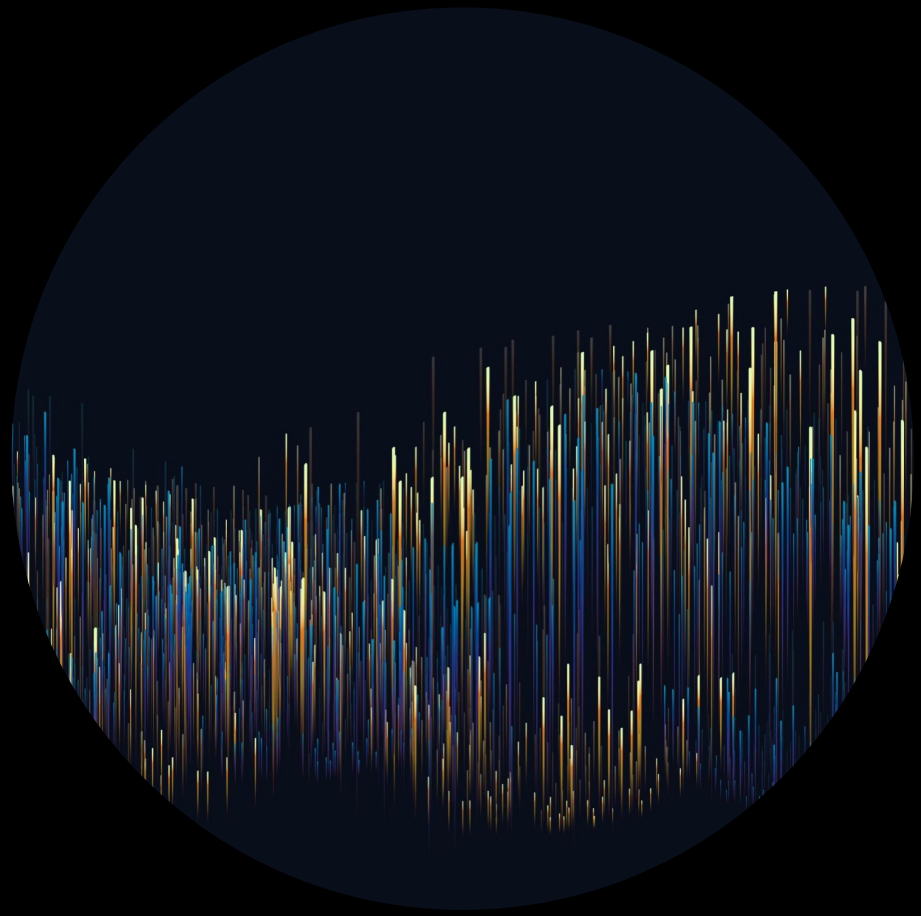
WORDWAVE
AN ENHANCED AUDIO LANGUAGE MODELING FOR A TEXT-TO-SPEECH SYSTEM

By:

Hansika Yedlapalli



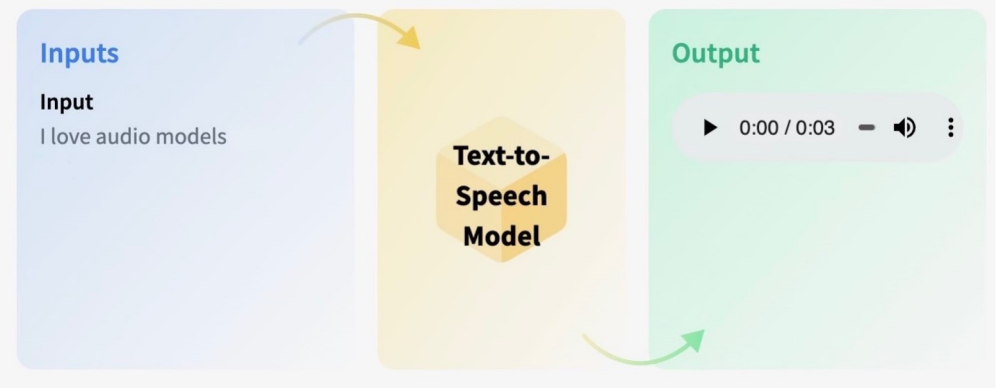
ROADMAP

Introduction and Understanding of TTS systems

Model & Methodology

Experiments & Results

WHAT IS TTS?


