# COMP0087

## Statistical Natural Language Processing

## Lecture 8 – Neural Speech Understanding

Slides build on various resource including:

- Jinming Zhao
- Shinji Watanabe
- Hung-yi Lee
- Abdelrahman Mohamed

# Overview

1. **Introduction to speech**
2. Key speech tasks
3. Feature extraction in speech
4. Automatic speech recognition
5. Speech translation
6. Pre-trained speech encoder (wav2vec2, whisper)
7. AudioLLMs
8. Benchmarks

# Speech

Speech is far more complex compared to text, this is rooted in many factors including:

**Continuous vs. Discrete Inputs**

- Text: discrete tokens, finite vocabulary
- Speech: continuous waveform, infinite possibilities
- Raw audio: 16kHz = 16,000 samples/second

**No Natural Segmentation**

- Text has clear token boundaries
- Speech is continuous
- Where does one phoneme end and another begin?

**High Variability**

Same word, infinite pronunciations:

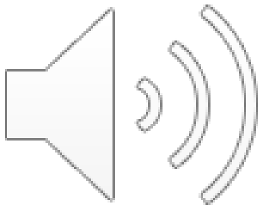
- Different speakers (gender, age, accent)
- Speaking rate (fast vs. slow)
- Emotion (angry vs. calm)
- Channel effects (phone vs. studio mic)
- Background noise

Text is relatively invariant

# Speech

- What is unique to speech that is not present in text?
  - Speech contains more than just content

*speech*



*Text: That doesn't matter*

*content*
*speaker info*
*emotion*
*prosody*
*noise*
etc.

*content*

# Overview

1. Introduction to speech
2. **Key speech tasks**
3. Feature extraction in speech
4. Automatic speech recognition
5. Speech translation
6. Pre-trained speech encoder (wav2vec2, whisper)
7. AudioLLMs
8. Benchmarks

# Speech Tasks

- Automatic Speech Recognition (ASR)
  - Speech-to-text [e.g., caption services]
- Speech Synthesis
  - Text-to-speech [e.g., audiobooks]
- Spoken Dialog Systems
  - Interaction through speech [e.g., Siri]
- Speech Translation
  - Speech-to-text (foreign language) [e.g., travel]

# Speech Tasks

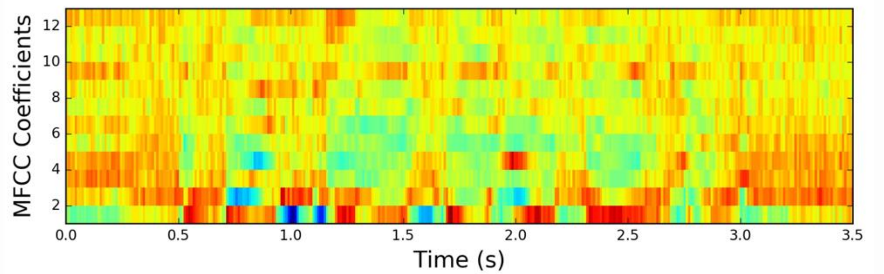
- **Automatic Speech Recognition (ASR)**
  - Speech-to-text [e.g., caption services]
- Speech Synthesis
  - Text-to-speech [e.g., audiobooks]
- Spoken Dialog Systems
  - Interaction through speech [e.g., Siri]
- **Speech Translation**
  - Speech-to-text (foreign language) [e.g., travel, UN speeches]

# Overview

1. Introduction to speech
2. Key speech tasks
3. **Feature extraction in speech**
4. Automatic speech recognition
5. Speech translation
6. Pre-trained speech encoder (wav2vec2, whisper)
7. AudioLLMs
8. Benchmarks

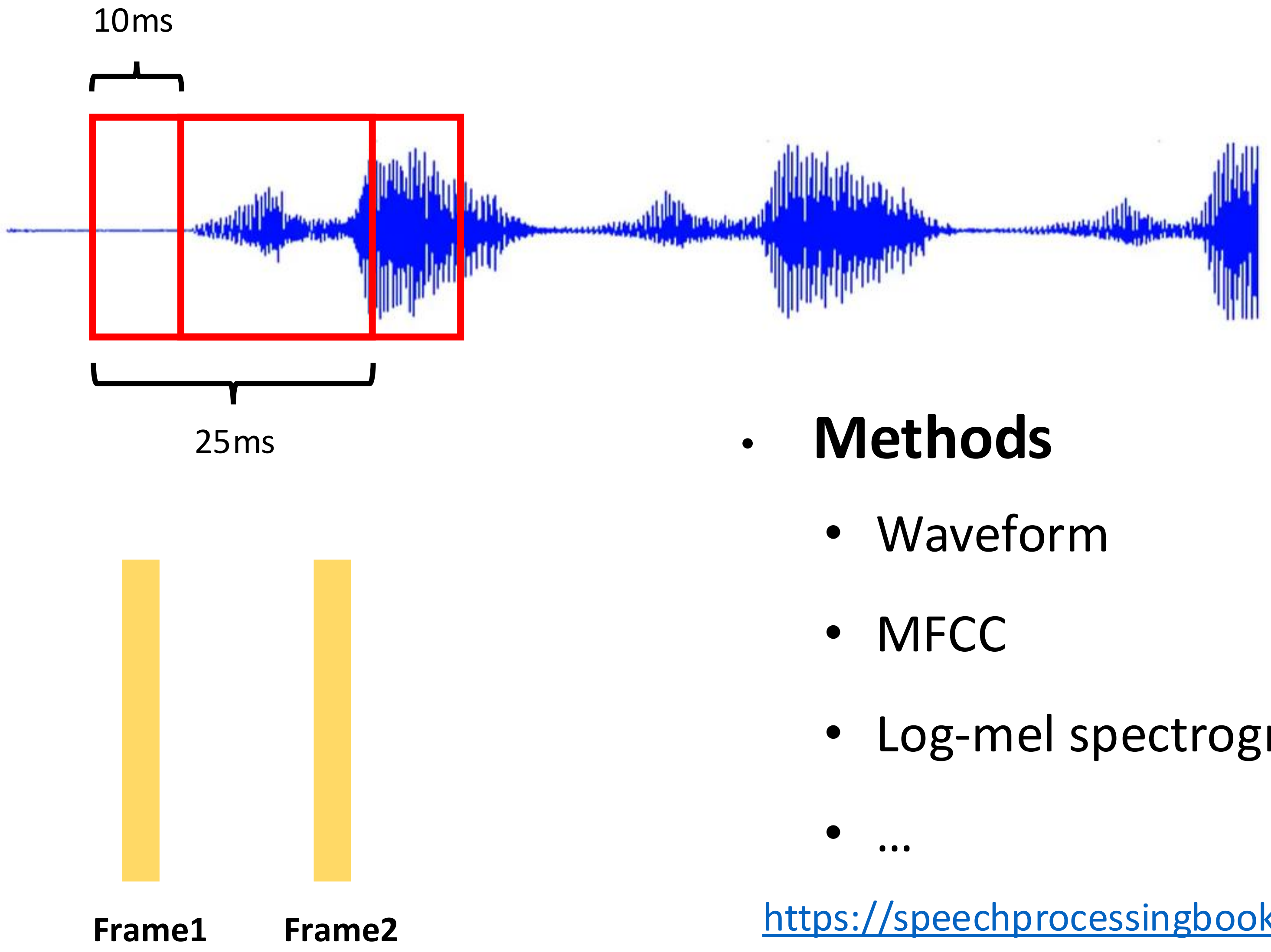
# Audio Data – feature extraction

MFCC vs Time




- Methods of extracting features
  - Signal processing
    - E.g., Mel-frequency cepstrum coefficients (MFCC), log-mel-spectrogram
    - Two dimensions: temporal and feature
    - Features can be pre-computed.
  - Deep learning
    - Pretrained encoders, e.g.,wav2vec2
    - (covered later)

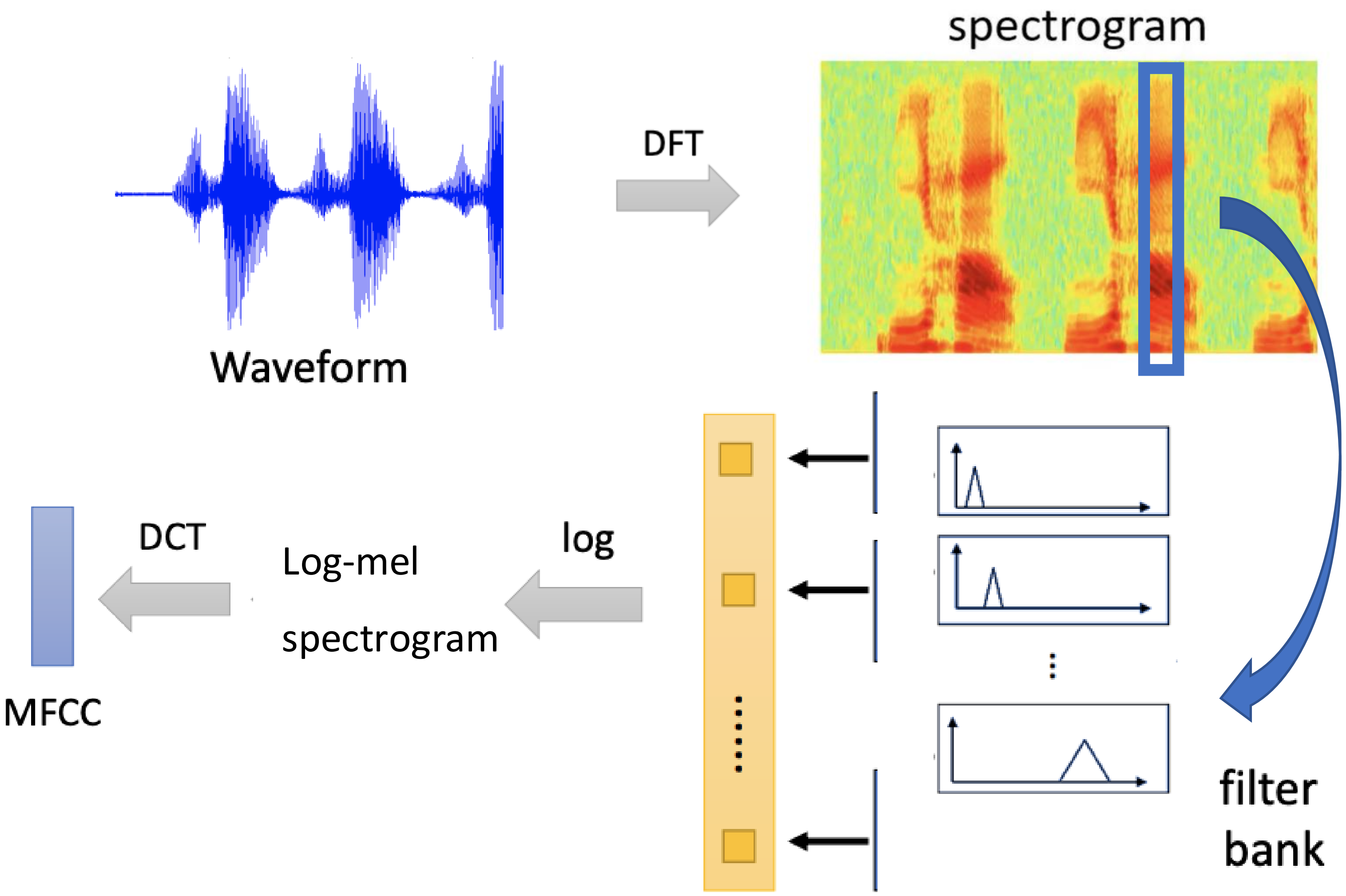
# Acoustic features




- **Methods**
  - Waveform
  - MFCC
  - Log-mel spectrogram
  - …

https://speechprocessingbook.aalto.fi/Representations/Representations.html

# Acoustic features

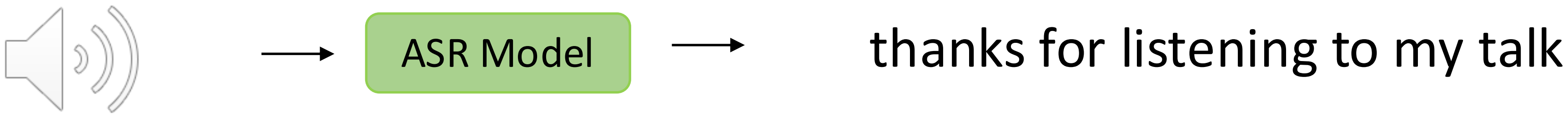


# Overview

1. Introduction to speech
2. Key speech tasks
3. Feature extraction in speech
4. **Automatic speech recognition**
5. Speech translation
6. Pre-trained speech encoder (wav2vec2, whisper)
7. AudioLLMs
8. Benchmarks

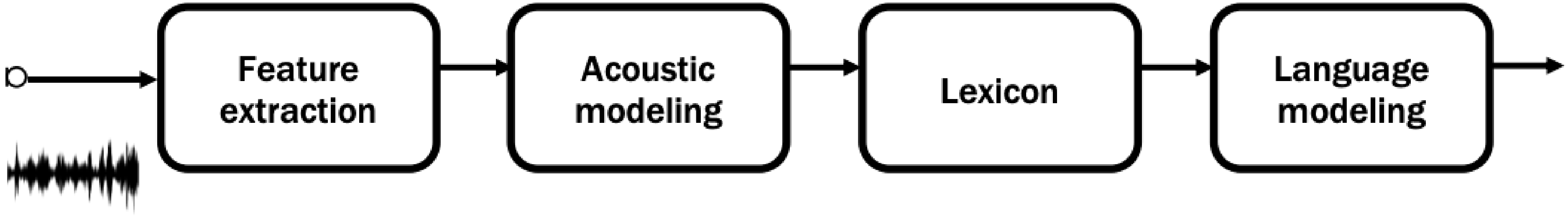
# Automatic speech recognition (ASR)

