NUS National University of Singapore | ISS

Conversational UIs

Spoken Language Processing

Dr. Gary Leung
Email: cc.leung@nus.edu.sg

Agenda

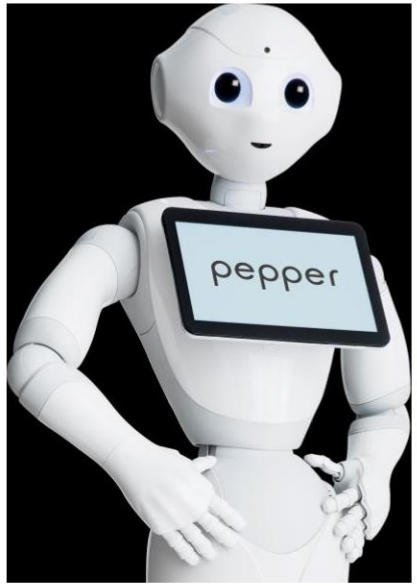
- Day 3
  - 1: Speech processing basics
  - 2: Speech recognition (Speech-to-text)
  - 3: Speaker diarization
- Day 4
  - 4: Speech synthesis (Text-to-speech)
  - 5: Voice conversion and generation
  - 6: Spoken dialogue system (Spoken chatbot)

Topic 1: Speech Processing Basics

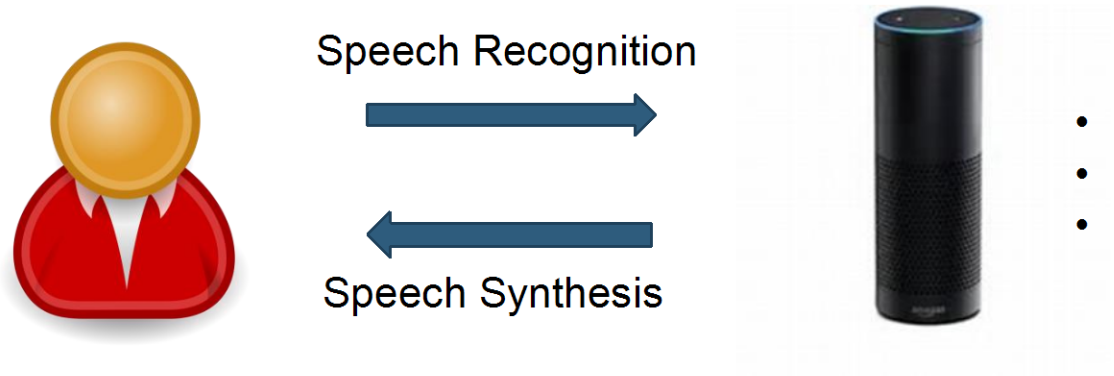
PART 1. SPEECH SIGNALS

Speech Applications

- **Voice Assistants and Smart Speakers**
  - Amazon Alexa, Google Assistant, Apple Siri, Microsoft Cortana.
- **Interactive Voice Response (IVR) Systems**
  - Customer service phone lines for banks, airlines, or utility companies
- **Voice Command Systems in Vehicles**
- **Voice-activated Home Automation**
- **Language Learning Apps**
- **Speech Therapy Tools**



Process of Spoken Dialogue



- Natural Language Understanding
- Dialogue Management
- Natural Language Generation

Automatic Speech Recognition

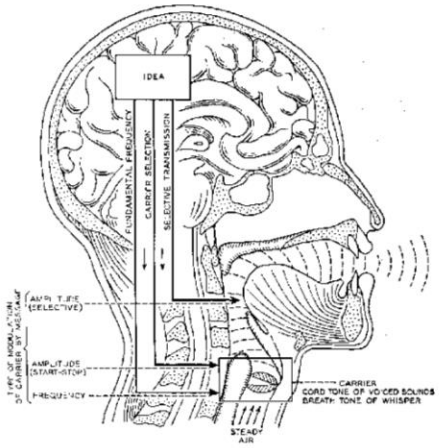
Speech (continuous time series) -> Text (discrete symbol sequence)

Speech Synthesis (Text-to-Speech)

Text (discrete symbol sequence) -> Speech (continuous time series)