Inputs
Input
I love audio models
Text-to-Speech Model
Output
0:00 / 0:03

APPLICATIONS OF TTS SYSTEMS

ACCESSIBILITY TOOLS

VIRTUAL ASSISTANTS

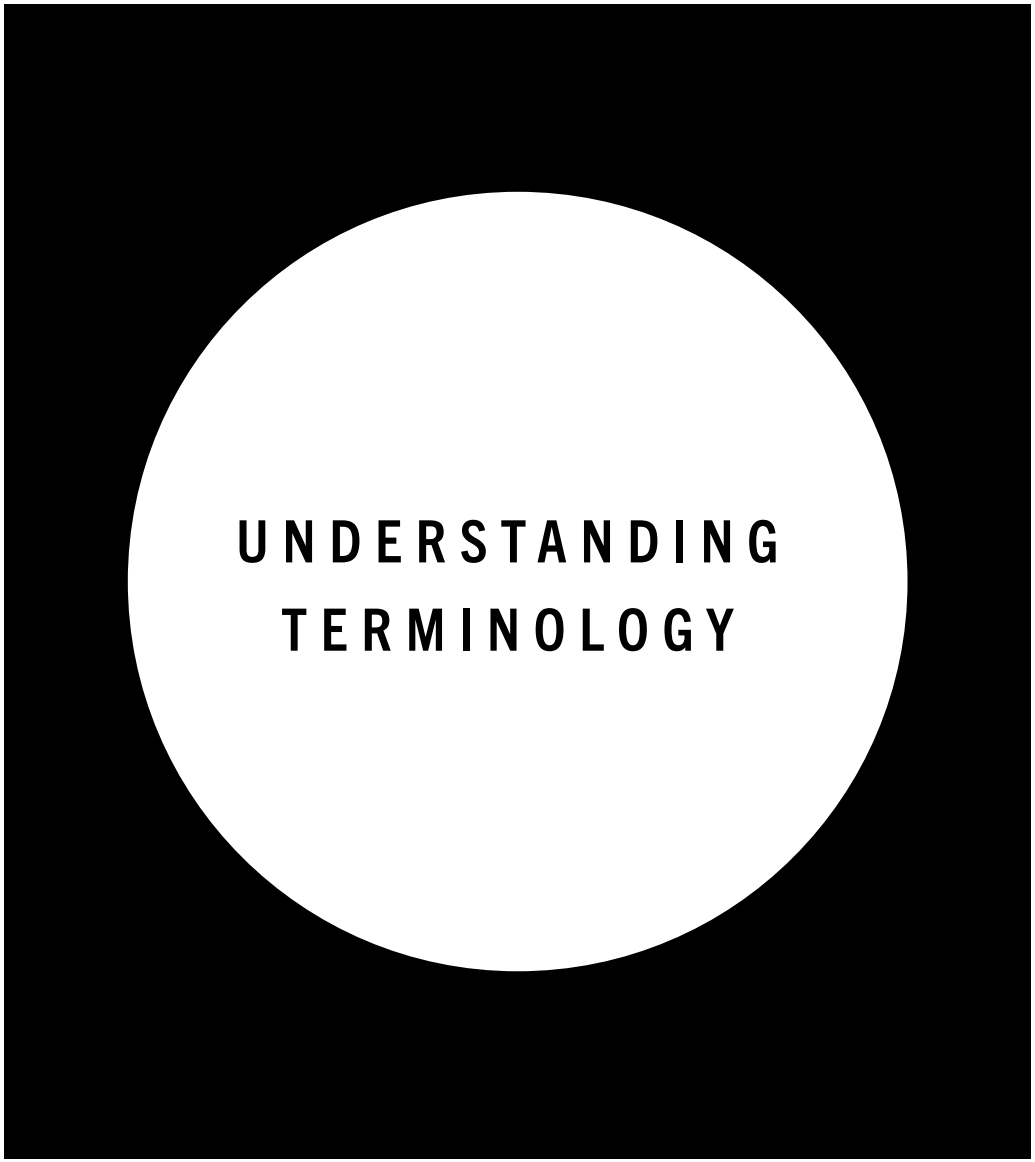
NAVIGATION SYSTEMS

LANGUAGE LEARNING

E-LEARNING AND EDUCATION

ENTERTAINMENT AND MEDIA

CUSTOMER SERVICE

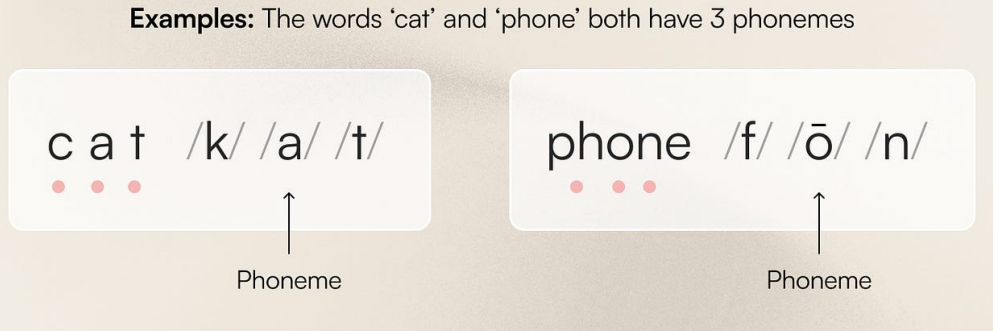


The Basic Terminology we need to learn are:

1. Phoneme
2. Prosody
3. Audio
4. Mel-spectrogram

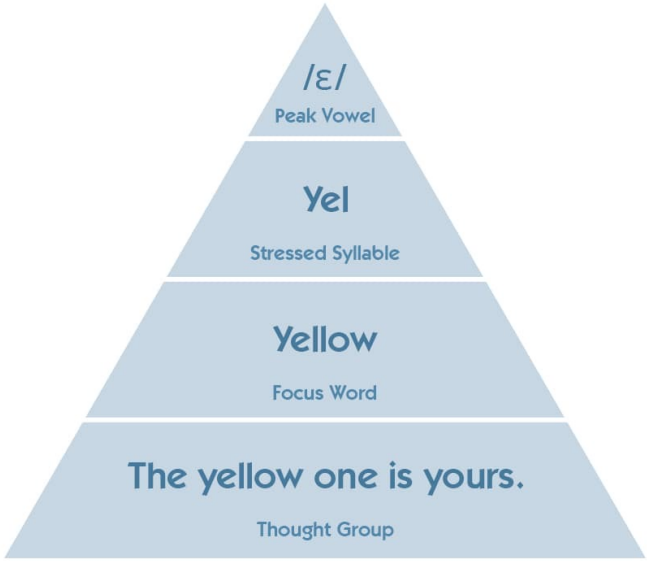
PHONEME

Phoneme is the smallest unit of sound that distinguishes one word from another in a language.



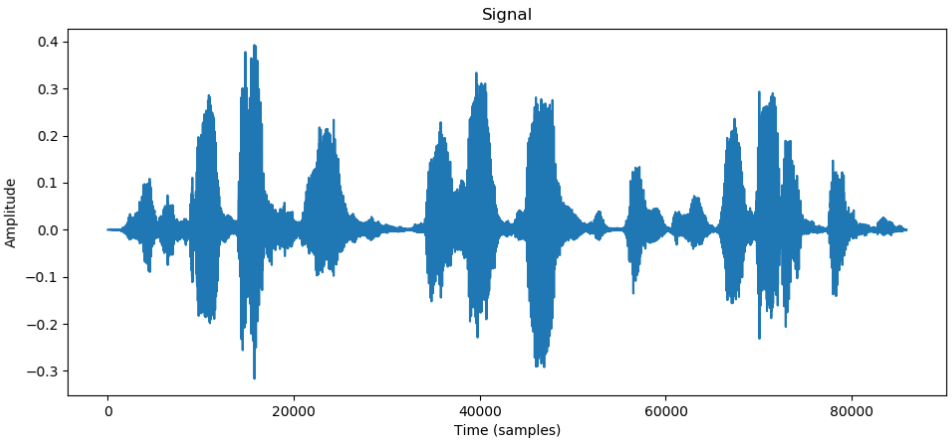
PROSODY

Prosody refers to the patterns of rhythm, intonation, and stress used in speech.



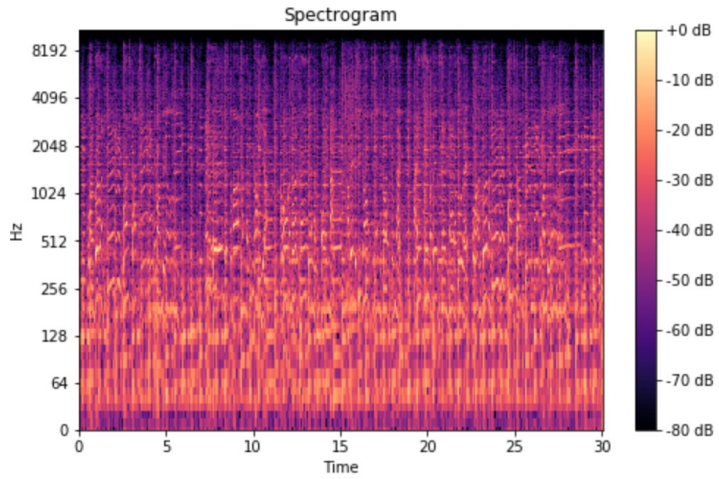
AUDIO

- Computers understand audio through sampled air pressure, converted into numerical data.
- Sampling rate commonly set at 16 kHz, capturing sequences of 16,000 amplitudes per second.
- Fourier transform helps extract information from the vast set of amplitudes. The Short-Time Fourier Transform (STFT) divides audio into shorter segments for analysis.
- STFT segments the signal into fixed intervals, controlling overlap and non-intersecting portions.
- Output represents amplitude in dB scale, providing insights into frequency and time.