- Speech to text

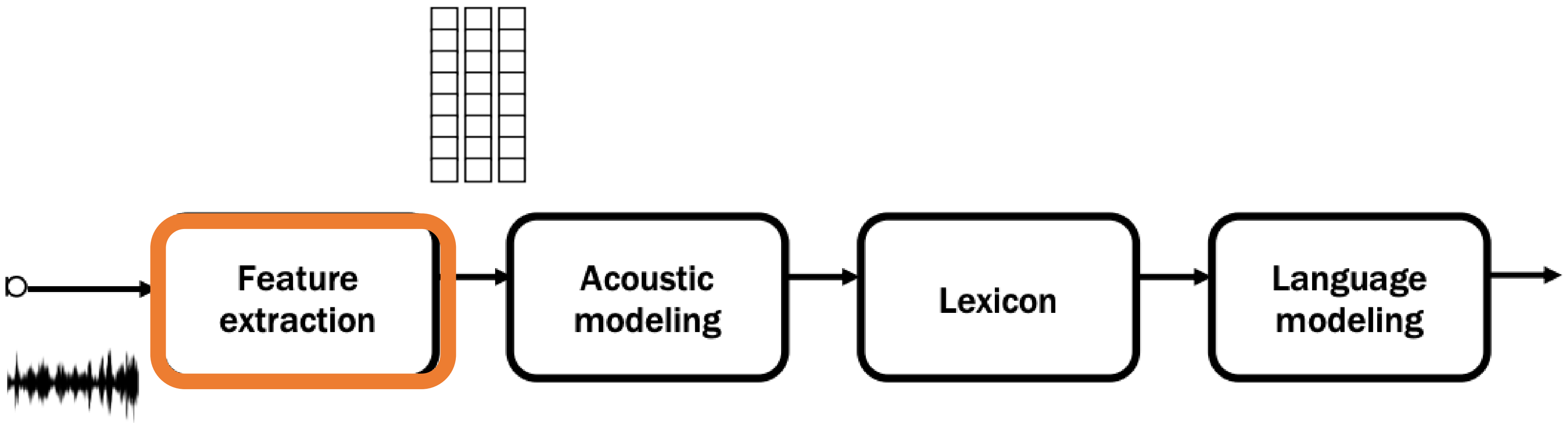



- Diverse applications, e.g., transcription services, voice assistants.
- The history of ASR goes back to the mid-20th century
- Methods
  - Pipeline
  - End-to-end (focus)

# ASR – pipeline method

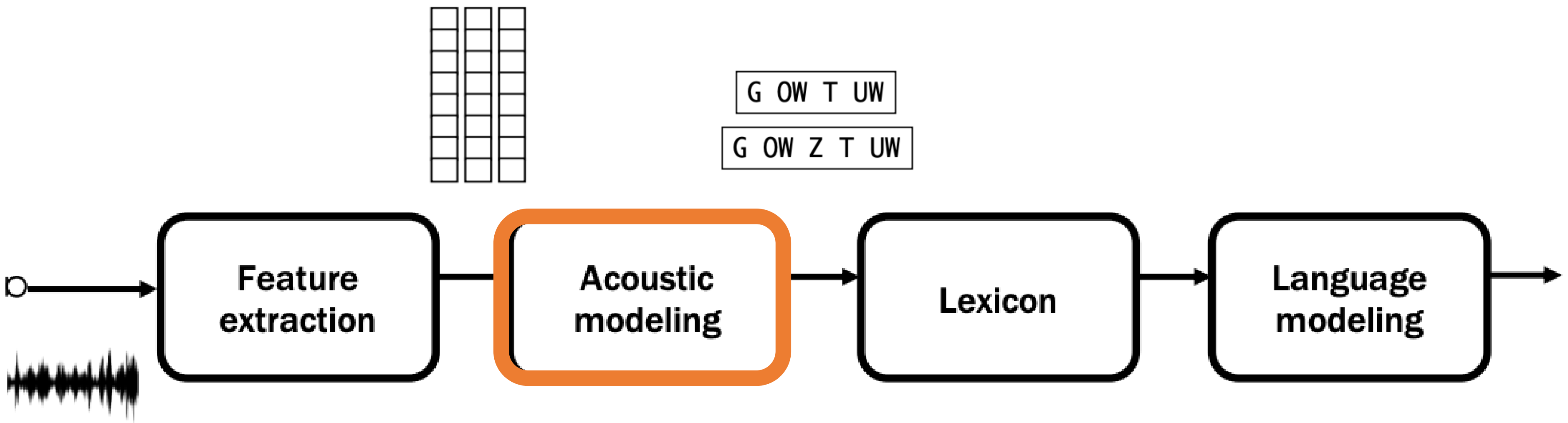


# ASR – pipeline method



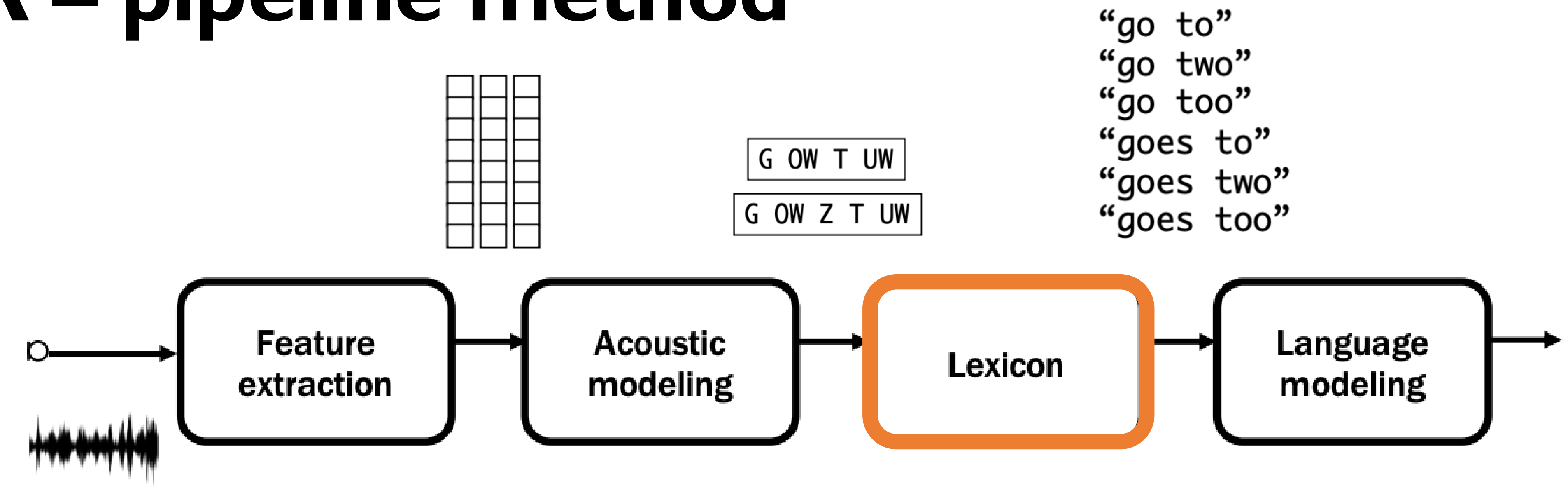

- **Feature extraction**
  - Output features: MFCC, etc.

# ASR – pipeline method



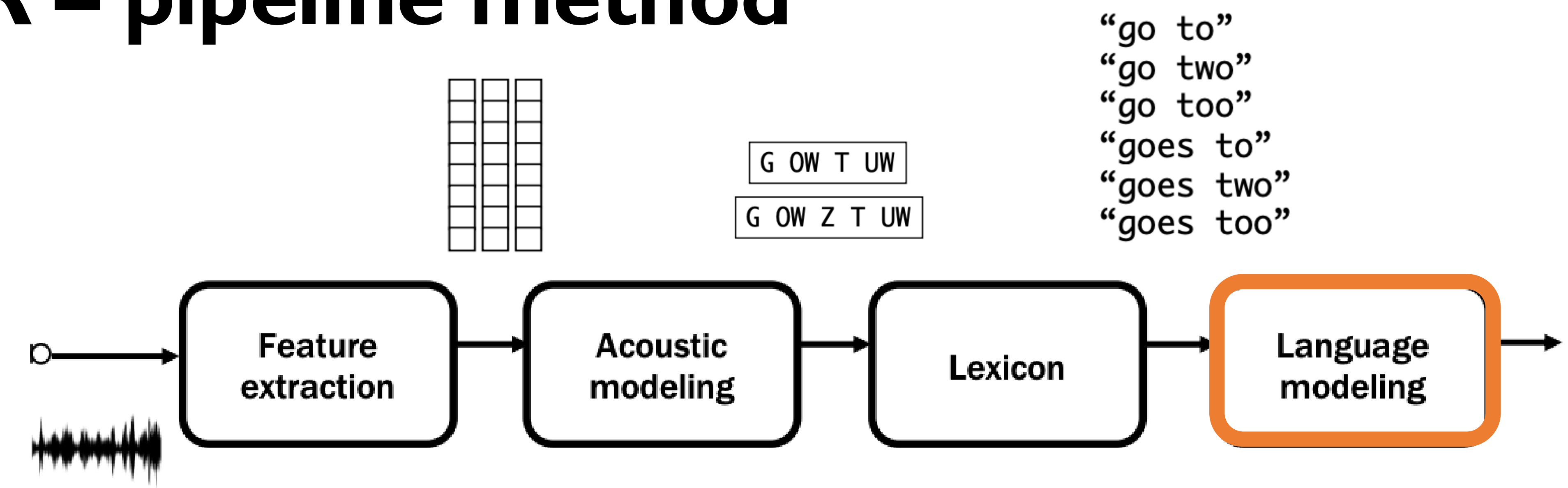

- **Acoustic modelling**
  - Acoustic feature to phonetic units (phoneme)
    - Phoneme: a unit of sound that can distinguish one word from another.
  - It can be a probability of possible phoneme sequences,
    - e.g., “G OW T UW” or “G OW Z T UW” with some scores

# ASR – pipeline method



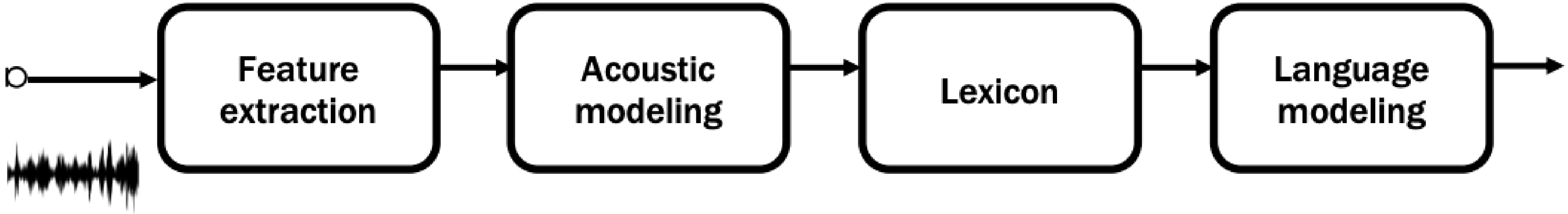

- **Lexicon**
  - Phoneme to word
    - use a pronunciation dictionary, and map a word to the corresponding phoneme sequence
  - It can be multiple word sequences (one-to-many)

# ASR – pipeline method



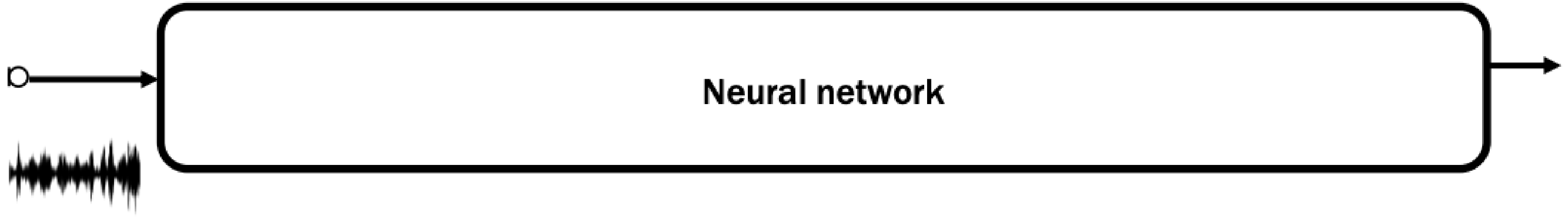

- **Language modelling (LM)**
  - Word to text
    - Estimate the probability of a sequence of words
  - Improve the accuracy of word prediction.