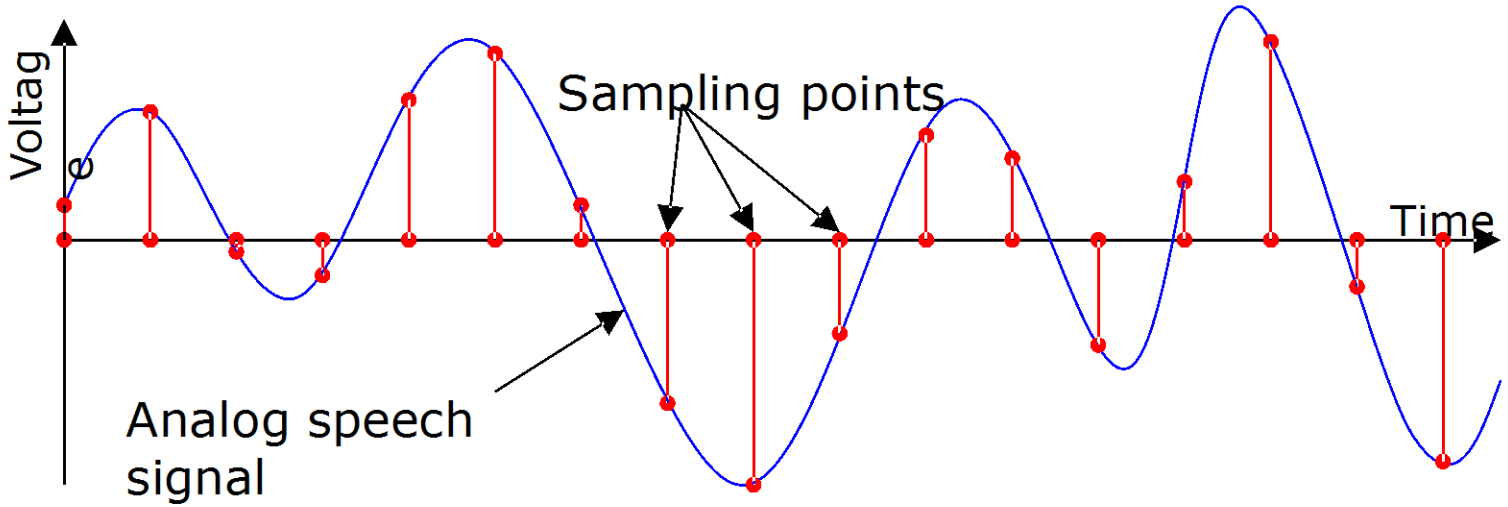
Speech Signal

- Human voice is captured by microphone
- Speech is recorded in computer as a sequence of numbers



Sampling Rate

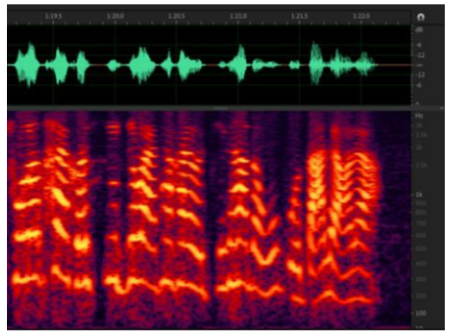
- The analog speech signal captures pressure variations in air that are produced by the speaker.
- The continuous analog speech signal is sampled at regular intervals to convert it into a sequence of discrete values.
- The rate at which the signal is sampled is called the sampling rate or sampling frequency.



Sampling Rate

- Analog speech signal contains many frequencies
- Human ear can perceive frequencies in the range 50Hz-15kHz.
- Ideally, a sampling rate of 30kHz or more is needed to capture all the details of speech (The Nyquist theorem)
- CD recordings: 44.1kHz
- For practical reasons, 8kHz (telephone) and 16kHz (PC and smartphone) are often used.

Speech Coding



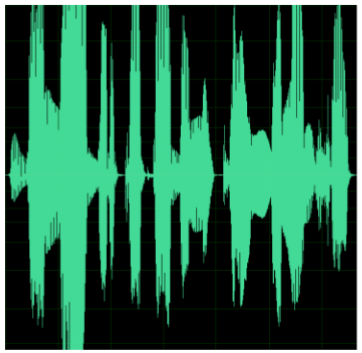
- **Quantization:**
  - Continuous signal level will be quantized to discrete values.
  - Each sample is represented with a fixed-point number in computer. (eg. 16bits, -32767 to 32767)
- **Encoding format:**
  - Linear coding method Pulse-Code Modulation (PCM) is the basic method of digital representation of audio signals. 16-bit PCM is mostly used in speech processing.
  - Non-linear coding: *A-law* (in Europe) and *μ-law* (used in US and Japan) encoding schemes use only 256 levels (8-bit encodings).
  - *A-law* and *μ-law* use logarithmic encoding techniques, increase the quantization resolution for low-amplitude signals, which improves the quality of quiet sounds

Audio File Formats

- **Popular audio formats**
  - Wav: Developed by Microsoft and IBM. Native format: PCM, Uncompressed lossless
  - MP3: A lossy data compression format.
  - FLAC: Free Lossless Audio Codec. Widely supported.
  - OGG: A free multimedia container format. Support many codecs.
- **In speech processing:**
  - Normally non-compressed PCM format is used for speech input.
  - Speech recognition models are often built for different sampling rates.