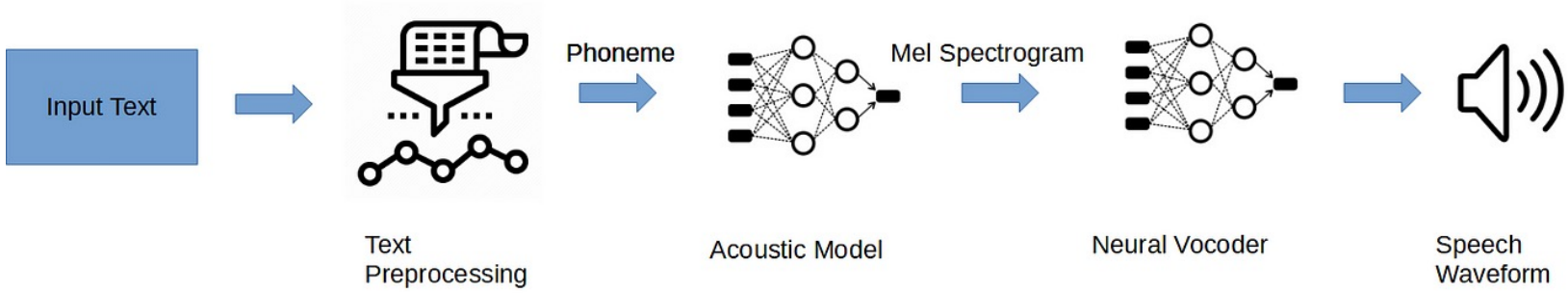


MEL SPECTROGRAM

- Helps TTS systems capture subtle sound differences and offers a practical way to analyze and generate speech.
- Mel Spectrogram is the representation of audio data in a way that matches how humans hear.
- Mel Scale: Adjusts frequencies to align with human perception. By using this, TTS systems can better capture nuances in speech.
- In a Mel Spectrogram, frequency is on the y-axis, time on the x-axis, and intensity shows sound amplitude.



BASIC ARCHITECTURE OF A TTS SYSTEM



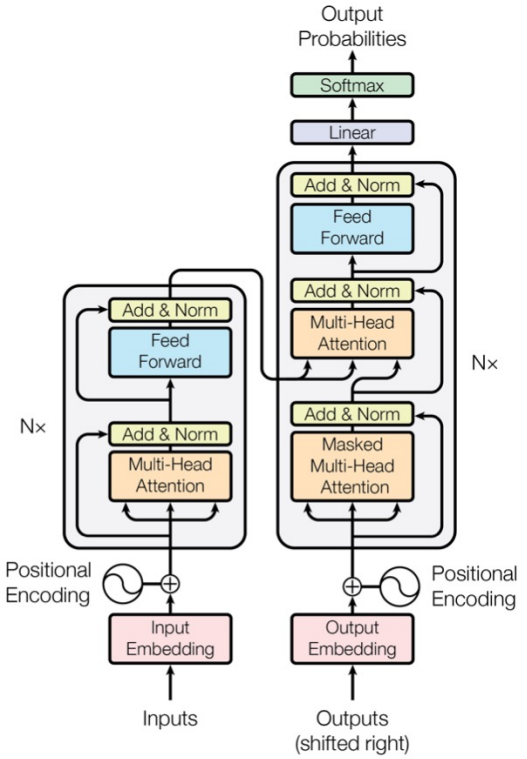
- **Text Preprocessing:** Extracts linguistic features from input text.
- **Acoustic Model:** Maps linguistic features to acoustic representations.
- **Vocoder:** Transforms acoustic features into speech waveform.

MODEL & METHODOLOGY

TEXT ANALYSIS

TRANSFORMERS

- Key Components:
  - **Self-Attention Mechanism**: Allows each word in the input sequence to attend to all other words, capturing long-range dependencies efficiently.
  - **Multi-Head Attention**: Enables the model to focus on different parts of the input simultaneously, enhancing its ability to learn complex relationships.
  - **Positional Encoding**: Provides information about the position of each word in the input sequence, overcoming the lack of sequential order in traditional neural networks.
- Advantages of Transformers:
  - **Parallelization**: Self-attention mechanism enables parallel processing of input tokens, leading to faster training and inference.
  - **Scalability**: Transformers can handle inputs of variable length, making them suitable for a wide range of NLP tasks.
  - **State-of-the-Art Performance**: Transformers have achieved remarkable results on various NLP benchmarks, surpassing traditional models in terms of accuracy and efficiency.



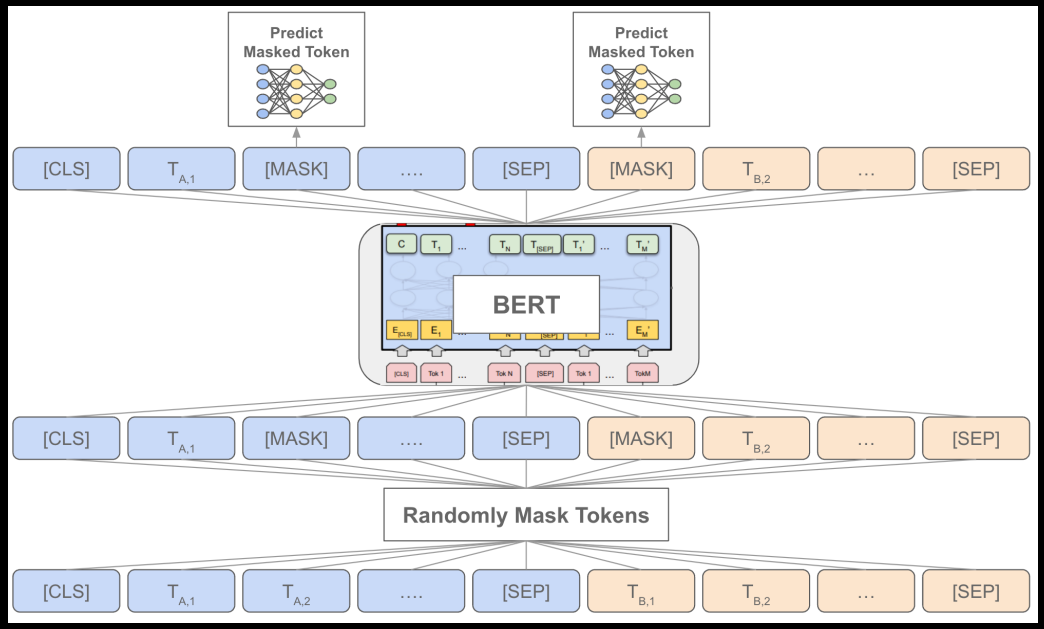
BERT - PRETRAINING

Masked Language Modeling (MLM):

- BERT randomly masks 15% of the tokens in each sequence.
- The model predicts the original value of the masked words based on context provided by other non-masked words.
- Enables bidirectional understanding of language, enriching contextual understanding.

