).
        - **Verde/Azul:** Intensidad media/baja.
        - **Azul oscuro:** Baja o nula energía (silencio relativo).

## 3. Flujo de Información del Modelo TTS




### 3.1 Preprocesamiento

1. **Entrada:** Texto.
2. **Tokenización fonética:**
    - El texto se convierte en una secuencia de fonemas (las unidades mínimas de sonido).
    - Ejemplo:
        - Entrada: *"Hello, how are you?"*
        - Fonemas: [h, ə, l, oʊ, h, aʊ, ɹ, j, uː].

### 3.2 Embeddings

1. Cada fonema se transforma en un vector en un espacio n-dimensional.

2. Los vectores reflejan similitudes fonéticas: fonemas similares tienen representaciones cercanas.
    - Ejemplo:
        - **h →** [0.2, 0.5, …]
        - **ə →** [0.3, 0.7, …].

**3.3 Codificación Posicional**

- Como el modelo no es autoregresivo, carece de información del orden de los fonemas.
- Se suma una codificación posicional a los embeddings para incluir esta información.

**Resultado:** Representaciones enriquecidas que integran las características del fonema y su posición en la secuencia.

**3.4 Procesamiento con Bloques FFT (Feed-Forward Transformer)**

1. **Primer conjunto de bloques FFT:**
    - **Objetivo:** Modelar relaciones contextuales entre los fonemas.
    - Componentes:
        - **Atención Multi-Cabezal:**
            - Identifica relaciones globales entre los fonemas.
        - **Convoluciones 1D:**
            - Capturan relaciones locales (patrones entre fonemas cercanos).
        - **Conexiones Residuales y Normalización:**
            - Evitan pérdida de información y estabilizan el aprendizaje.

**Resultado:** Representaciones intermedias enriquecidas.

**3.5 Regulador de Longitud**

- **Función:** Ajusta la longitud de la secuencia de fonemas para coincidir con la duración del mel-espectrograma.
- **Componentes:**
    - **Predictor de Duración:** Predice cuántos frames de audio corresponden a cada fonema.
        - Durante el entrenamiento, las duraciones se extraen de un modelo TTS autoregresivo.
    - **Expansión Temporal:** Repite cada fonema según la duración predicha.

- **Velocidad ajustable:** Controlada por un factor α\alphaα (e.g., α=1.0\alpha = 1.0α=1.0 para velocidad normal).

### 3.6 Segundo Conjunto de Bloques FFT

- **Propósito:** Convertir la secuencia expandida en una representación más cercana al mel-espectrograma.

### 3.7 Generación del Mel-Espectrograma

1. **Capa Lineal:** Transforma las representaciones de los bloques FFT en un mel-espectrograma.
2. **Salida:** Una representación que describe las características acústicas del audio.

### 3.8 Conversión del Mel-Espectrograma a Audio

- Un **vocoder** convierte el mel-espectrograma en ondas sonoras reproducibles.
- Ejemplos de vocoders:
  - **Autoregresivos:** WaveNet.
  - **No autoregresivos:** WaveGlow, HiFi-GAN.

## 4. Tipos de Modelos TTS

### 4.1 Modelos Autoregresivos (AR)

- **Flujo:** Cada frame del espectrograma depende de los anteriores.
- **Ejemplo:** Tacotron.
- **Ventajas:**
  - Alta calidad y naturalidad.
  - Manejan bien dependencias de largo alcance.
- **Desventajas:**
  - Lentitud debido a la generación secuencial.
  - Errores acumulativos.

### 4.2 Modelos No Autoregresivos (NAR)

- **Flujo:** Los frames se generan en paralelo.
- **Ejemplo:** FastSpeech.
- **Ventajas:**
  - Mayor velocidad, ideal para tiempo real.

    - Estabilidad (menos errores acumulativos).
- **Desventajas:**
    - Menor precisión en prosodia y dependencias temporales.

**4.3 Modelos Híbridos**

- Combinan aspectos de ambos enfoques.
- **Ejemplos:** FastSpeech 2, ParaNet.
- **Ventajas:**
    - Compromiso entre calidad y velocidad.
    - Mejor control sobre prosodia.
- **Desventajas:**
    - Más complejos de implementar.

**5. Relación con los Vocoders**

- **Autoregresivos:** Alta calidad pero más lentos (e.g., WaveNet).
- **No autoregresivos:** Más rápidos, ideales para tiempo real (e.g., HiFi-GAN).

**6. Resumen del Flujo de un Modelo TTS No Autoregresivo**

1. **Entrada:** Texto → Fonemas → Embeddings → Codificación Posicional.
2. **Procesamiento:** Bloques FFT iniciales → Regulador de Longitud → Bloques FFT finales.
3. **Generación:** Mel-espectrograma → Vocoder → Audio.

**7.Conclusiones y recomendaciones**

**-** El fine tuning del modelo tiene carencias en la calidad de audio, se recomienda mejorar el modelo entrenando con más datos.

- En general, se cumplió con los objetivos que eran probar y entrenar un modelo de texto a voz