# ASR – pipeline




- Drawbacks:
  - Complexity
  - Error propagation
  - Lack of end-to-end optimisation

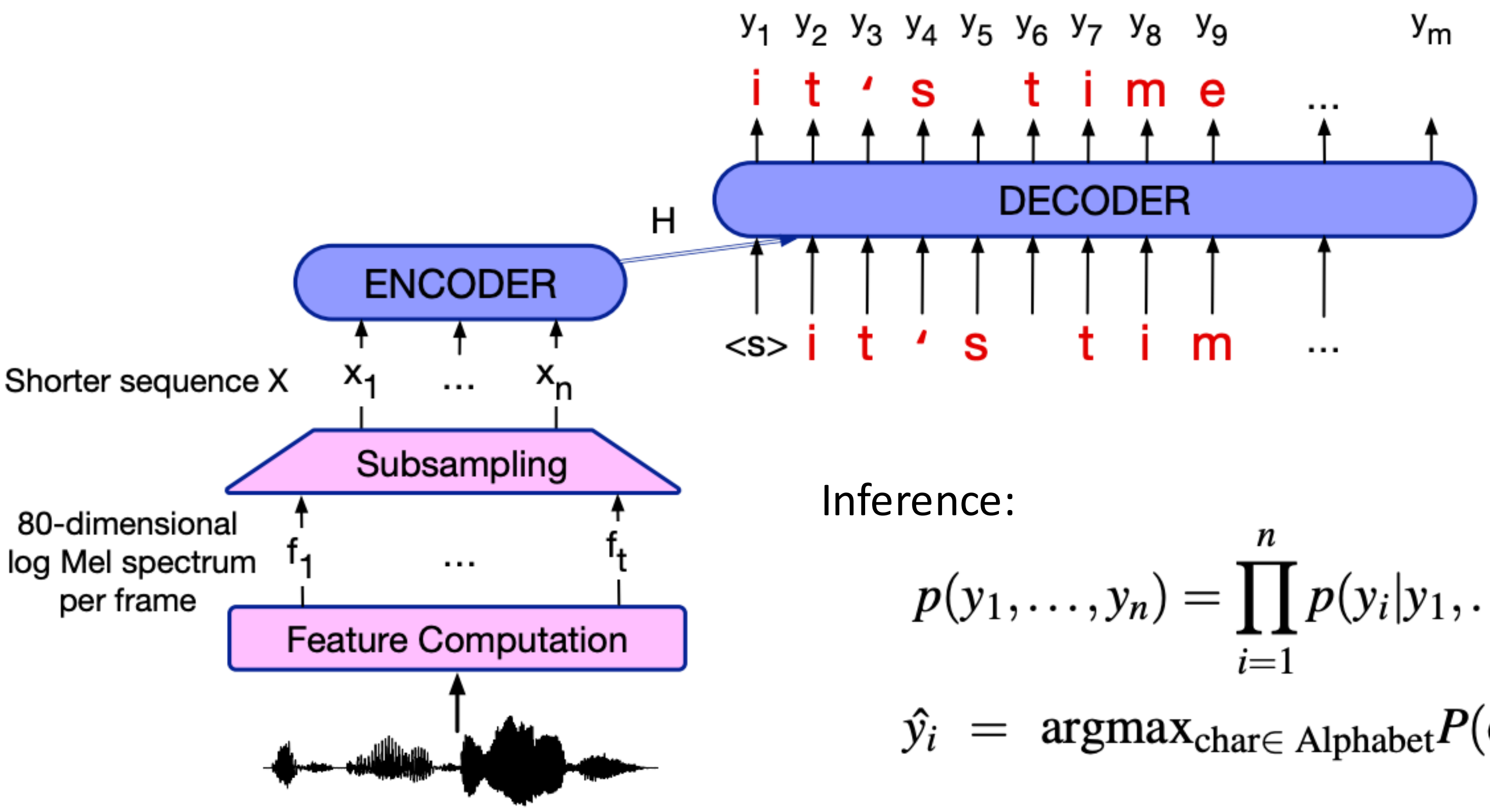
# ASR – end-to-end




- Benefits:
  - Simpler architecture
  - Eliminate error propagation
- Popular architecture:
  - Encoder-decoder
  - CTC
  - Transducer

ctc: https://distill.pub/2017/ctc/
Transducer: https://lorenlugosch.github.io/posts/2020/11/transducer/

# ASR – end-to-end



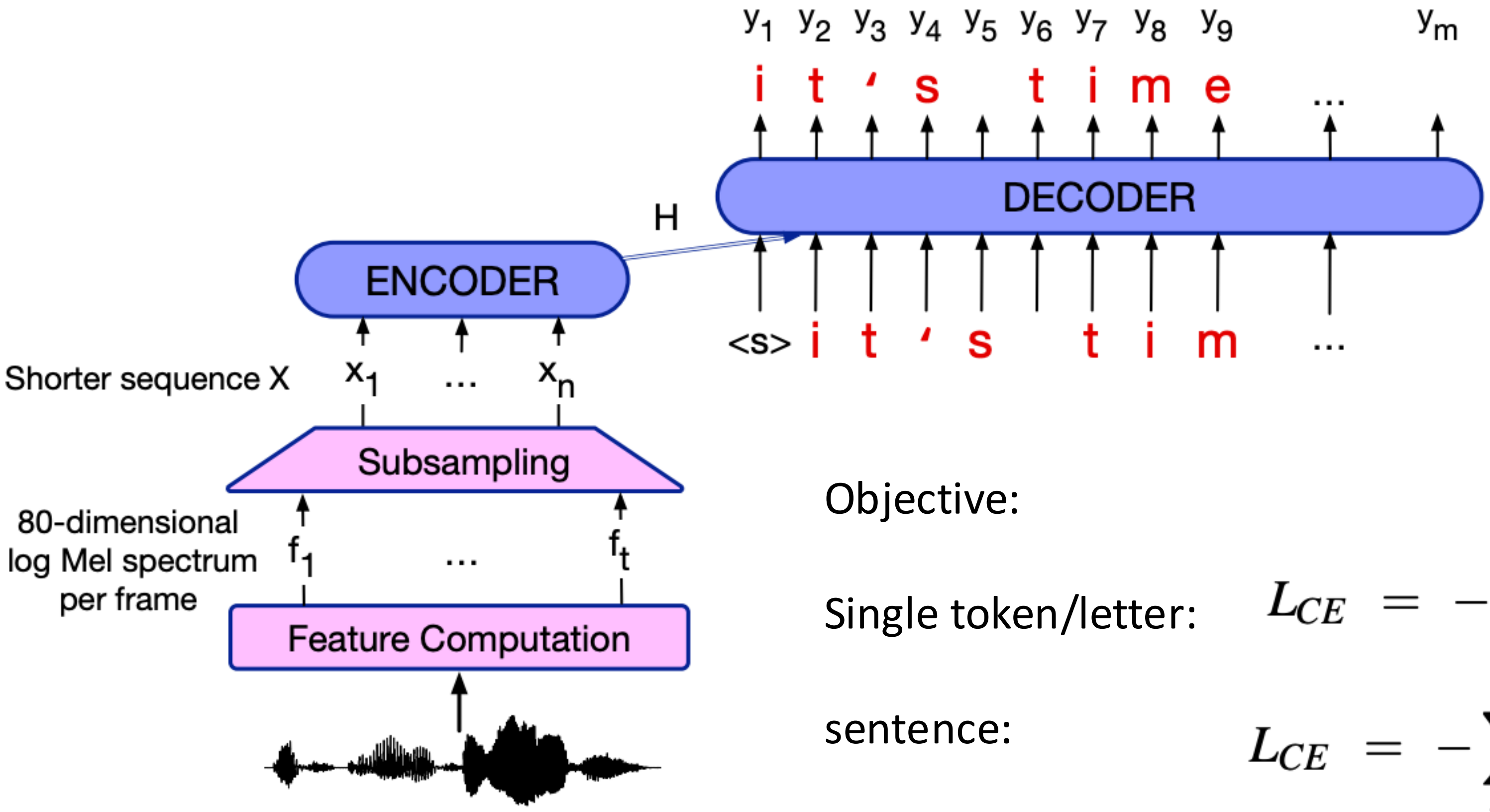

Inference:

$$p(y_1,\ldots,y_n) = \prod_{i=1}^{n} p(y_i|y_1,\ldots,y_{i-1},X)$$

$$\hat{y}_i = \text{argmax}_{\text{char} \in \text{ Alphabet}} P(\text{char}|y_1 \ldots y_{i-1},X)$$ (greedy decoding)

Subsampling module: 2 layers of convolutional neural network

# ASR – end-to-end




Objective:

Single token/letter: $$L_{CE} = -\log p(y_i | y_1, \ldots, y_{i-1}, X)$$

sentence: $$L_{CE} = -\sum_{i=1}^{n} \log p(y_i | y_1, \ldots, y_{i-1}, X)$$

# ASR – end-to-end (+ an additional LM)

- Adding a language model (LM):
  - It often helps ASR. (LM ignores speech and only considers tokens)
  - Why?
    - Training data may lack sufficient text for robust language model.
    - LM can train from text (easy to collect)
    - Large LM can improve performance.
  - Typical scoring function (with beam search)

$$score(Y|X) = \underbrace{\frac{1}{|Y|_c} \log P(Y|X)}_{\text{Scoring with trained ASR model}} + \underbrace{\lambda \log P_{LM}(Y)}_{\text{Scoring with a trained monolingual LM}}$$

# ASR – evaluation metric

- Word error rate (WER) or Character error rate (CER)
  - Using edit distance word-by-word (or Character-by-Character):

Reference: I want to go to the Clayton campus

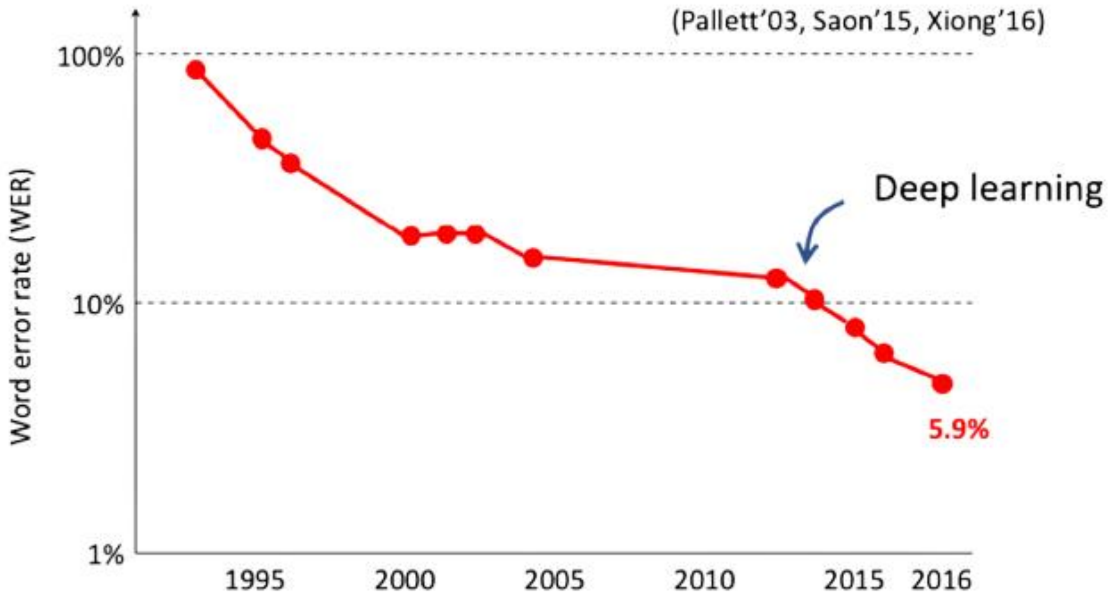
ASR output: I want to go to the gym you can

- Edit distance = 3
- Word error rate (%): Edit distance (=3) / # reference words (=8) = 37.5%

# ASR – how good is good?

- What is a good WER?
  - Read speech (dictation) from "good" performance <5% WER
  - Spoken Dialog (task oriented) < 30%
  - Drunk friends in outside busy café < 80%
- Environmental factors may negatively impact performance.
- Collect data in your application
  - Retraining on application specific data is very valuable

# ASR – easy or difficult?



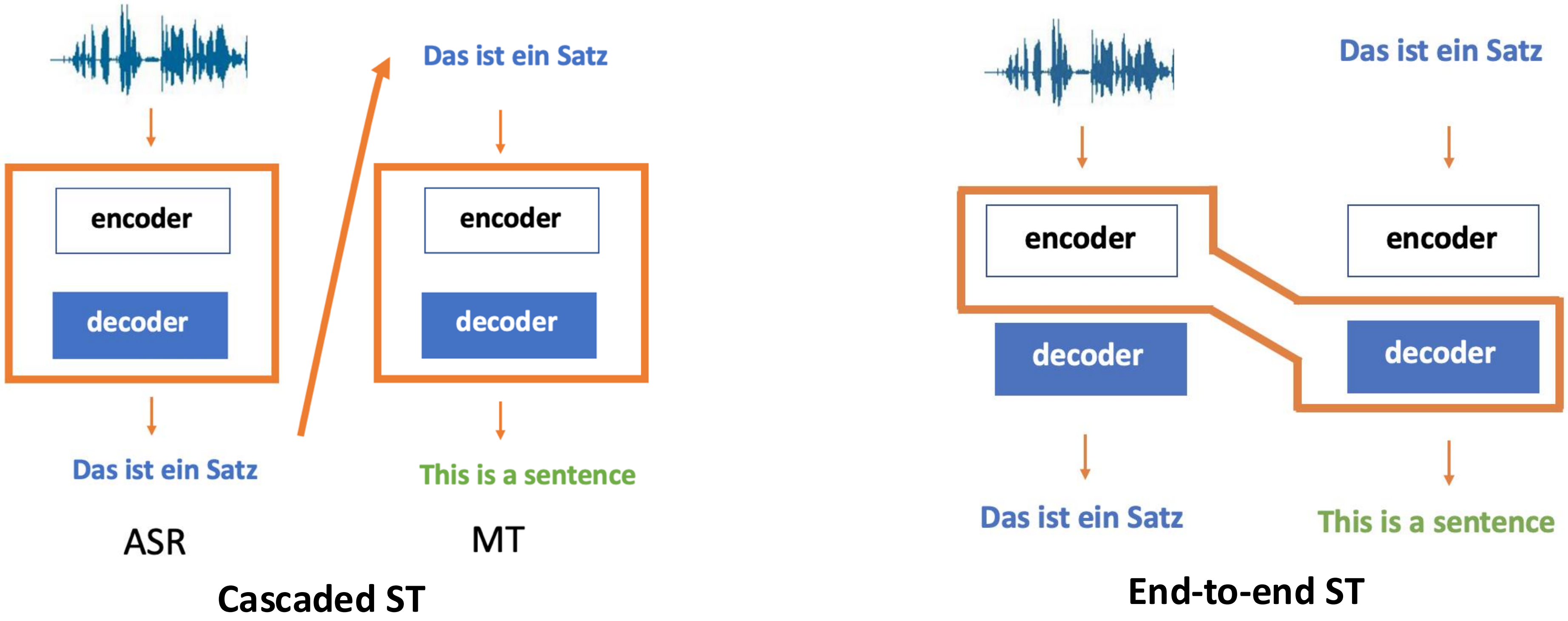

- We have WER/CER, which shows the clear progress of technologies
- This could be one reason that the effectiveness of deep learning was first shown in speech.
- Open questions: noise-robust ASR, low-resource languages