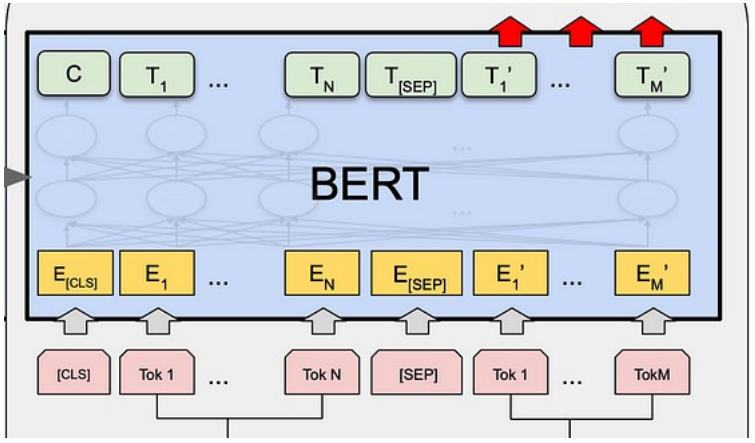
Next Sentence Prediction (NSP):

- BERT learns to predict if the second sentence logically follows the first in pairs of sentences.
- Enhances understanding of relationships between consecutive sentences.
- Beneficial for tasks requiring comprehension of sentence relationships, like question answering and natural language inference.

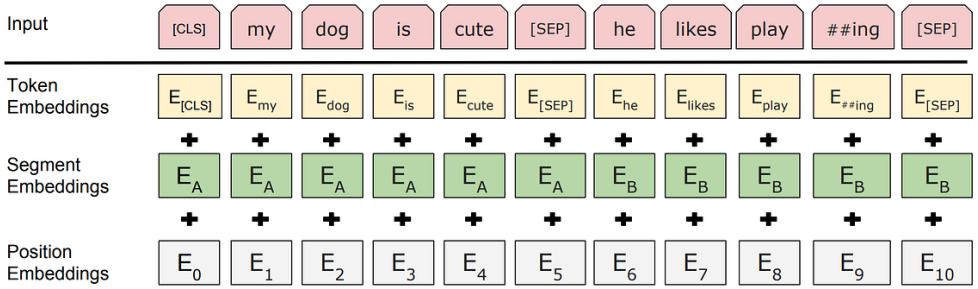


BERT – FINETUNING

- In this phase, BERT is adapted to specific tasks using labeled task-specific data, refining its performance.
- Fine-tuning Process:
  - Input Task-Specific Data: Provide BERT with labeled examples relevant to the downstream task.
  - Adjust Model Parameters: Update BERT's parameters during training to optimize performance for the specific task.
  - Transfer Knowledge: BERT leverages its pretrained knowledge to learn task-specific patterns from the labeled data.
  - Task-Specific Layers: Optionally, add task-specific layers on top of BERT to tailor it further to the task.



INPUT EMBEDDINGS



Tokenization:

- Raw textual data is segmented into individual tokens, representing words or sub-words.

Special Token Insertion:

- BERT's input sequence begins with a [CLS] token and ends with a [SEP] token.
- Additional [SEP] tokens are added between consecutive sentences.

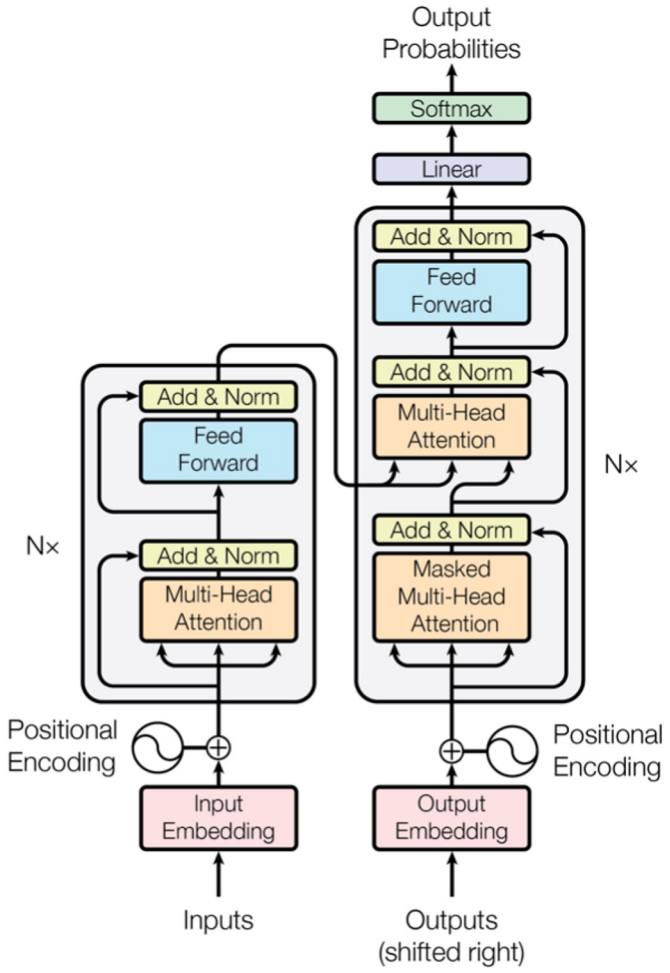
Embedding:

- Each token is converted into its corresponding Word Piece embedding vector.

Additive Embeddings:

- Learnable embeddings are added to each token vector to indicate its position in the sequence and sentence boundaries.
- This step ensures self-attention can distinguish each token's position and relationship within the sequence.

UNDERSTANDING TEXT ANALYSIS



SPEECH ANALYSIS

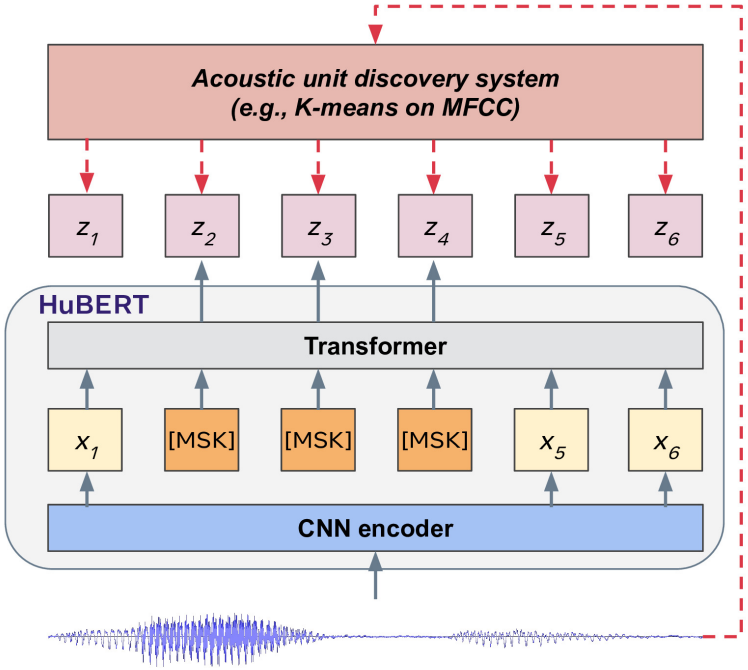
WHY CHOOSE HUBERT?

- Existing model: The existing system relies on model wav2vec 2.0, utilizing CNNs and transformers. While effective, it struggles with capturing fine-grained acoustic details and long-range dependencies.
- HuBERT: This represents a novel approach that enhances traditional methods through advanced clustering techniques. These techniques contribute to improved speech representation learning.
- Advantages:
  - Improved Representation: HuBERT's clustering mechanisms excel in capturing fine-grained acoustic details and long-term dependencies, surpassing wav2vec 2.0.
  - Enhanced Flexibility: Through iterative refinement of cluster assignments and cluster ensembles, HuBERT offers adaptability across diverse speech characteristics.
  - Robust Performance: Experimental results consistently demonstrate HuBERT's superiority over wav2vec 2.0 in various speech-related tasks.