Recording Quality

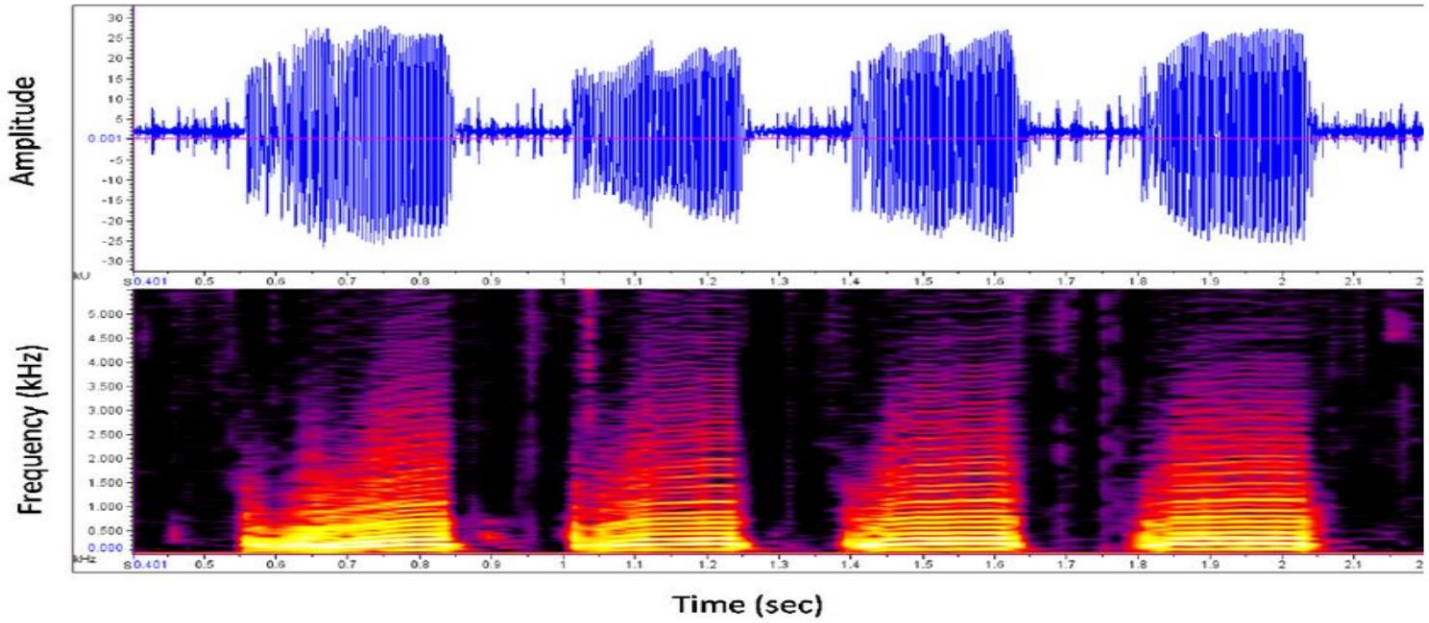
- **The microphone quality**
- **Environmental quality: ambient noise level**
- **Proper setting of the recording level**
  - Too low: losing resolution
  - To high: clipping (signal value exceeds maximum)



PART 2. SPEECH FEATURES

Spectrogram

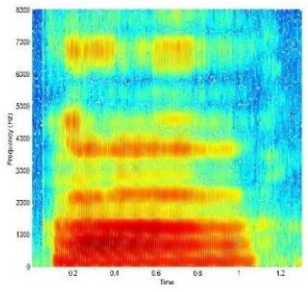
- Speech is a one-dimensional signal.
- To effectively analyse the signal, it is often converted into a two-dimensional image called spectrogram.
- Spectrogram shows the strength of different frequencies of the signal at any time.



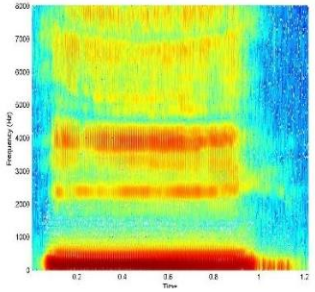
Spectral Differences for Pronunciations

- **Different sounds show different energy levels at different frequencies.**

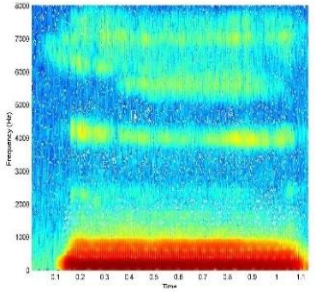
AA



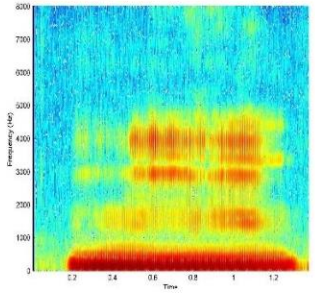
IY



UW

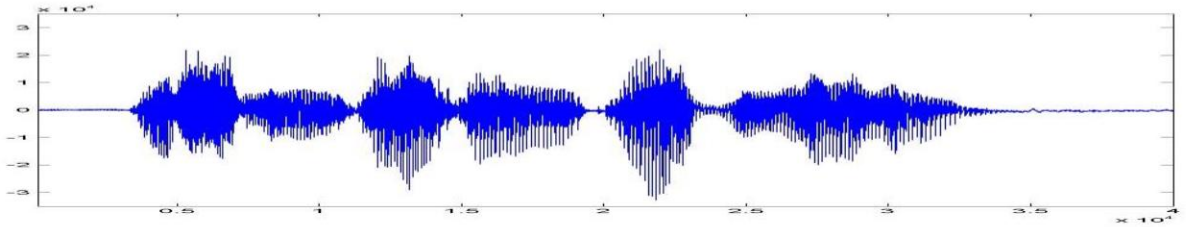


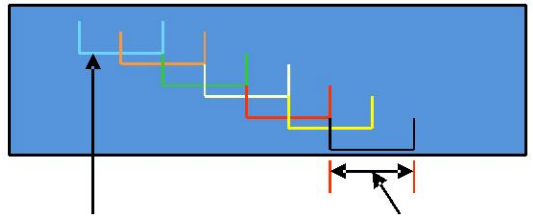
M



Frame

- Speech signal is normally processed by segments. Each segment is called a frame.
- Frame Size: size of the speech segment
- Frame Shift: number of samples shifted to the next frame
- Frame can be overlapped with each other.