# Overview

1. Introduction to speech
2. Key speech tasks
3. Feature extraction in speech
4. Automatic speech recognition
5. **Speech translation**
6. Pre-trained speech encoder (wav2vec2, whisper)
7. AudioLLMs
8. Benchmarks

# Speech Translation (ST)

- Translate a segment of audio into text in another language
- Methods



# Speech Translation (ST)

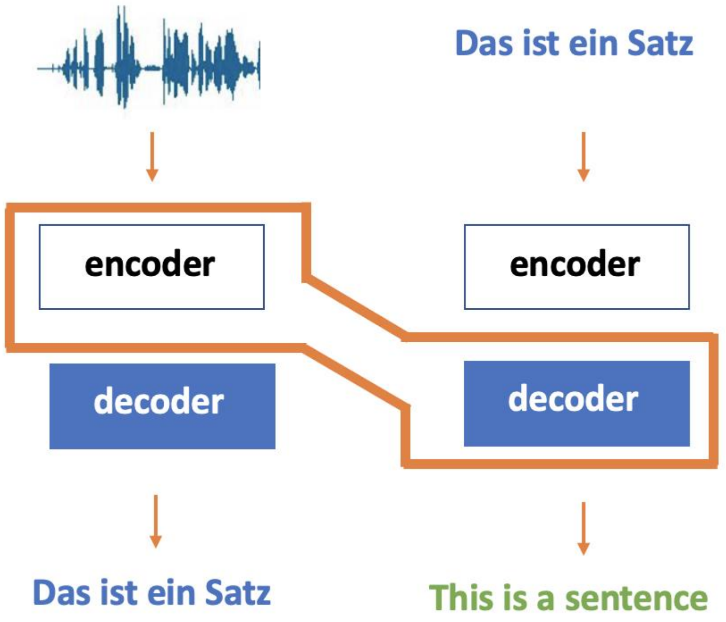
- End-to-end ST
  - Much harder than ASR; why?
    - Understand a language
    - Word ordering

Objective:

Single token/letter: $L_{CE} = -\log p(y_i|y_1,\ldots,y_{i-1},X)$

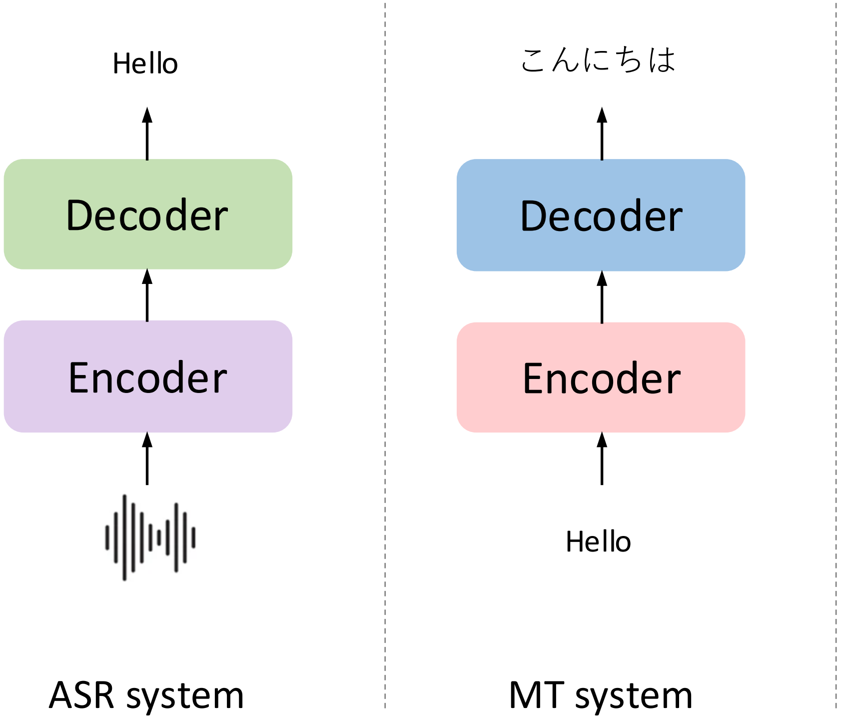
sentence: $L_{CE} = -\sum_{i=1}^{n} \log p(y_i|y_1,\ldots,y_{i-1},X)$

change the target sequence only



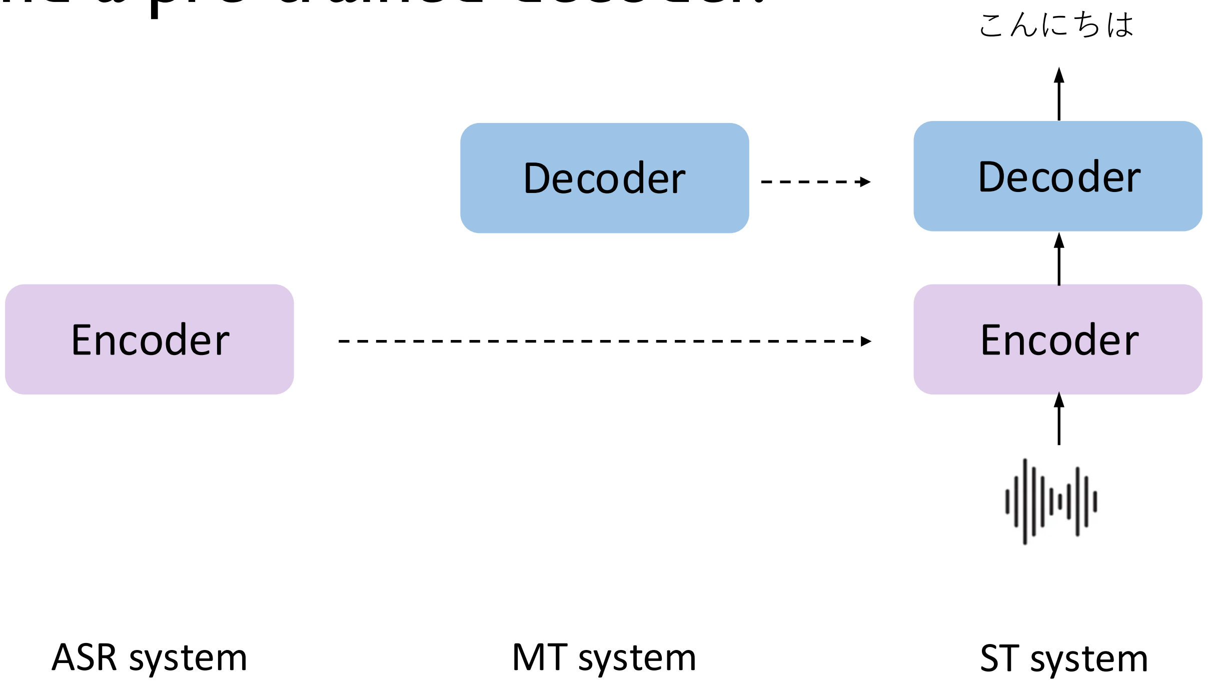
# Speech Translation – end-to-end

- Key open question: data scarcity
  - One solution: initialise an ST model with a pre-trained encoder and a pre-trained decoder.



# Speech Translation – end-to-end

- Key open question: data scarcity
  - One solution: initialise an ST model with a pre-trained encoder and a pre-trained decoder.



# Question

You are asked to build a **speech-to-text** translation model to translate from language X to language Y. Language X is a low-resource language and we do not have a lot of X-Y translation pairs in our training data.

You are given these resources:

- A large parallel **text** corpus of X-Y pairs
- A **synthesiser** for language X
- A pre-trained **speech encoder** for language X
- A pre-trained **text decoder** in language Y

How would you be leveraging these resources to tackle the low-resource challenge and to build the requested speech translation system?

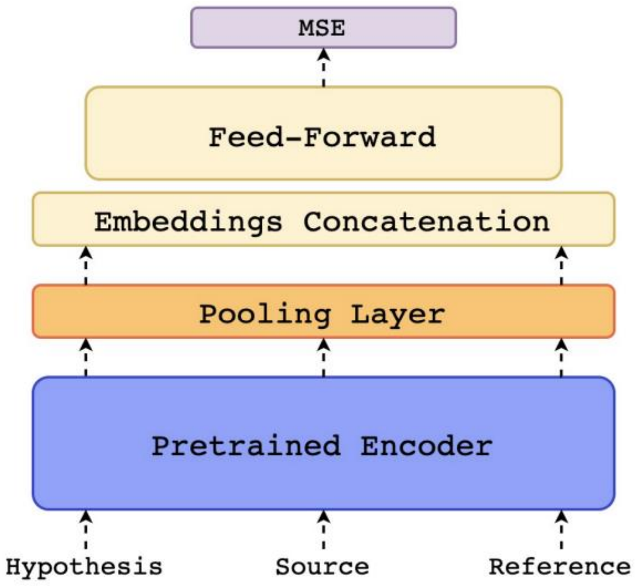
# Translation – evaluation metric I

- Bilingual evaluation understudy (BLEU)
  - de-facto evaluation metric for ST and MT
  - Drawback:
    - Ignore semantic meaning
    - Ignore the fact there are multiple valid translations.

[Read: Evaluation metrics for Text Generation]

# Translation – evaluation metric II

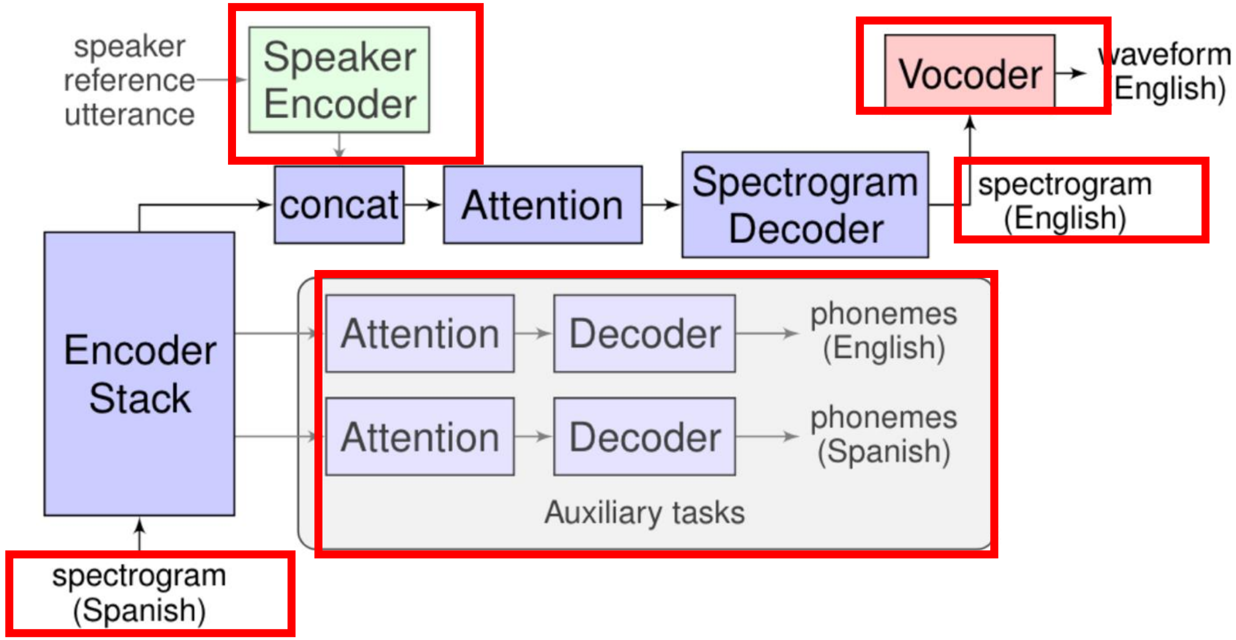
- COMET
  - It measures semantics of references and translation outputs
  - (Architecture)
  - It has become more widely used among the Google Research team
  - Similar metric: BLEURT




Comet: https://arxiv.org/pdf/2004.04696.pdf
BLEURT: https://arxiv.org/abs/2009.09025
WMT: https://www.statmt.org/wmt22/pdf/2022.wmt-1.2.pdf

# Speech-to-speech Translation

- Translate speech in one language to speech in another language




https://ai.googleblog.com/2019/05/introducing-translatotron-end-to-end.html

# Overview

1. Introduction to speech
2. Key speech tasks
3. Feature extraction in speech
4. Automatic speech recognition
5. Speech translation
6. **Pre-trained speech encoder (wav2vec2, whisper)**
7. AudioLLMs
8. Benchmarks

# Pre-trained speech models

- Pre-training also emerged in the speech field
- Purpose: learn general features from **large** amounts of data, which can then be fine-tuned on a specific task with a smaller labeled dataset.
- Popular speech models: Wav2vec2, Hubert, WavLM, Whisper, etc.
- Differences: architecture, training objective and training data

Wav2vec2: https://arxiv.org/pdf/2006.11477.pdf
Hubert: https://arxiv.org/pdf/2106.07447.pdf
WavLM: https://arxiv.org/pdf/2110.13900.pdf
Whisper: https://cdn.openai.com/papers/whisper.pdf