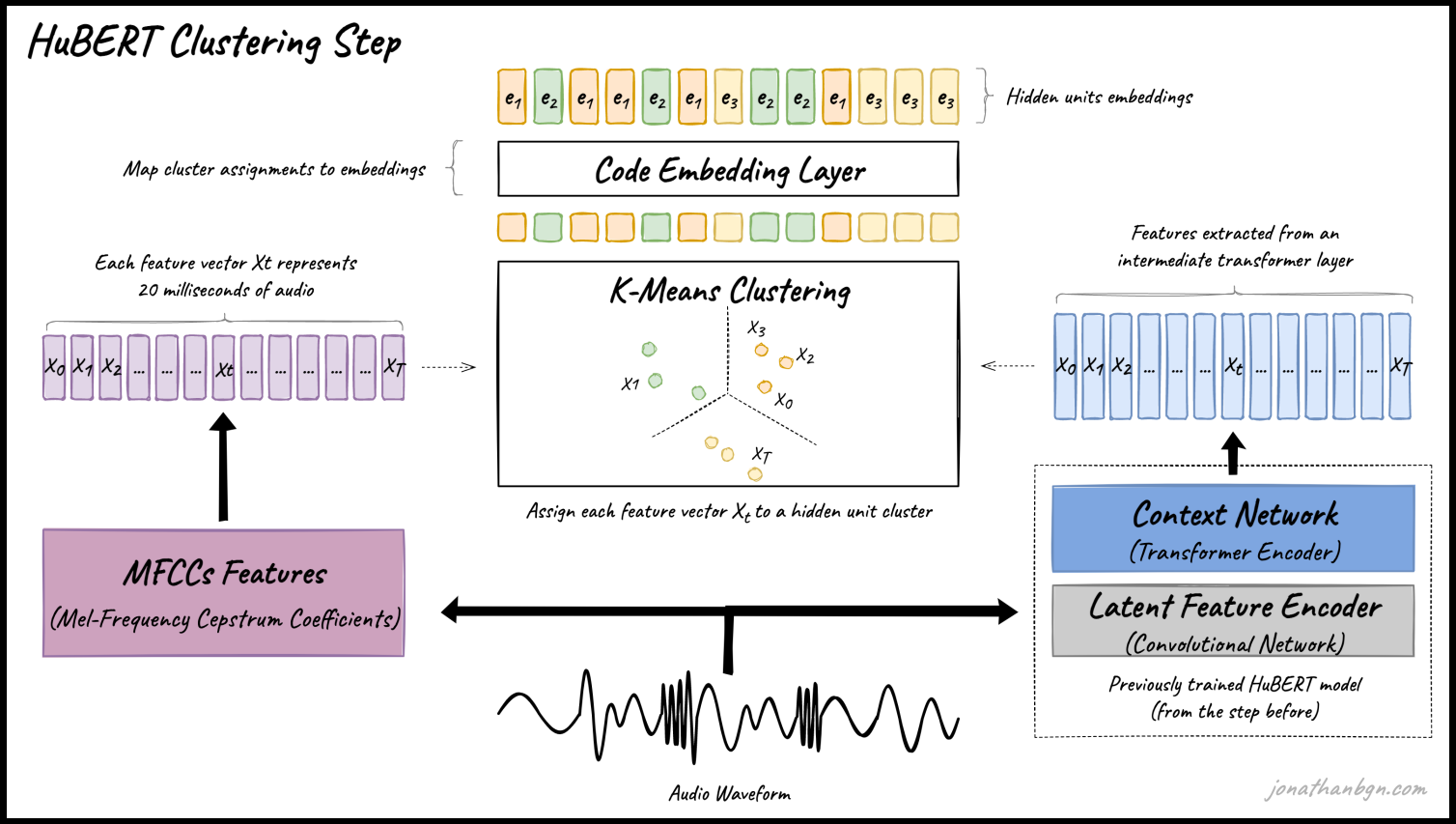
Note: ref https://arxiv.org/pdf/2308.03226.pdf

HUBERT ARCHITECTURE

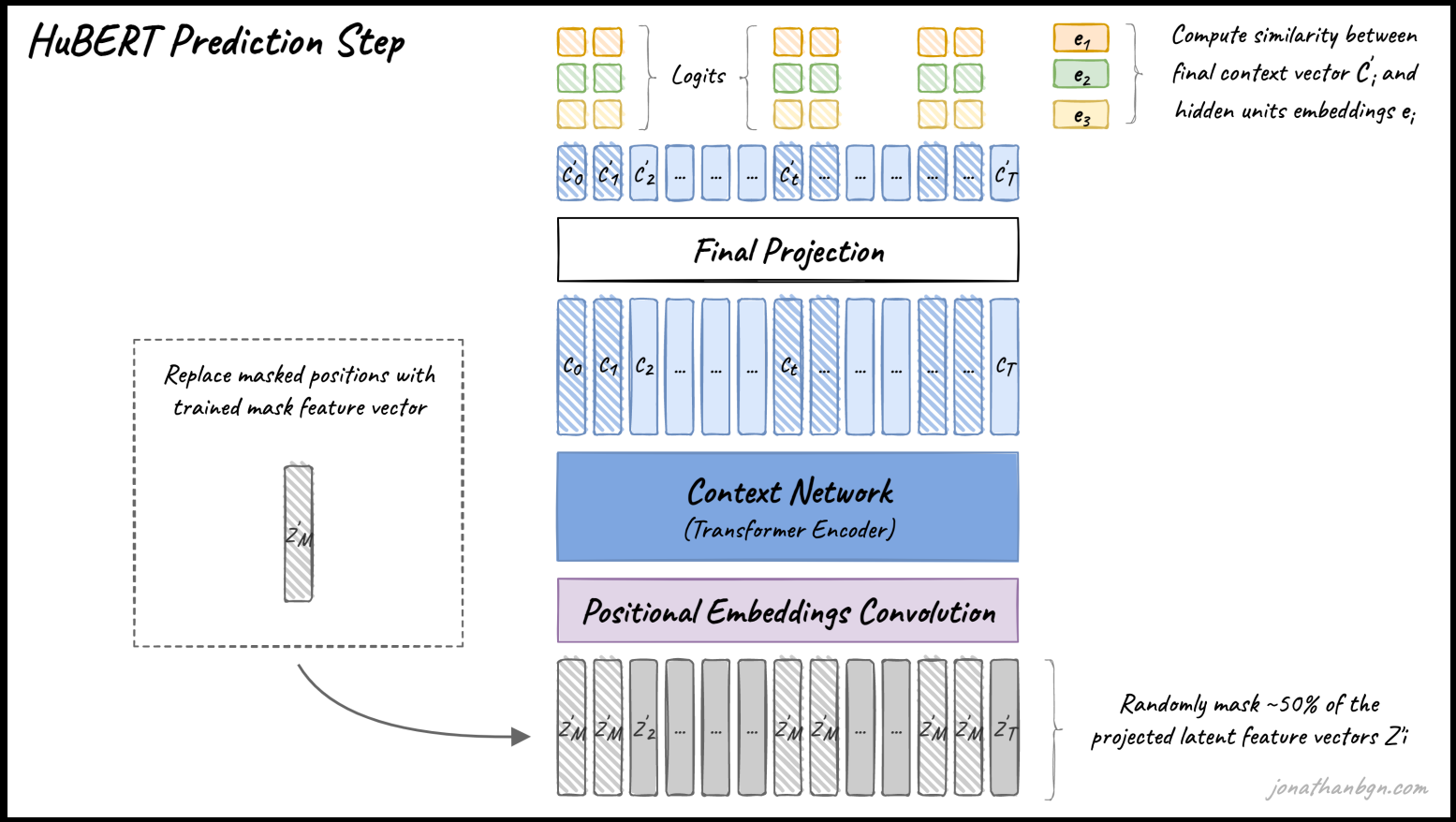
- HuBERT has two main objectives:
  - Acoustic modeling: Converting audio inputs into continuous latent representations.
  - Capturing temporal dependencies: Understanding how these representations evolve over time.
- It encodes unmasked audio inputs into continuous latent representations to address the acoustic modeling challenge.
- The model emphasizes the importance of consistency in the k-means mapping for clustering, enabling it to focus on sequential structure understanding.
- Iterative improvement occurs through alternating clustering and prediction steps, with early iterations informing subsequent clustering for better representation.