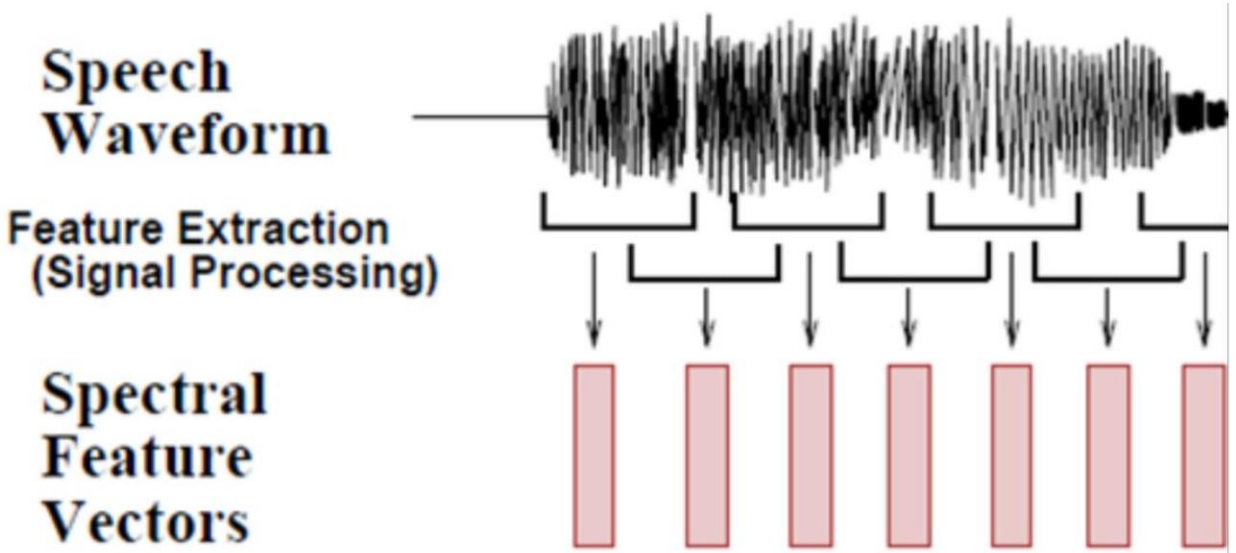


Segments shift every
10 milliseconds

Each segment is typically
20 or 25 milliseconds wide

Speech Feature

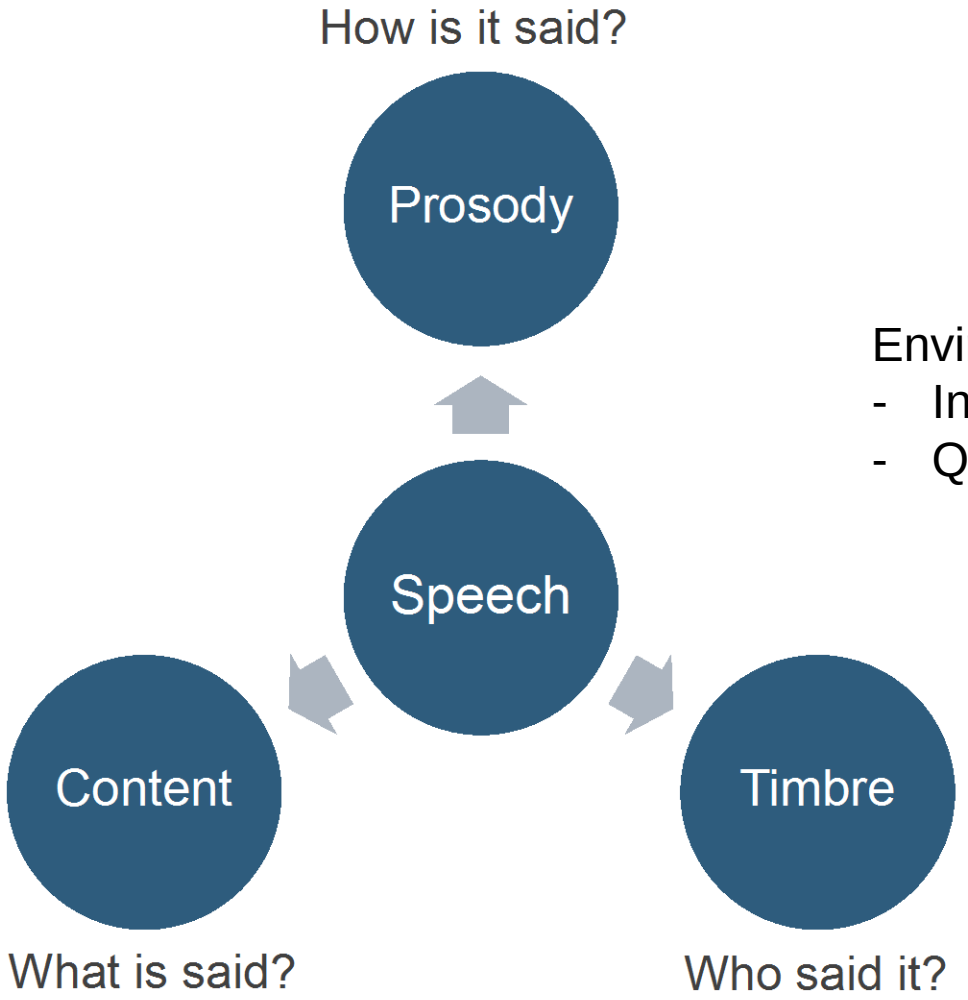
- Speech signal is analysed frame by frame.
- Each frame can be converted into vector.
- So, a speech signal can be represented with a sequence of feature vectors.