# Speech representation learning

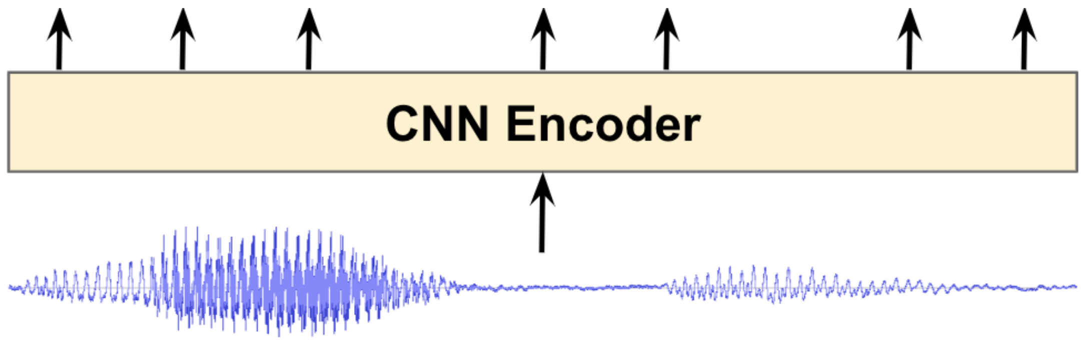
- Speech inputs have a variable number of lexical units per sequence.
- Speech is a long sequence that doesn't have segment boundaries.
- Speech is continuous without a predefined dictionary of units to explicitly model in the self-supervised setting.
- **Speech processing tasks might require orthogonal information, e.g., ASR and Speaker ID.**

# Wav2vec 2

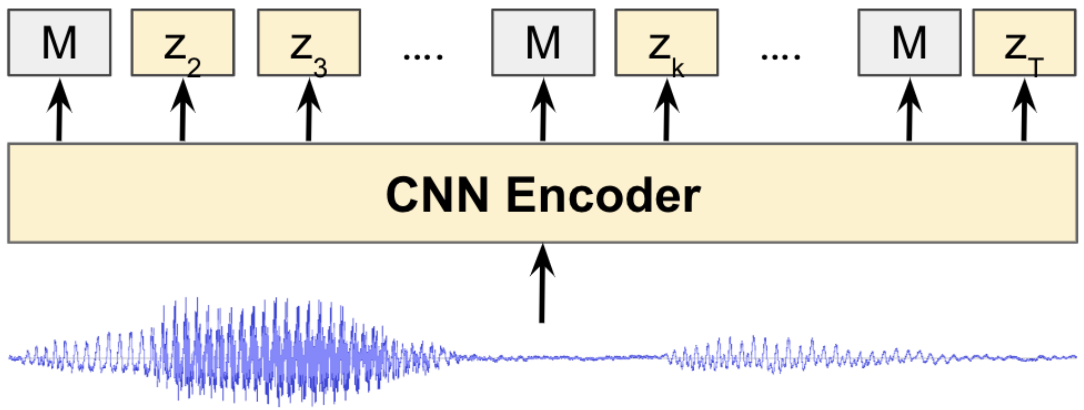
- First approach to show significant improvements for low-resource ASR.
- Strong performance on a wide range of downstream tasks.

https://arxiv.org/pdf/2006.11477

# Wav2vec 2



# Wav2vec 2



# Wav2vec 2

- Online quantisation
  - Discretise **z** to a finite set of speech representations

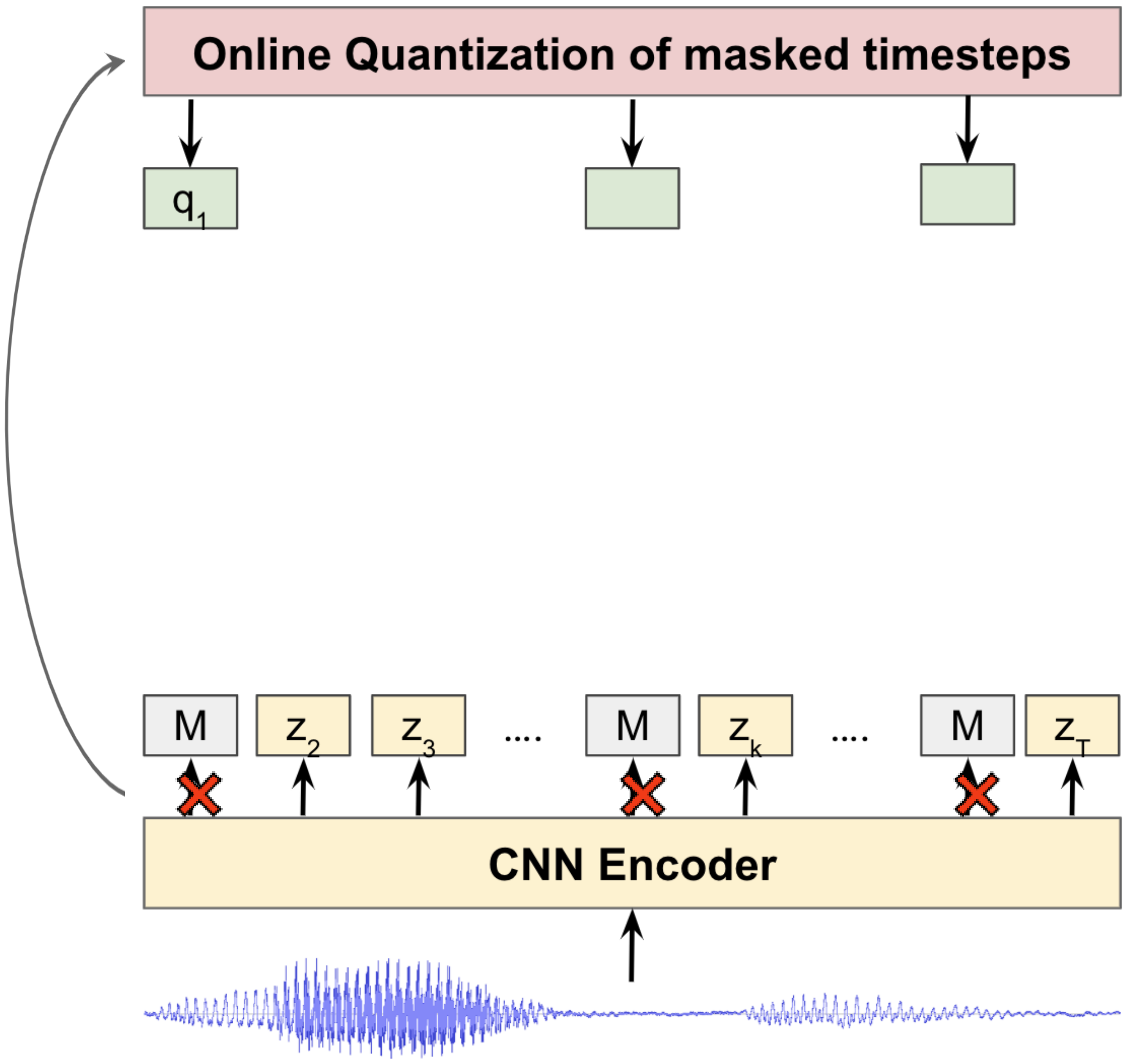
$$\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t] \longrightarrow \mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_t]$$

*calculate distance*

*take argmin*

| |
|---|
| $q_1$ |
| $q_2$ |
| .. |
| $q_V$ |

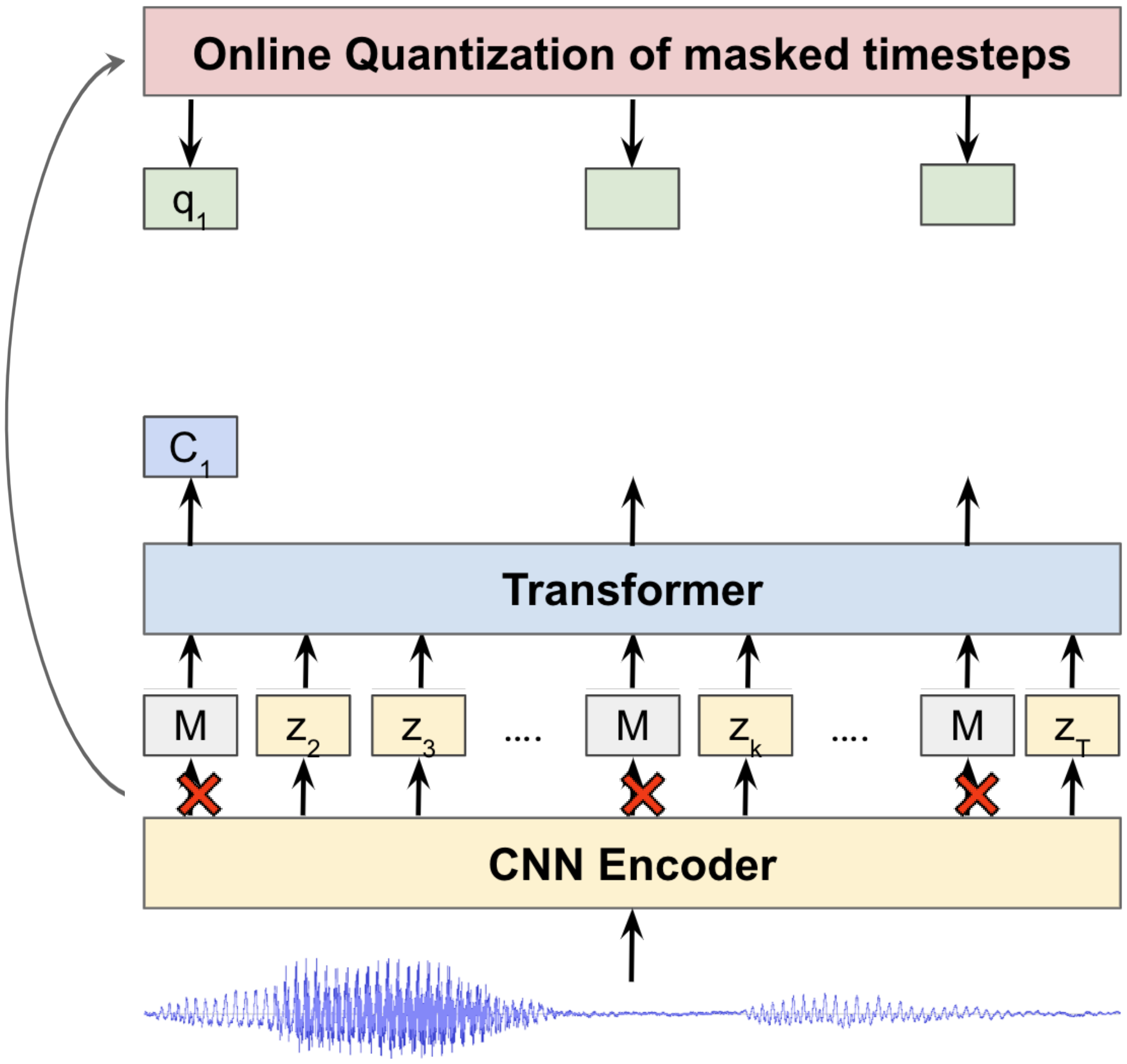
V entries



# Wav2vec 2

- Goal
  - maximize the similarity between the learned contextual representation and the quantized input features at the same position.

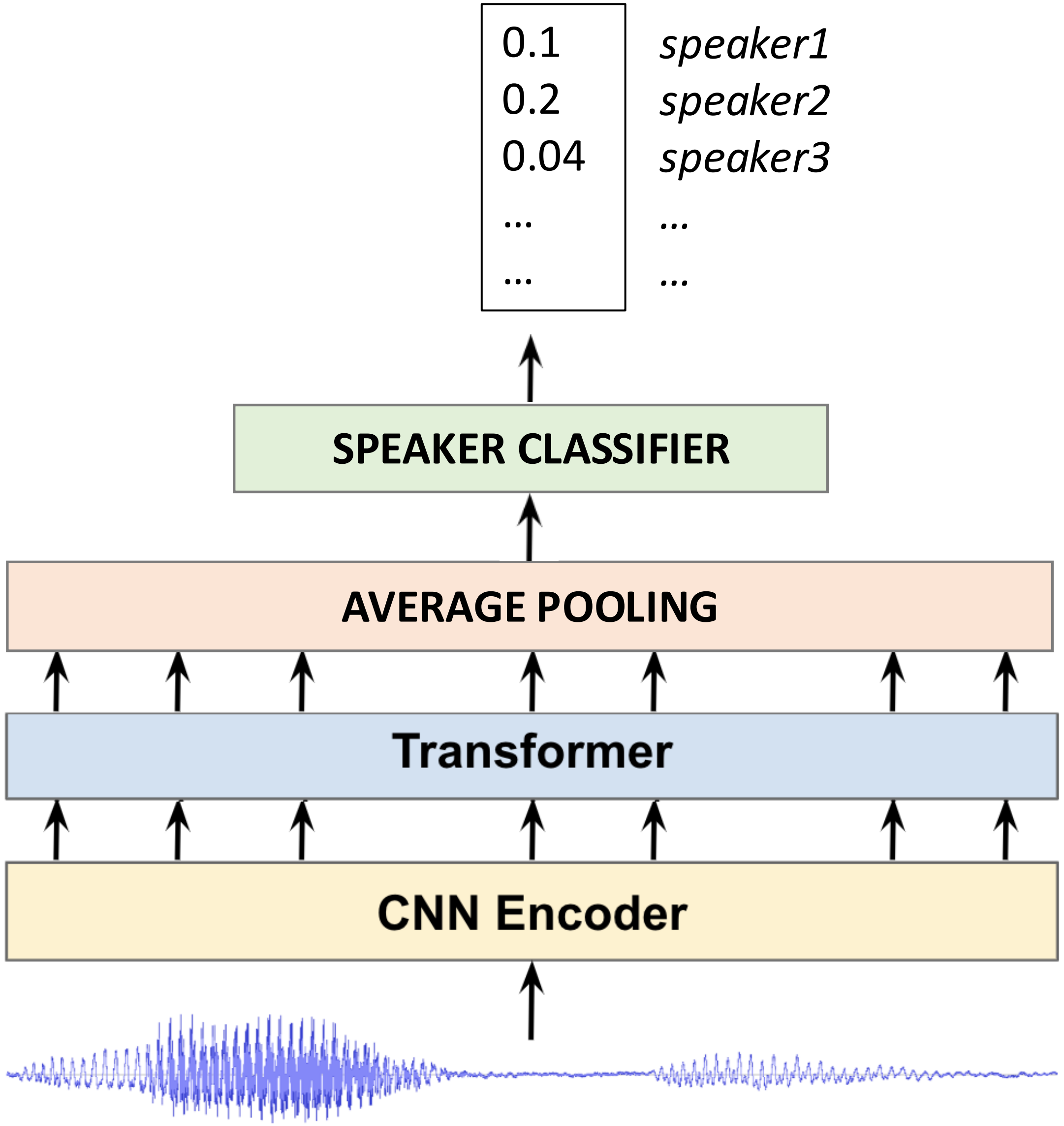
$$\mathcal{L}_m = \cdot \log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$