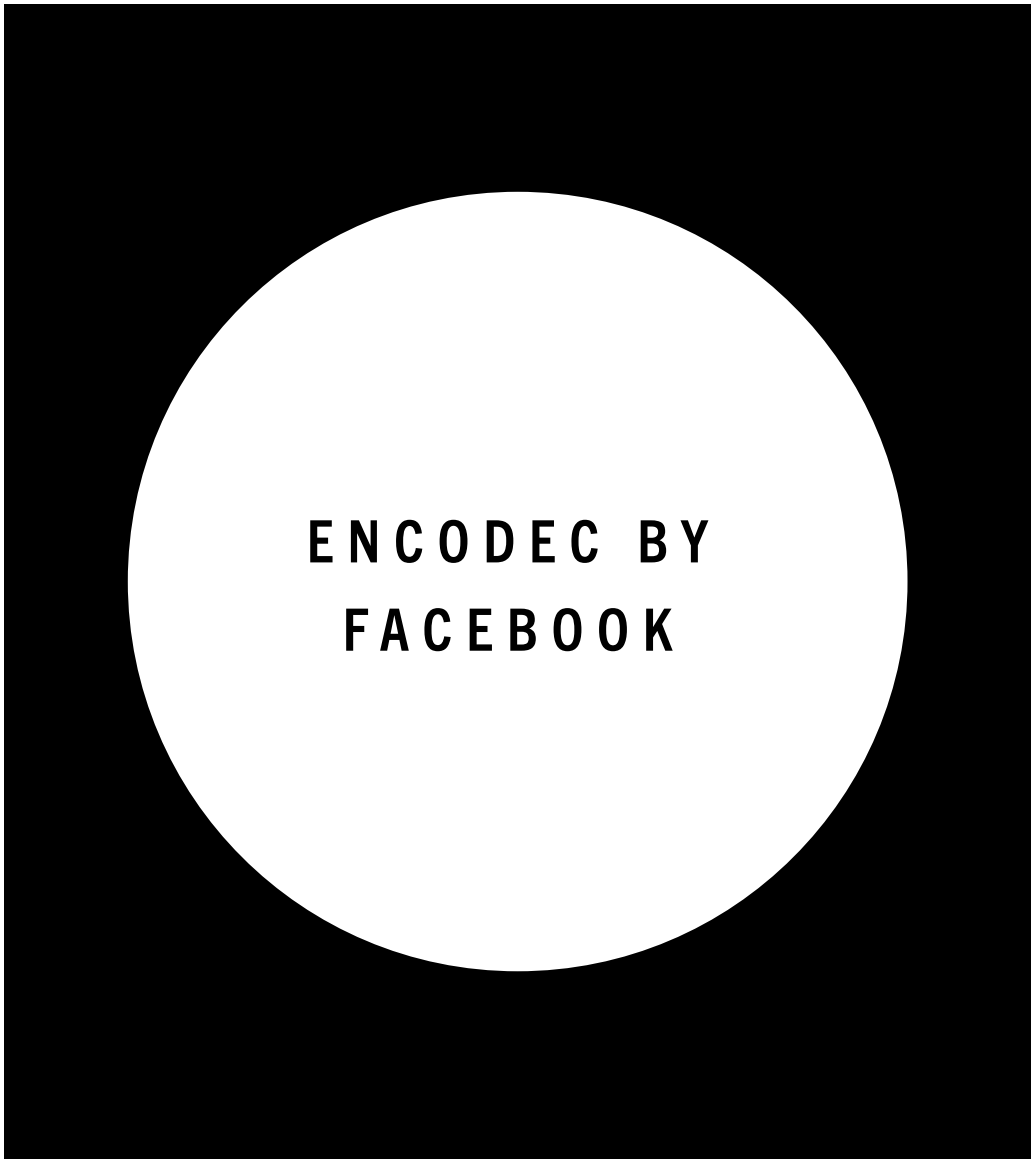
HUBERT CLUSTERING



HUBERT PREDICTION



AUDIO GENERATION



EnCodec generates high-quality speech waveforms directly from linguistic features.

What EnCodec Offers:

- Seamless Integration: EnCodec seamlessly integrates into TTS pipelines, serving as a crucial component for synthesizing natural-sounding speech.
- High Fidelity: It produces speech waveforms with high fidelity and naturalness, capturing nuances in tone, intonation, and pronunciation.
- Low Latency: EnCodec operates efficiently, minimizing latency in real-time speech synthesis applications.
- Flexibility: The architecture of EnCodec allows for easy customization and adaptation to various languages, dialects, and speech styles.