Common Speech Features

- **Mel-Frequency Cepstral Coefficients (MFCCs)**
  - Represent the short-term power spectrum of sound based on a nonlinear mel scale of frequency. Capture the phonetic content of the signal.
  - Widely used in speech and speaker recognition due to their ability to capture the characteristics of human speech effectively.
- **Linear Predictive Coding (LPC)**
  - LPC analyzes the spectral envelope of a speech signal. It predicts the current sample based on its previous samples and is used to estimate the formants in speech.
  - Common in speech compression, speech synthesis, and recognition

Information in Speech



Environmental information as well
- Indoor / Outdoor
- Quiet / Noisy

Content and Timbre

- **Content:**
  - Text transcription of speech signal
  - Speech recognition is to derive the content from speech signal
  - Speech synthesis is to implement text information with speech signal.
- **Timbre**
  - Timbre represents the speaker information of the speech.
  - Different speaker has different voice timbre.

Prosody

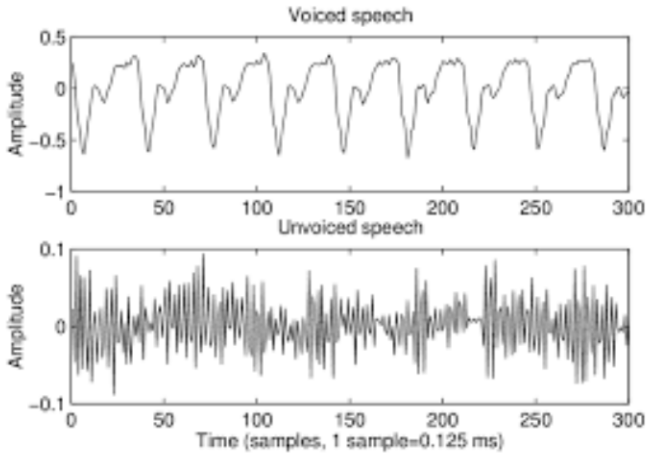
- **What is prosody (from perception level)**
  - The same text can be read in different ways. The way to read the text is determined by prosody.
  - Example: I bought two books from the shop.
  - Prosody is perceived as intonation, rhythm, pause, emotion, speaking styles, speech rate, etc.
- **Major elements (from acoustic level)**
  - Fundamental frequency (pitch of voiced signals)
  - Duration (length of each phonetic unit)
  - Energy (loudness of each phonetic unit)

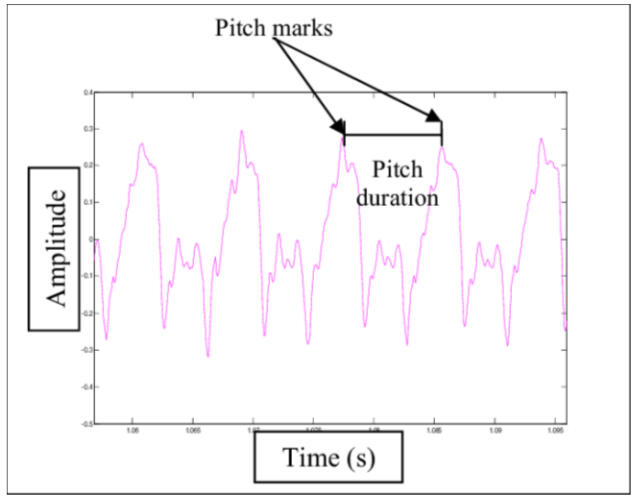
Examples of Perceived Prosody

- **Intonation**
  - Statements normally have a falling intonation. Questions may have rising intonation.
- **Lexical tone**
  - Mandarin has four lexical tones.
- **Emphasis**
  - Some words are emphasized in speech
- **Emotion**
  - Angry and happy speeches have different prosody.
- **Phrase break**
  - Phrase break within sentence is also part of prosody.

Fundamental Frequency (pitch)

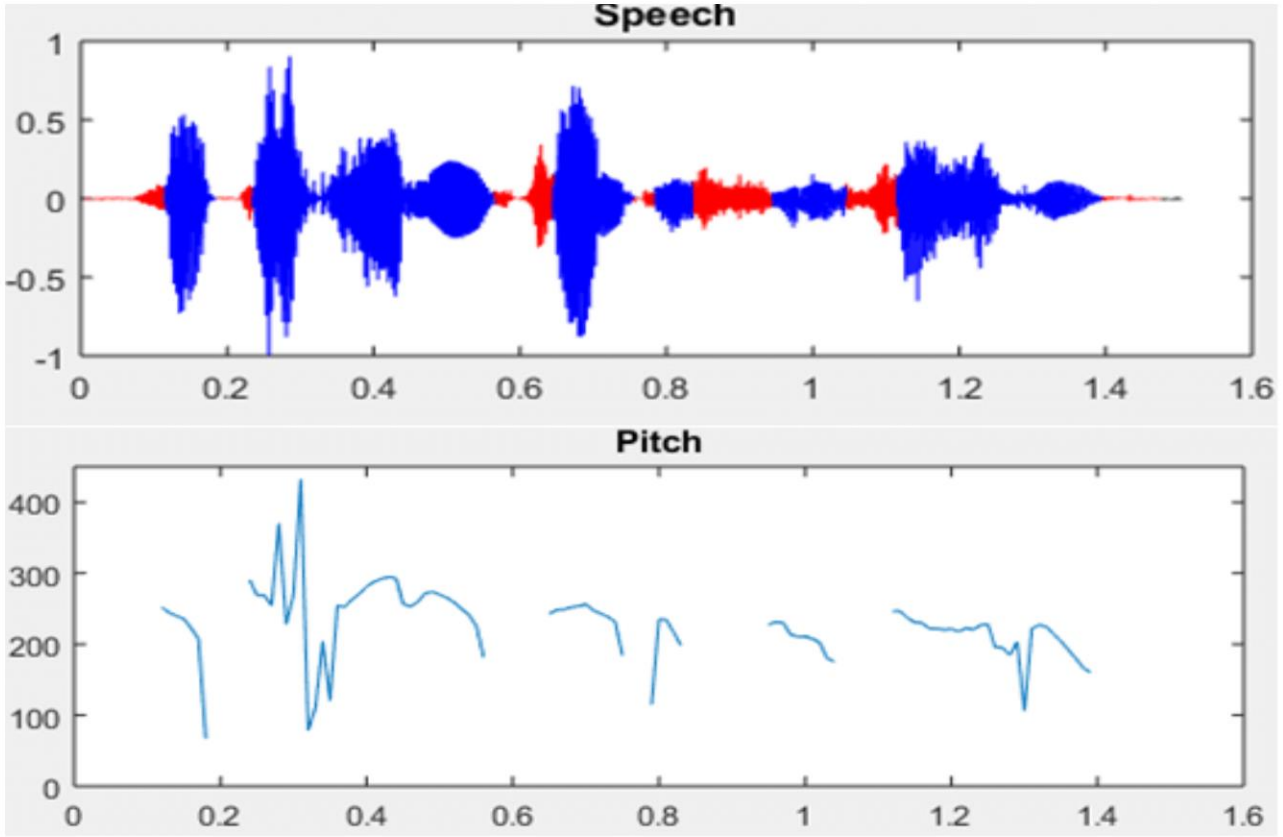
- **Voiced and Unvoiced**
  - Voiced speech signals are produced when the vocal cords vibrate. The rest are unvoiced signals.
  - Voiced signals are periodic ones. Unvoiced signals are noise.
  - All vowels are voiced.
  - Eg. /s/, /f/ are unvoiced; /z/, /v/ are voiced
- **Pitch**
  - Pitch exists in voice signals only.
  - Fundamental Frequency (F0)
  - Pitch duration: duration between two pitch marks.
  - Frequency = 1 / period