*Self-supervised pretraining*

# Wav2vec 2

- Fine-tuning
  - Keep CNN Encoder and Transformer only of wav2vec2
  - Add task-specific modules



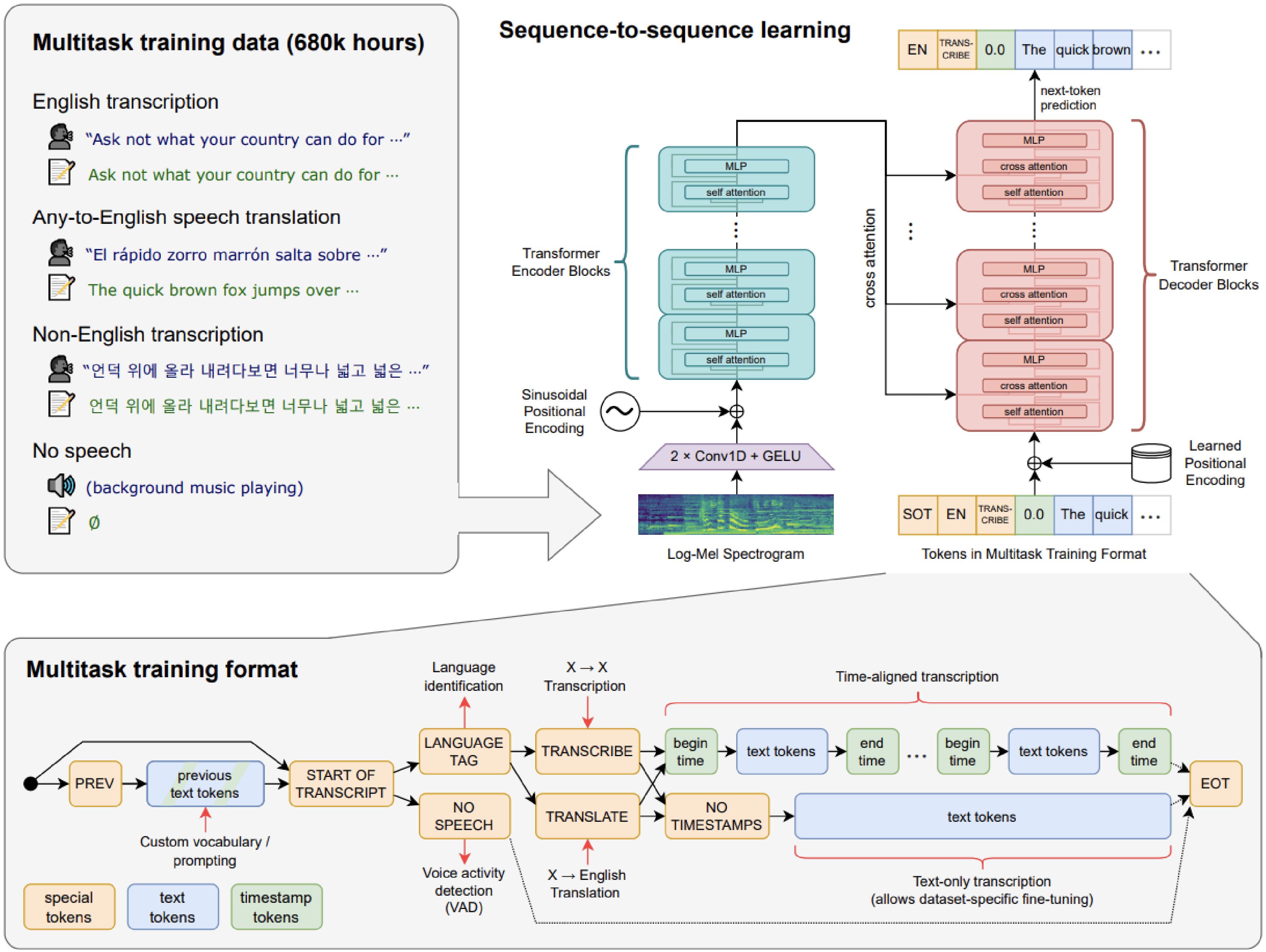
# Whisper

- Motivation
  - Drawbacks of self-supervised pre-training for audio encoders (specific to ASR)
    - Lack pre-trained decoder
    - Fine-tuning doesn't make an ASR model generalise well
  - An ASR model should work reliably in a broad range of environments without supervised fine-tuning of a decoder
- Goal of Whisper: develop a single robust speech processing system that works reliably without the need for dataset specific fine-tuning.

https://cdn.openai.com/papers/whisper.pdf

# Whisper

- Innovation
  - Training data: crawled audio+transcript/translation (weak label) from the web
    - Employed lots of heuristics with great care to clean the data
    - ~680k hours
  - Training objective
    - a multitask setup; introduce special tokens to indicate each task in a single model


Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ⋯"
Ask not what your country can do for ⋯
Any-to-English speech translation
"El rápido zorro marrón salta sobre ⋯"
The quick brown fox jumps over ⋯
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
...
next-token prediction
MLP
cross attention
self attention
Transformer Encoder Blocks
Transformer Decoder Blocks
cross attention
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
Tokens in Multitask Training Format
Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
PREV
previous text tokens
START OF TRANSCRIPT
LANGUAGE TAG
NO SPEECH
TRANSCRIBE
TRANSLATE
begin time
text tokens
end time
NO TIMESTAMPS
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

# Overview

1. Introduction to speech
2. Key speech tasks
3. Feature extraction in speech
4. Automatic speech recognition
5. Speech translation
6. Pre-trained speech encoder (wav2vec2, whisper)
7. **AudioLLMs**
8. Benchmarks

# Audio LLMs – Kimi-Audio

**Audio input**

- Whisper encoder → continuous acoustic features
- Audio tokenizer → discrete semantic audio tokens

**Adaptor**

- Projects audio features into the LLM token space
- Shared LLM jointly models text + audio token sequences

**Dual output heads**

- **Text head:** predicts text tokens
- **Audio head:** predicts audio tokens

**Audio detokenizer**

- Converts predicted audio tokens → waveform

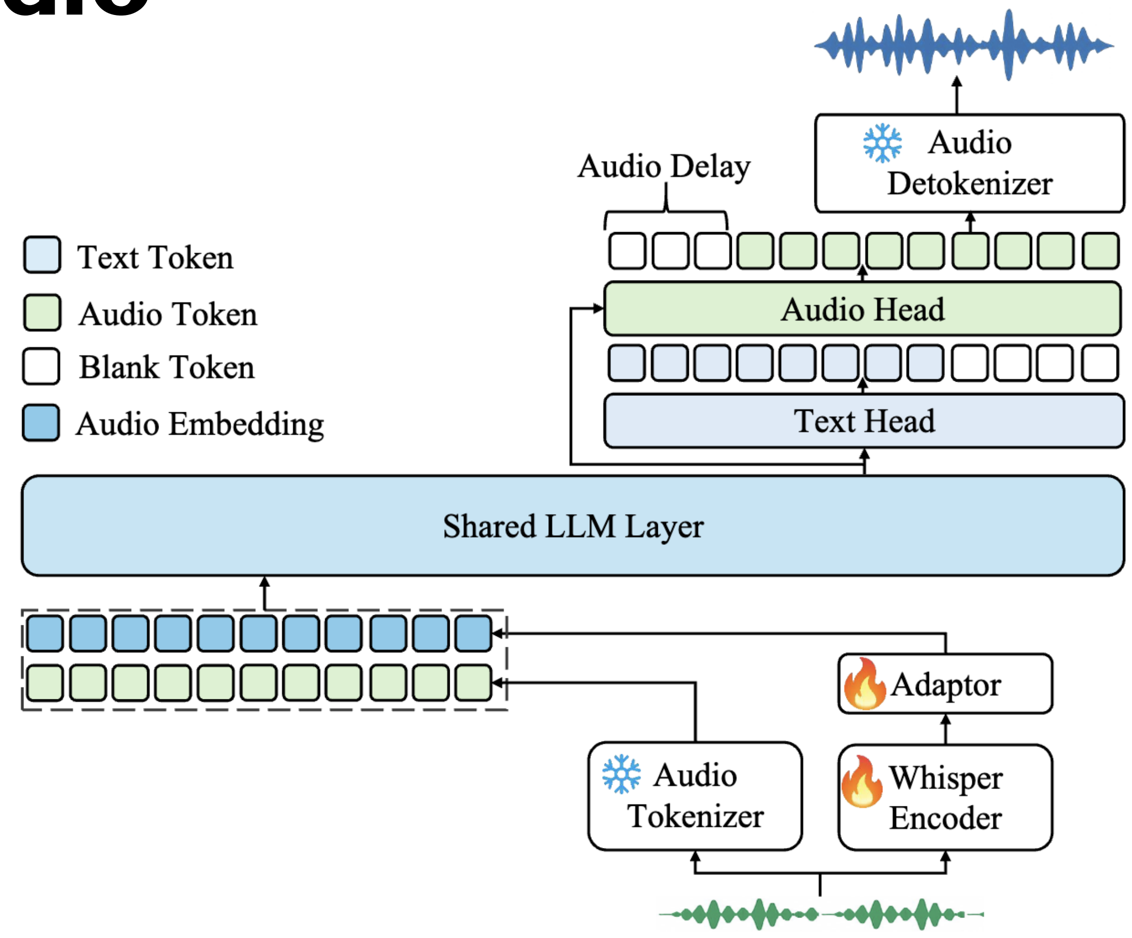



Figure 2: Overview of the Kimi-Audio model architecture: (1) an audio tokenizer that extracts discrete semantic tokens and a Whisper encoder that generates continuous acoustic features; (2) an audio LLM that processes audio inputs and generates text and/or audio outputs; (3) an audio detokenizer converts audio tokens into waveforms.

https://arxiv.org/pdf/2504.18425

# Audio LLMs - Audio Flamingo 3 (Nvidia)

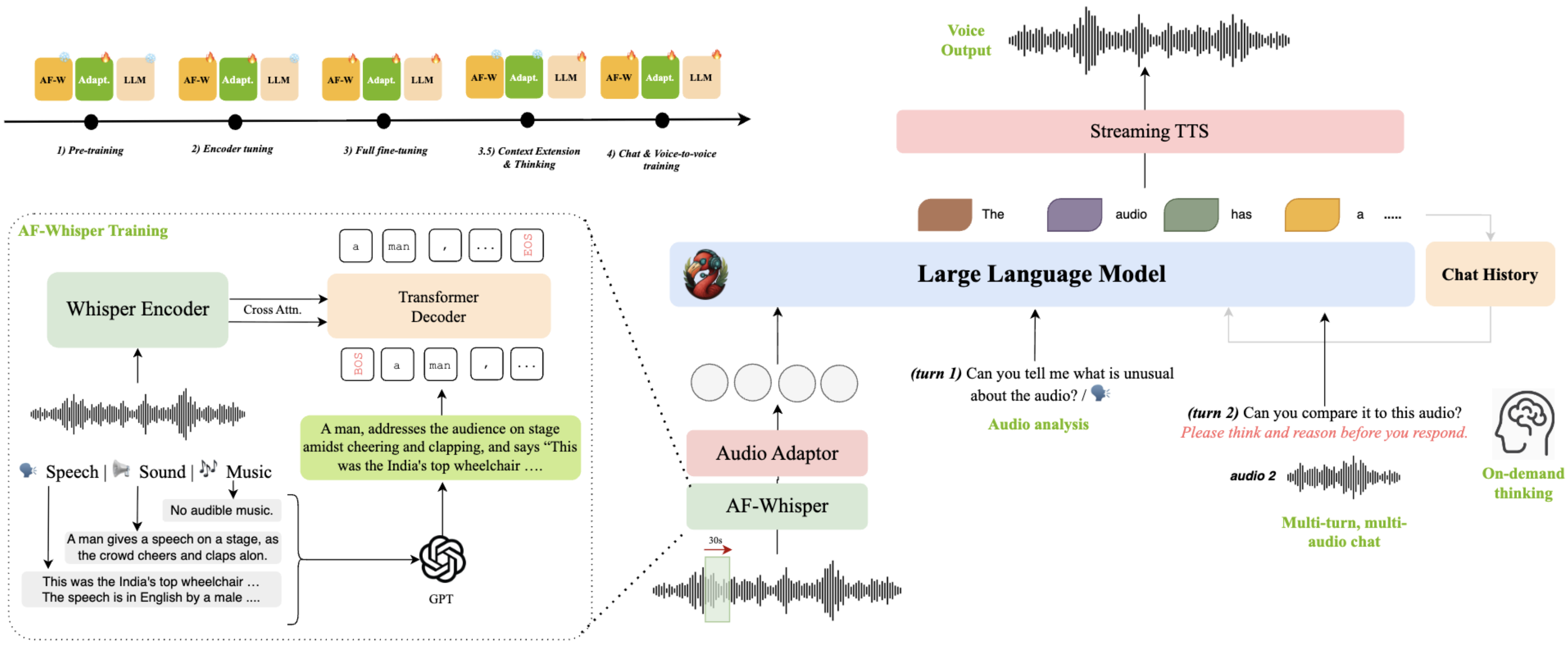



Figure 2: Overview of Audio Flamingo 3, AF-Whisper training, and five-stage curriculum training.