WHY? ENCODEC

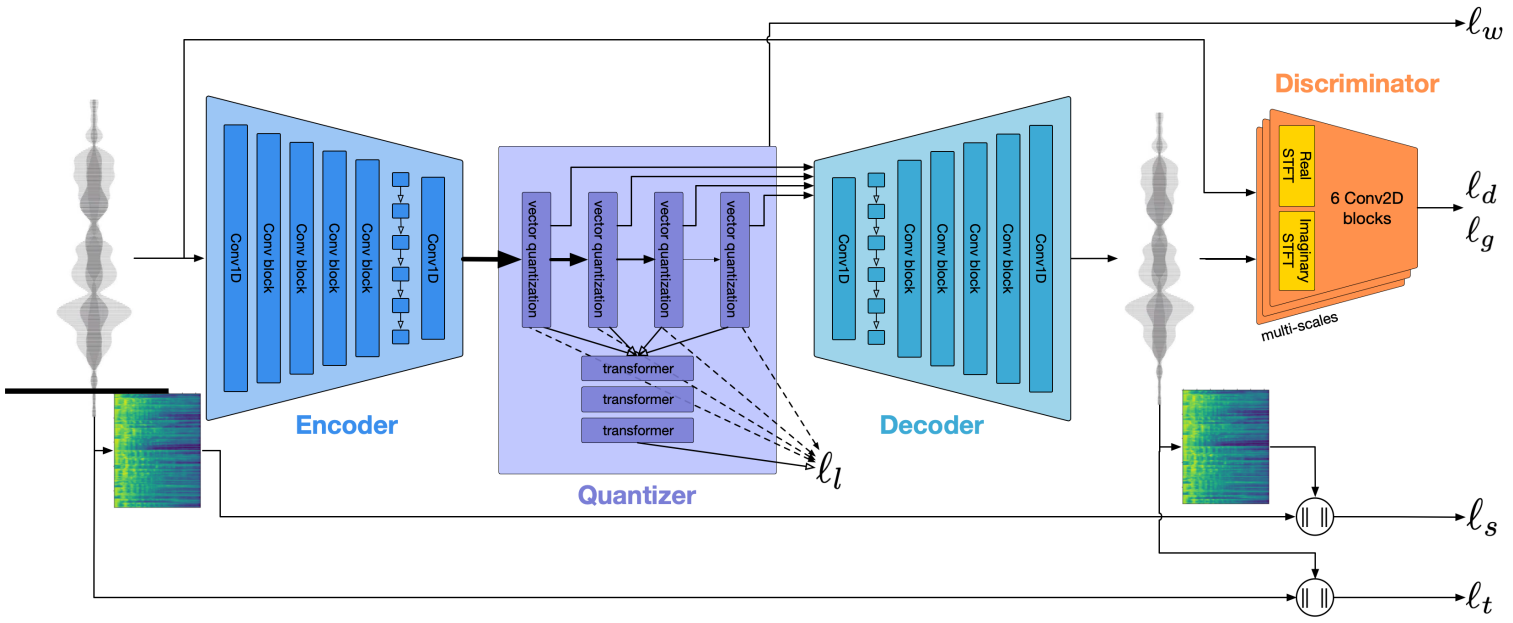
Superior Quality: EnCodec delivers superior speech quality compared to traditional vocoders, ensuring a more engaging and lifelike user experience.

State-of-the-Art Technology: Developed by Facebook Research, EnCodec represents the latest advancements in neural vocoder research, offering cutting-edge performance.

Robust Support: With ongoing updates and support from Facebook, EnCodec ensures reliability and scalability for your TTS project.

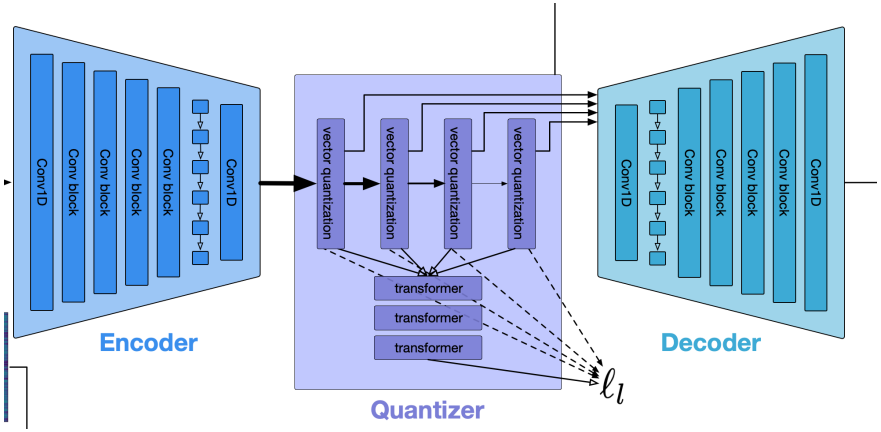
Enhanced User Experience: By incorporating EnCodec into your TTS system, you can provide users with an enhanced auditory experience, making interactions more immersive and engaging.

ENCODEC ARCHITECTURE

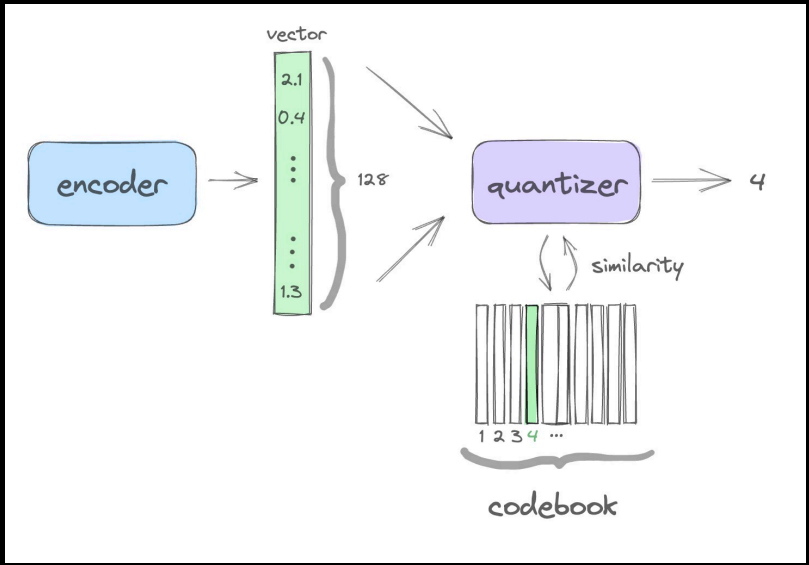


ENCODEC

- EnCodec architecture consists of an encoder, quantizer, and decoder, trained simultaneously.
- The encoder converts audio samples into fixed-dimensional vectors.
- The quantizer compresses encoded vectors using Residual Vector Quantization (RVQ).
- The decoder reconstructs compressed signals into audio streams.