Fundamental Frequency (pitch)

- Fundamental frequency is referred to as Pitch or F0.
- Pitch is especially important in speech perception or generation.



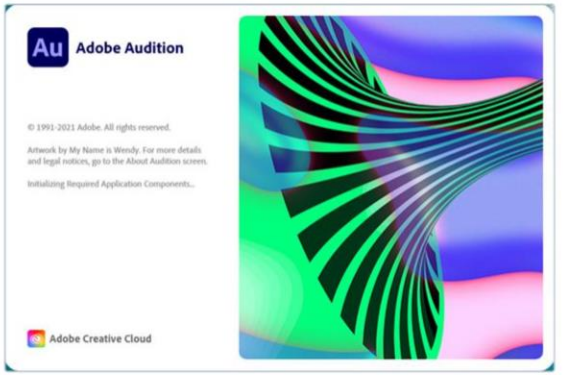
Common Speech Preprocessing

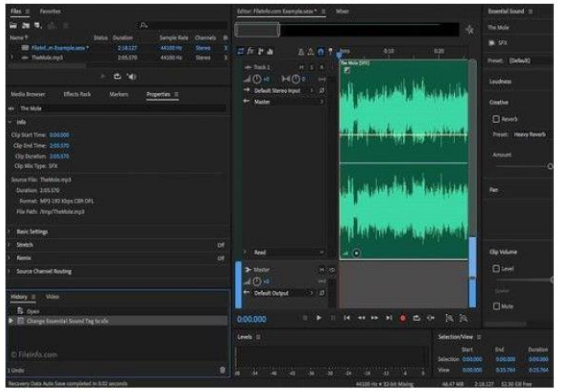
- **Voice Activity Detection (VAD)**
  - Microphone may be on all the time. But computer only start processing when voice is detected.
  - VAD is to find the valid speech segments to process.
- **Speech Enhancement**
  - In many cases, speech signal needs to be enhanced, and noise needs to be reduced.
  - Enhancement is to improve the speech quality with signal processing methods.
- **Speech Normalization**
  - To convert the speech signal to keep consistency.
  - Time domain normalization.
  - Frequency domain normalization

PART 3. AUDIO SOFTWARE

Audio software

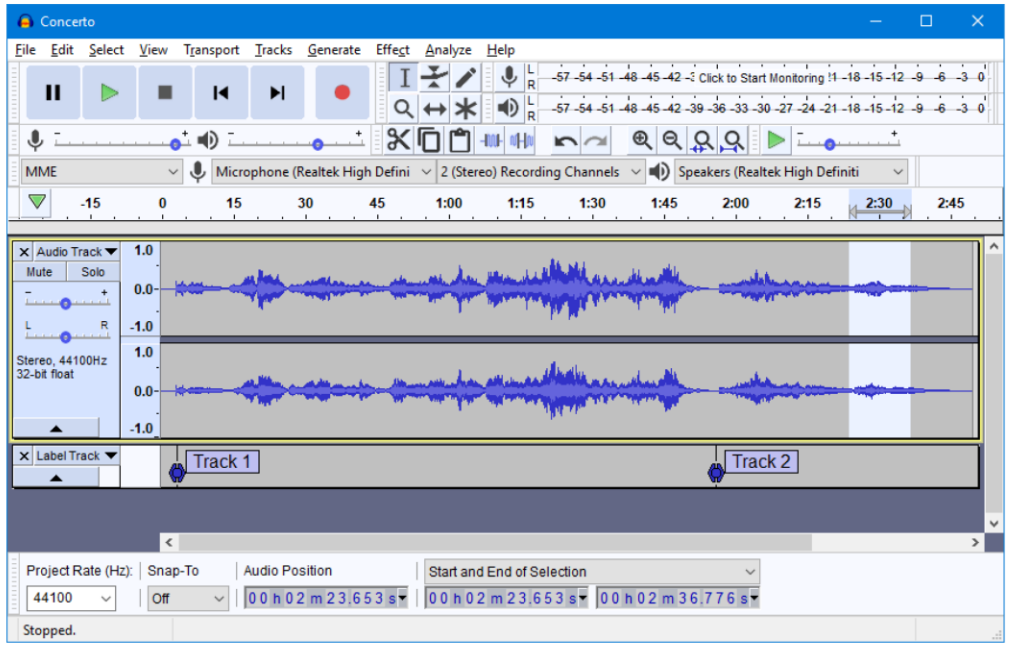
- **Adobe Audition**
  - Professional Audio Editor.
  - Record, view and play audio signals.
  - Cut, copy, paste, mix
  - Normalize, filter, remove noise
  - Show spectrogram
  - Change sampling rate, convert file format