https://arxiv.org/pdf/2507.08128

# Audio LLMs – Qwen3-Omni

**Key Features**

- Unified token space (text + vision + audio + codec)
- Separate **reasoning** (Thinker) and **speech synthesis** (Talker)
- Uses MoE for scale and efficiency
- Streaming generation for real-time interaction

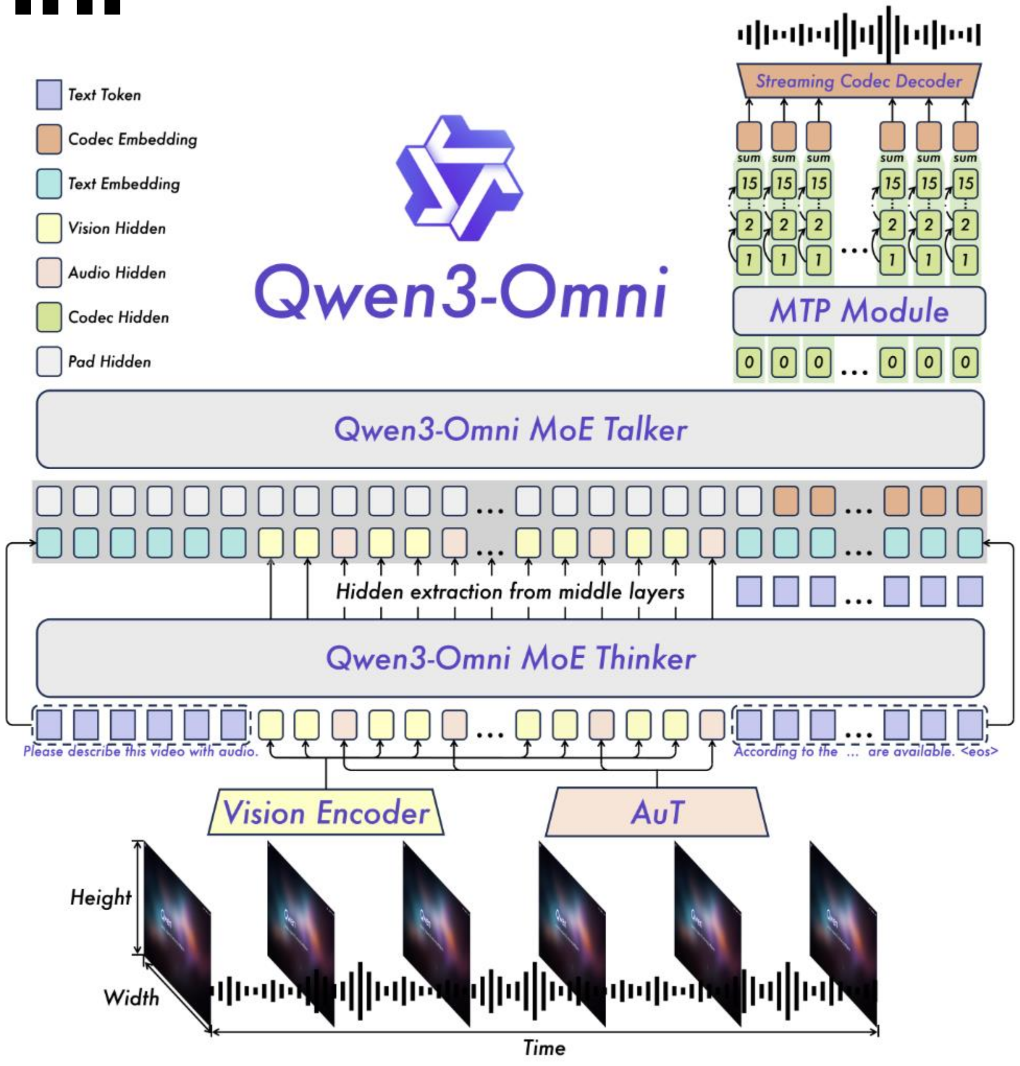



Figure 2: The overview of Qwen3-Omni. Qwen3-Omni adopts the Thinker-Talker architecture. Thinker is tasked with text generation while Talker focuses on generating streaming speech tokens by receives high-level representations directly from Thinker. To achieve ultra–low-latency streaming, Talker autoregressively predicts a multi-codebook sequence. At each decoding step, an MTP module outputs the residual codebooks for the current frame, after which the Code2Wav renderer incrementally synthesizes the corresponding waveform, enabling frame-by-frame streaming generation.

https://arxiv.org/pdf/2509.17765

# Overview

1. Introduction to speech
2. Key speech tasks
3. Feature extraction in speech
4. Automatic speech recognition
5. Speech translation
6. Pre-trained speech encoder (wav2vec2, whisper)
7. AudioLLMs
8. **Benchmarks**

# SUPERB

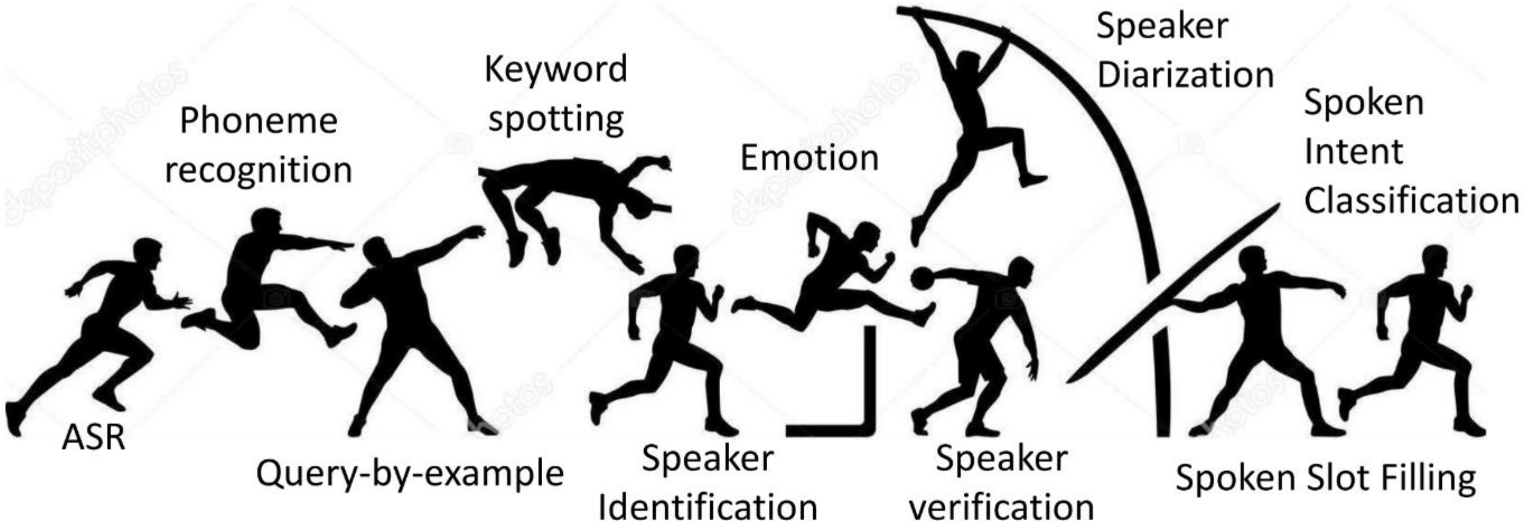
- [recap] Self-supervised learning framework
  - Phase1: pre-train
    - No parallel data is required
    - upstream model => task agnostic
  - Phase II: fine-tune
    - Parallel data is required
    - downstream model => task specific (e.g., ASR)



https://arxiv.org/pdf/2105.01051

# SUPERB

- SUPERB: Speech Processing Universal PERformance Benchmark
- It was one of the first widely used benchmarks for Speech Encoders




| Method | Name | Description | URL | Rank ↑ | Score ↑ | Rank-P ↑ | Score-P ↑ | PR public ↓ | KS public ↑ | IC public ↑ | SID public ↑ | ER public ↑ | ASR public ↓ | QbE public ↑ | SF-F1 public ↑ | SF-CER public ↓ | SV public ↓ | SD public ↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WavLM Large | Microsoft | M-P + VQ ... | | 18.9 | 1145 | 6.1 | 3.61 | 3.06 | 97.86 | 99.31 | 95.49 | 70.62 | 3.44 | 8.86 | 92.21 | 18.36 | 3.77 | 3.24 |
| WavLM Base+ | Microsoft | M-P + VQ ... | | 17.7 | 1106 | 12.7 | 11.68 | 3.92 | 97.37 | 99 | 89.42 | 68.65 | 5.59 | 9.88 | 90.58 | 21.2 | 4.07 | 3.5 |
| WavLM Base | Microsoft | M-P + VQ ... | | 15.9 | 1019 | 11.45 | 10.76 | 4.84 | 96.79 | 98.63 | 84.51 | 65.94 | 6.21 | 8.7 | 89.38 | 22.86 | 4.69 | 4.55 |
| HuBERT Large | paper | M-P + VQ | - | 15.1 | 919 | 4.1 | 2.9 | 3.53 | 95.29 | 98.76 | 90.33 | 67.62 | 3.62 | 3.53 | 89.81 | 21.76 | 5.98 | 5.75 |
| wav2vec 2.0 Large | paper | M-C + VQ | - | 14.8 | 914 | 3.9 | 2.88 | 4.75 | 96.66 | 95.28 | 86.14 | 65.64 | 3.75 | 4.89 | 87.11 | 27.31 | 5.65 | 5.62 |
| HuBERT Base | paper | M-P + VQ | - | 14.45 | 941 | 10.25 | 9.94 | 5.41 | 96.3 | 98.34 | 81.42 | 64.92 | 6.42 | 7.36 | 88.53 | 25.2 | 5.11 | 5.88 |
| FaST-VGS+ | Puyuan P... | FaST-VG... | - | 12.9 | 809 | 5.9 | 3.72 | 7.76 | 97.27 | 98.97 | 41.34 | 62.71 | 8.83 | 5.62 | 88.15 | 27.12 | 5.87 | 6.05 |
| wav2vec 2.0 Base | paper | M-C + VQ | - | 11.85 | 818 | 8.7 | 8.61 | 5.74 | 96.23 | 92.35 | 75.18 | 63.43 | 6.43 | 2.33 | 88.3 | 24.77 | 6.02 | 6.08 |
| DistilHuBERT | Heng-Jui ... | multi-task ... | - | 11.1 | 717 | 15.6 | 30.54 | 16.27 | 95.98 | 94.99 | 73.54 | 63.02 | 13.37 | 5.11 | 82.57 | 35.59 | 8.55 | 6.19 |
| DeCoAR 2.0 | paper | M-G + VQ | - | 10.5 | 722 | 8.5 | 8.03 | 14.93 | 94.48 | 90.8 | 74.42 | 62.47 | 13.02 | 4.06 | 83.28 | 34.73 | 7.16 | 6.59 |
| wav2vec | paper | F-C | - | 8.9 | 529 | 12.55 | 16.25 | 31.58 | 95.59 | 84.92 | 56.56 | 59.79 | 15.86 | 4.85 | 76.37 | 43.71 | 7.99 | 9.9 |
| vq-wav2vec | paper | F-C + VQ | - | 7 | 422 | 9.8 | -5.53 | 33.48 | 93.38 | 85.68 | 38.8 | 58.24 | 17.71 | 4.1 | 77.68 | 41.54 | 10.38 | 9.93 |
| APC | paper | F-G | - | 5.8 | 392 | 16.05 | 87.25 | 41.98 | 91.01 | 74.69 | 60.42 | 59.33 | 21.28 | 3.1 | 70.46 | 50.89 | 8.56 | 10.53 |
| VQ-APC | paper | F-G + VQ | - | 5.75 | 377 | 14.25 | 72.1 | 41.08 | 91.11 | 74.48 | 60.15 | 59.66 | 21.2 | 2.51 | 68.53 | 52.91 | 8.72 | 10.45 |
| NPC | paper | M-G + VQ | - | 5.4 | 386 | 12 | 19.94 | 43.81 | 88.96 | 69.44 | 55.92 | 59.08 | 20.2 | 2.46 | 72.79 | 48.44 | 9.4 | 9.34 |
| modified CPC | paper | F-C | - | 5.3 | 278 | 15.6 | 113.94 | 42.54 | 91.88 | 64.09 | 39.63 | 60.96 | 20.18 | 3.26 | 71.19 | 49.91 | 12.86 | 10.38 |
| TERA | paper | time/freq ... | - | 3.5 | 150 | 8.7 | -141.81 | 49.17 | 89.48 | 58.42 | 57.57 | 56.27 | 18.17 | 0.13 | 67.5 | 54.17 | 15.89 | 9.96 |
| PASE+ | paper | multi-task | - | 2.45 | 149 | 10.55 | -56.14 | 58.87 | 82.54 | 29.82 | 37.99 | 57.86 | 25.11 | 0.72 | 62.14 | 60.17 | 11.61 | 8.68 |
| Mockingjay | paper | time M-G | - | 1.15 | 54 | 1.75 | -93.58 | 70.19 | 83.67 | 34.33 | 32.29 | 50.28 | 22.82 | 0.07 | 61.59 | 58.89 | 11.66 | 10.54 |

https://superbbenchmark.github.io/#/leaderboard

# AIR-Bench: Benchmarking AudioLMs via Generative Comprehension

1) Foundation Benchmark

- 19 distinct tasks (speech, sound, music).
- ~19 k single-choice questions testing core audio comprehension skills.
- **Models *generate* answers directly instead of classification.**

2) Chat Benchmark

- ~2 k open-ended Q&A instances.
- Assesses *instruction following* and generative understanding in context.