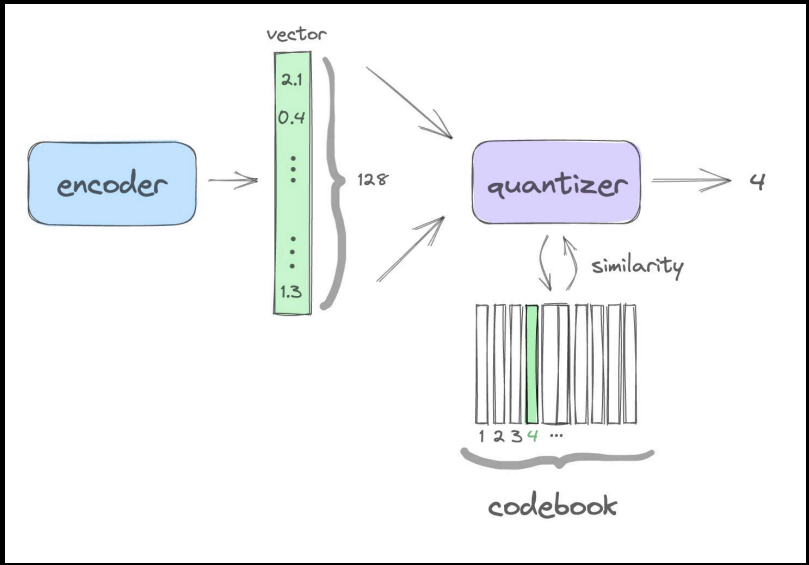


QUANTIZER



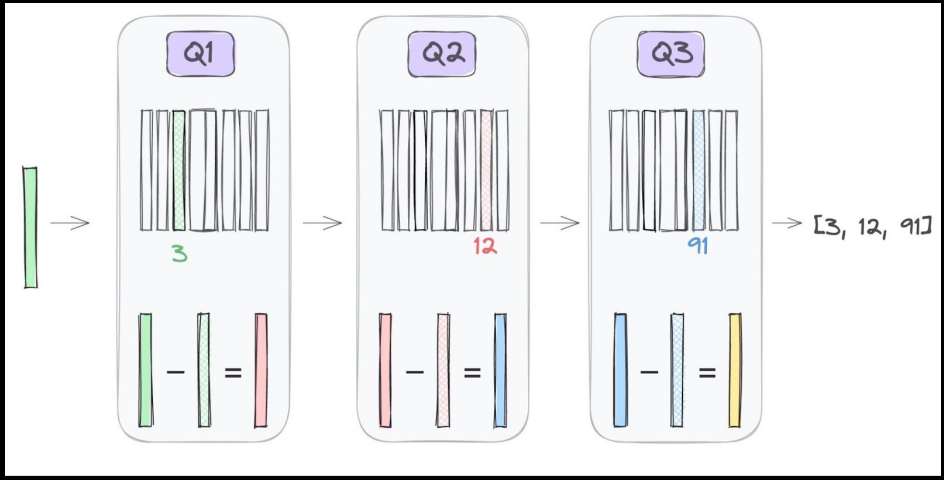
- The quantizer operates by comparing the input vector, which is typically a high-dimensional representation of an audio sample, with vectors in a codebook.
- For example, consider an audio sample transformed by the encoder into a vector of 128 dimensions.
- In the quantization process, the input vector is compared with vectors stored in the codebook table.
- The quantizer then selects the index of the vector in the codebook that is most similar to the input vector.
- This compression stage effectively reduces the original vector of 128 numbers to a single number, representing the index in the codebook.

QUANTIZER



- The quantizer operates by comparing the input vector, which is typically a high-dimensional representation of an audio sample, with vectors in a codebook.
- For example, consider an audio sample transformed by the encoder into a vector of 128 dimensions.
- In the quantization process, the input vector is compared with vectors stored in the codebook table.
- The quantizer then selects the index of the vector in the codebook that is most similar to the input vector.
- This compression stage effectively reduces the original vector of 128 numbers to a single number, representing the index in the codebook.

RESIDUAL VECTOR QUANTIZATION (RVQ)



It employs a cascade of codebooks to provide a progressively finer approximation of the input vectors.

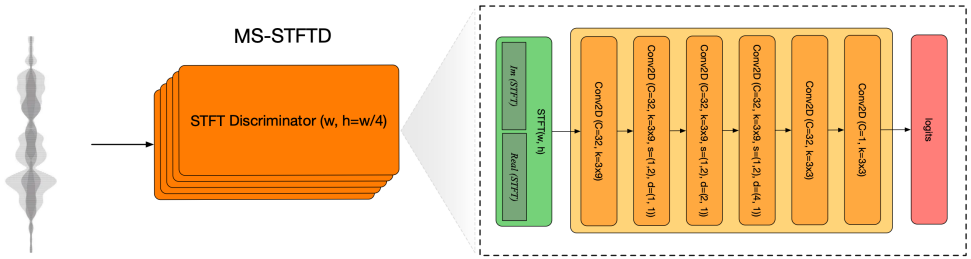
Layered Approach:

- RVQ breaks down the quantization process into multiple layers, each addressing the residual error from the preceding layer.
- This layered approach enables scalability, allowing the system to operate across different bitrates by adjusting the number of layers.

Efficiency and Precision:

- By breaking down the quantization problem into layers, RVQ achieves high precision with reduced computational costs.
- Rather than attempting to quantize a high-dimensional vector with a single large codebook, RVQ distributes the task across multiple stages, improving efficiency.

MS-STFTD DISCRIMINATOR



The Multi-Scale Short-Time Fourier Transform Discriminator (MS-STFTD) identifies real and generated audio samples by analyzing spectral and temporal features.

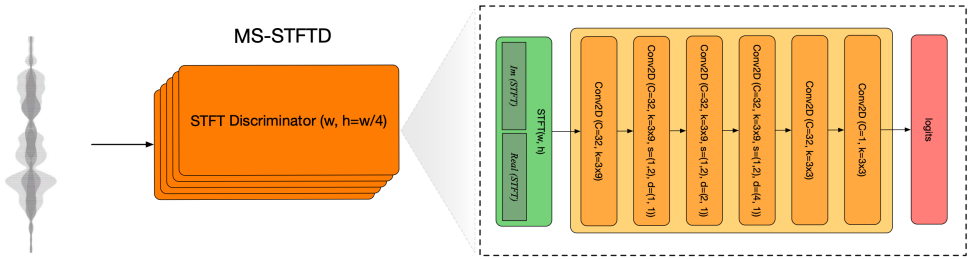
Components of (MS-STFTD):

- Applies multi-scale STFT to capture spectral features at various resolutions.
- Employs convolutional layers for feature extraction.

Convolutional Neural Network (CNN):

- Processes multi-scale STFT representations to extract discriminative features.
- Learns hierarchical representations of input spectrograms.

MS-STFTD DISCRIMINATOR



Input Spectrogram:

- Converts audio samples into spectrograms using STFT.
- Multi-Scale Analysis:
- Utilizes STFT with varying window sizes for multi-scale analysis.
- Convolutional Processing:
- Extracts discriminative features using convolutional layers.

Output:

- The MS-STFTD Discriminator provides a probability score indicating the likelihood of the input spectrogram belonging to the original audio distribution.
- High scores suggest real audio, while low scores indicate generated audio.

CONCLUSION

Integration of BERT, HuBERT, and Facebook's EnCodec offers a comprehensive solution for text-to-speech synthesis.

Semantic Understanding:

BERT and GPT excel at understanding text semantics, enabling accurate tokenization and semantic modeling.

Acoustic Representation Learning:

HuBERT specializes in self-supervised representation learning for speech, facilitating the conversion of semantic tokens to coarse acoustic tokens.

Fine Acoustic Detailing:

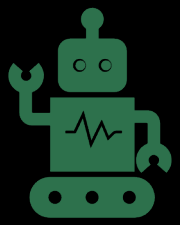
Facebook's EnCodec handles the transformation from coarse to fine acoustic tokens, capturing intricate acoustic details for high-quality audio synthesis.

By leveraging the strengths of each model, the synthesis pipeline achieves efficiency in natural-sounding synthesized speech.

FUTURE DIRECTIONS

Multimodal Integration

Explore the integration of visual and textual inputs to enable multimodal speech synthesis, catering to applications like video captioning and augmented reality.



Real-Time Speech Synthesis

Develop optimization strategies to enable real-time speech synthesis, facilitating applications like virtual assistants and live captioning.

User Interaction and Personalization

Investigate methods for user interaction and personalization, allowing users to clone voices or customize the synthesized speech based on preferences and context.

DEMO

THANK YOU!