Audio software

- **Audacity**
  - Audacity: https://www.audacityteam.org/
  - A free and open-source audio editor and recording software.
  - Available for Windows, macOS, Linux.



Audacity

- **Record and save file**
  - Select sampling rate, file format, resolution
- **View and edit**
  - Zoom in, Zoom out
  - Cut, copy, past, etc
  - Generate silence, noises
- **Mix and effect**
  - Stereo to mono
  - Normalize, noise reduction,
  - Change pitch, speed, etc

Audio Software

- **Sox (Sound of Exchange)**
  - Sox: https://sourceforge.net/projects/sox/
  - An open-source cross-platform audio editing software
  - Command line tool.
  - Converting sampling rate, bits, stereo/mono, etc
  - Editing: Concatenate, trim, pad, repeat, reverse, volume, fade, normalise
  - Effects: chorus, flanger, echo, phaser, compressor, delay, filter
  - Adjustment of speed, pitch, tempo, etc

Sox

- **Examples:**
  - **sox --i test1.wav**
    show information of the file
  - **sox test1.wav -r 8k test1-out-8k.wav**
    change sampling rate
    -r 16k = sampling rate:16khz,
  - **sox test2.wav -c 1 test2-out.wav**
    convert to mono wave file
    -c 1 = single channel (mono)

Sox

- **Examples:**
  - **sox -r 8k -b 8 -c 1 -e signed test3.raw test3-out.wav**

    raw format wav format

    -e signed = signed integer
  - **sox test3.wav test2.wav longfile.wav**

    Concatenate two files into one long file.
  - **sox test2.wav test2-fast.wav speed 1.1**

    Adjust speed of the speech

Audio Software

- **WaveSurf**
  - Open-source tool for sound visualization and manipulation
  - Cross-platform support – runs under Linux, macOS, and Windows
  - Website: https://sourceforge.net/project/wavesurder

PART 4. AUDIO PROGRAMMING

Python tool and audio libraries

- Anaconda
  - A distribution of Python programming language.
- SoundFile, Wave, scipy.io.wavfile, audioread
  - Libraries for reading and writing audio files
- PyAudio, SoundDevice
  - Libraries for playing and recording speech files
- LibROSA
  - Library for music and audio analysis
  - Feature extraction, spectrogram display
  - Voice effects

Note: You can ask ChatGPT (or Gemini in Colab) code segments for specific operations.

Anaconda - Python

- https://www.anaconda.com/
- A python distribution for scientific computing.
- Supports Windows, Linux, MacOS
- Contains basic packages and programming tools.
- Spyder: An interactive development environment (IDE) tool
- Jupyter Notebook: A Web-based IDE tool

SoundFile – Python audio library

- **Download and Install:**
  - https://pypi.org/project/SoundFile/
  - https://pysoundfile.readthedocs.io/en/latest/
  - pip install soundfile
  - sudo apt-get install libsndfile1
- **Features:**
  - Support WAV, FLAC, OGG, MAT files.
- **Program in Python**
  - import soundfile as sf
  - data, samplerate = sf.read('existing_file.wav')
  - sf.write('new_file.flac', data, samplerate)

PyAudio – Audio playing and recording

- **Install**
  - https://people.csail.mit.edu/hubert/pyaudio/
  - python -m pip install pyaudio (windows)
  - sudo apt-get install python-pyaudio python3-pyaudio (linux)
- **Programming**
  - import pyaudio
  - p = pyaudio.PyAudio()
  - stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
  - channels=wf.getnchannels(), rate=wf.getframerate(), output=True)
  - data = wf.readframes(1024)

LibROSA – Feature calculation and display

- **Installation**
  - https://librosa.org/
  - pip install librosa
- **Programming**
  - D = librosa.amplitude_to_db(np.abs(librosa.stft(y)), ref=np.max)
  - plt.figure()
  - librosa.display.specshow(D, y_axis='linear')
  - plt.colorbar(format='%+2.0f dB')
  - plt.title('Linear-frequency power spectrogram')

Topic 2: Speech Recognition

PART 1. INTRODUCTION

What is Speech Recognition

- Automatic Speech Recognition (ASR) or Speech-to-Text is a process to automatically convert speech into text
- A natural way to input information to computer.
- The first step to understand speech. No understanding of the meaning yet.
- Convert digital signal (a sequence of continuous values) into text (discrete symbol representations)