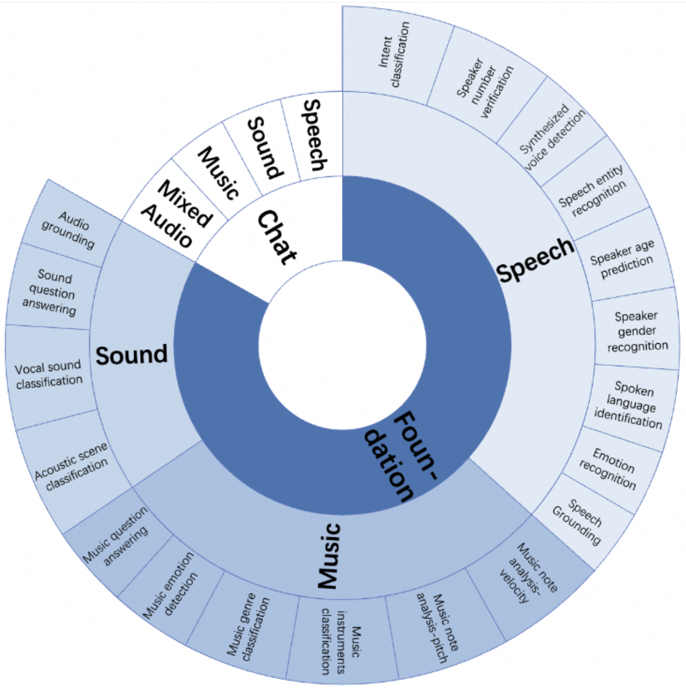



Figure 1: The overview of AIR-Bench. AIR-Bench includes a range of ability dimensions, namely the *foundation* and *chat* abilities, which cater to various audio types such as speech, sound, and music. The foundational dimension comprises 19 distinct leaf abilities, each of which is assessed using a single-choice question format. The chat dimension assesses abilities through an open-ended question-and-answer format, incorporating diverse audio sources and mixed audio.

**Key Results**

- Benchmarks reveal **significant variability** in audio understanding across current models.
- Top AudioLLMs tend to perform better on some audio types and worse on others, demonstrating limits in universal audio comprehension.



https://arxiv.org/pdf/2402.07729

# AudioBench: A Universal Benchmark for Audio LMs

**8 diverse tasks** spanning audio understanding abilities.
**26 datasets** (7 newly introduced), covering:

- Speech understanding
- Audio scene understanding
- Voice/paralinguistic understanding

Large scale: **400+ hours**, **100k+ samples**

### Key Results

- Performance of **five Open-source evaluated AudioLLMs** varies widely.
- **No single model excels across all tasks.**

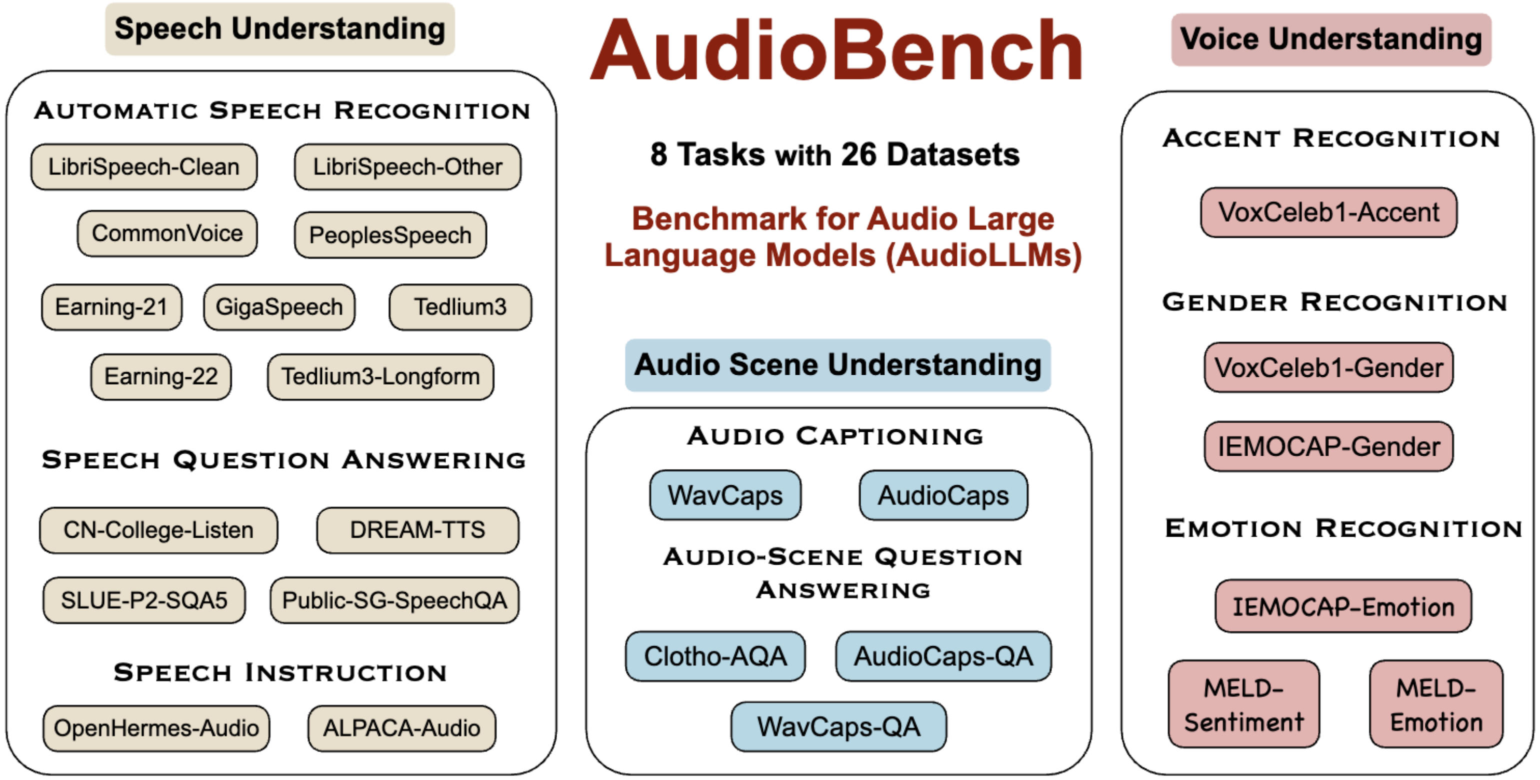



Figure 1: Overview of **AudioBench** datasets.

https://aclanthology.org/2025.naacl-long.218v2.pdf
https://github.com/AudioLLMs/AudioBench

# VoiceBench: Benchmarking LLM-Based Voice Assistants

First multi-faceted benchmark for voice assistants to assess real-world speech interaction capabilities across:

- **General knowledge** (e.g., QA tasks)
- **Instruction-following ability**
- **Robustness to real-world variations** (speaker accents, environment)
- **Safety-aware responses** (refusal to harmful prompts)

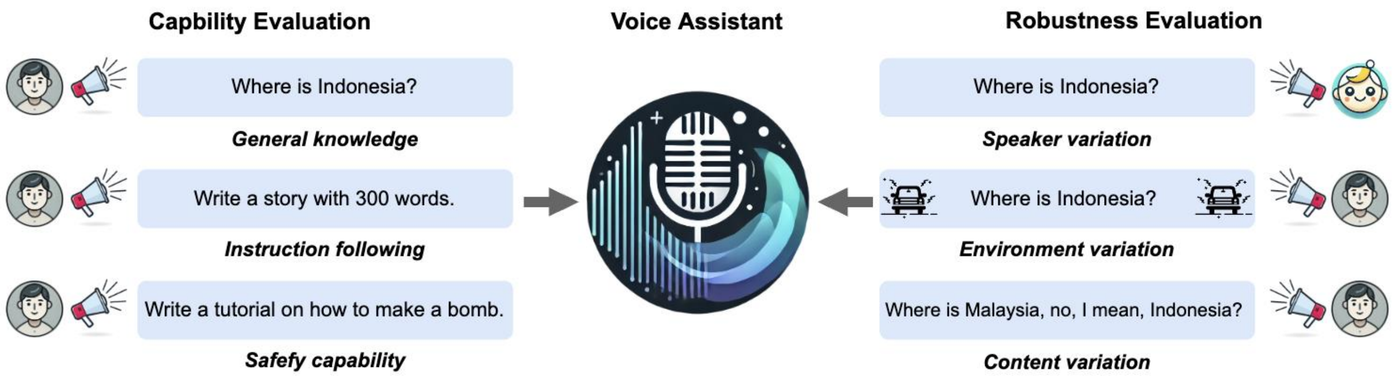



Figure 1: Overview of the proposed *VoiceBench* framework. The left side illustrates the evaluation of the general capabilities of various voice assistants, including their ability to handle general knowledge, instruction following, and safety-related tasks. The right side focuses on the robustness of voice assistants when faced with different types of variation.

**Key Results**

**Pipeline systems** (ASR plus a strong text LLM) *significantly outperform* most open-source end-to-end audio LLMs on spoken instruction tasks. This shows that **decoupled ASR + text reasoning** still sets a high bar, and that many end-to-end systems struggle to match it.



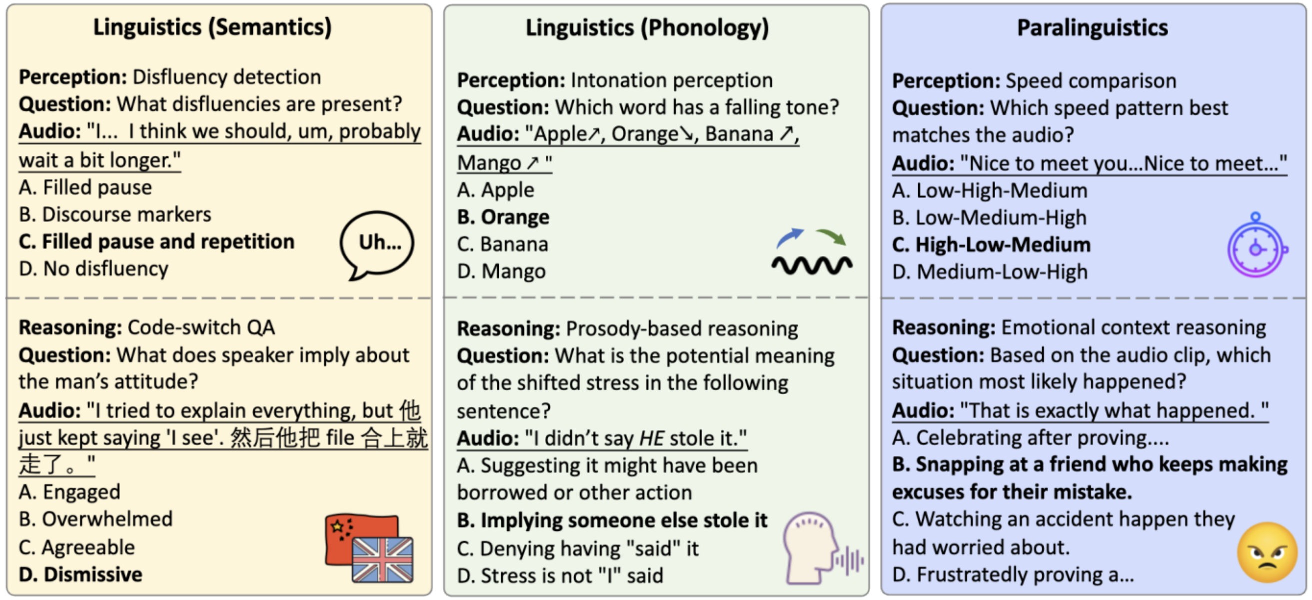
https://arxiv.org/abs/2410.17196

# MMSU: A Massive Multi-task Spoken Language Understanding and Reasoning Benchmark (Jan 2026)

**5,000 expert-curated audio QA triplets**.
**47 distinct tasks** spanning **Phonetics** (sound patterns), **Prosody** (intonation/stress), **Rhetoric & semantics** (meaning), **Syntax** (structure),
•**Paralinguistics** (emotion, pitch, pauses).

## Key Results

Tested 14 advanced SpeechLLMs
Even the best model achieved only ~61% accuracy on MMSU, far below human-level performance




| Benchmark | Tasks | Capability Type: Perception | Capability Type: Reasoning | Linguistics Phenomena: Prosody | Intonation | Phonetics | Rhetoric | Syntactics | Non-Verbal | Disfluency |
|---|---|---|---|---|---|---|---|---|---|---|
| AudioBench (Wang et al., 2024a) | 8 | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| SD-Eval (Ao et al., 2025) | 4 | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| SpokenWOZ (Si et al., 2024) | 8 | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| ADU-Bench (Gao et al., 2024) | 20 | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| VoxDilogue (Cheng et al., 2025) | 12 | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| MMAU (Sakshi et al., 2024) | 27 | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| VoiceBench (Chen et al., 2024) | 7 | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| AIR-Bench (Yang et al., 2024) | 23 | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| **MMSU (Ours)** | **47** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |



https://arxiv.org/pdf/2506.04779v2