Difficulties

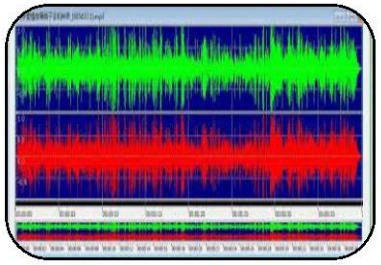
Device and Channel

Background noise

Speaker differences

Similar pronunciations

Accent, dialect



Pronunciation Variations in continuous speech *

Prosody differences

Words with multiple pronunciations

E.g. “Read” in English

- Present tense: /ri:d/ (reed)
- Past tense: /rɛd/ (red)

*Further Reading: Read vs Spontaneous Speech (https://languagelog.ldc.upenn.edu/nll/?p=60956)
Reverberation and overlapping audio sources as well
https://www.youtube.com/watch?v=-gi2P-FGgKg
https://www.youtube.com/watch?v=etTcru7CrWU
https://www.youtube.com/watch?v=JolecbvEW6U

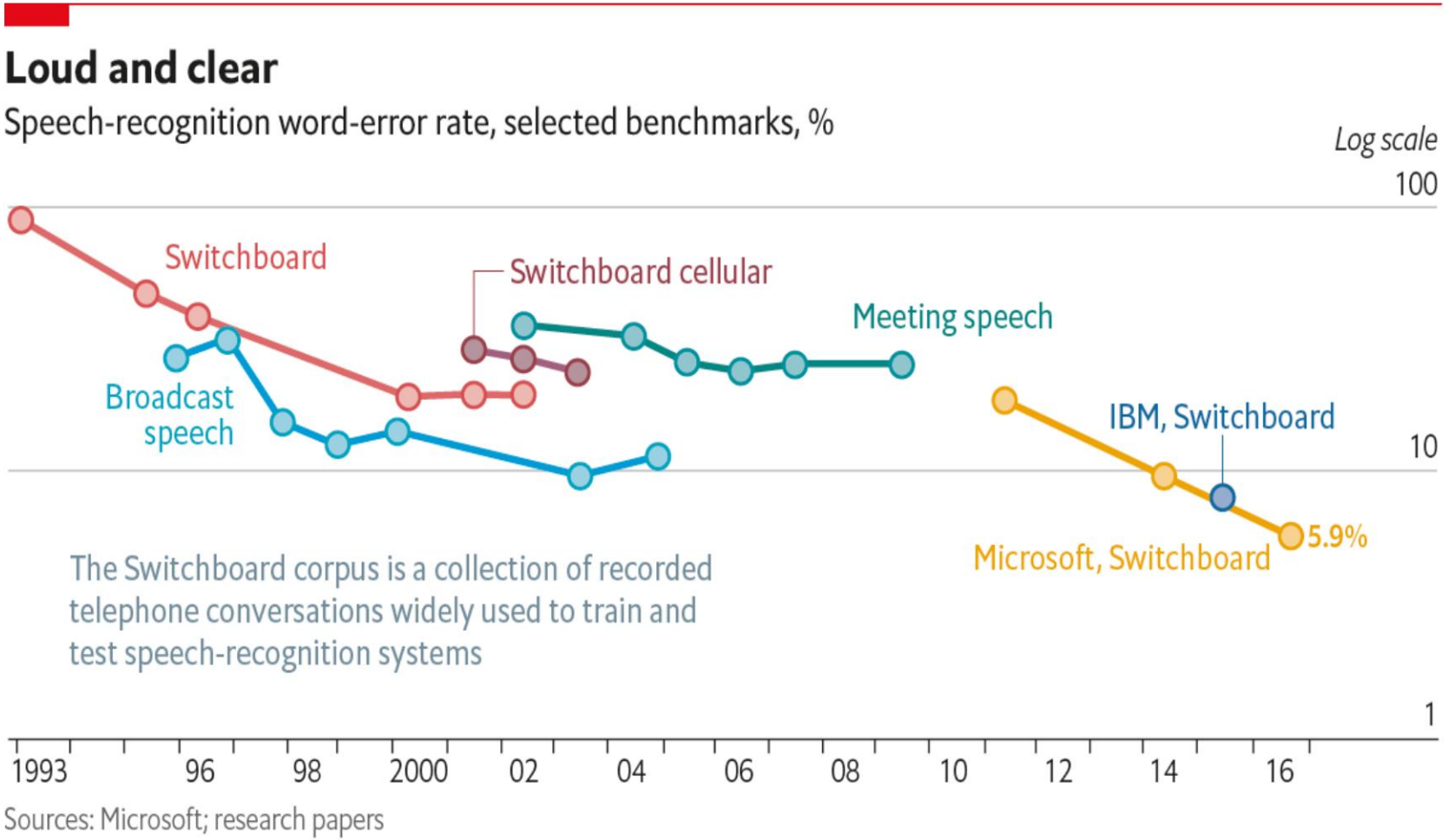
History of Speech Recognition

1. **1950s - Early Beginnings**:
   - Bell Labs developed the "Audrey" system in 1952, which could recognize spoken digits.
2. **1960s - Pattern Recognition**:
   - Began using statistical methods to recognize speech patterns. The "Shoebox" machine developed by IBM could recognize 16 English words.
3. **1970s - Template Matching**:
   - Template matching: to compare spoken words to pre-recorded templates.
   - DARPA (Defense Advanced Research Projects Agency) began funding speech recognition research, leading to significant advancements.
4. **1980s - Hidden Markov Models (HMMs)**:
   - The introduction of Hidden Markov Models in the 1980s was a major breakthrough. HMMs could model the statistical properties of speech and became the dominant approach for several decades.
   - The first commercial speech recognition systems began to appear.

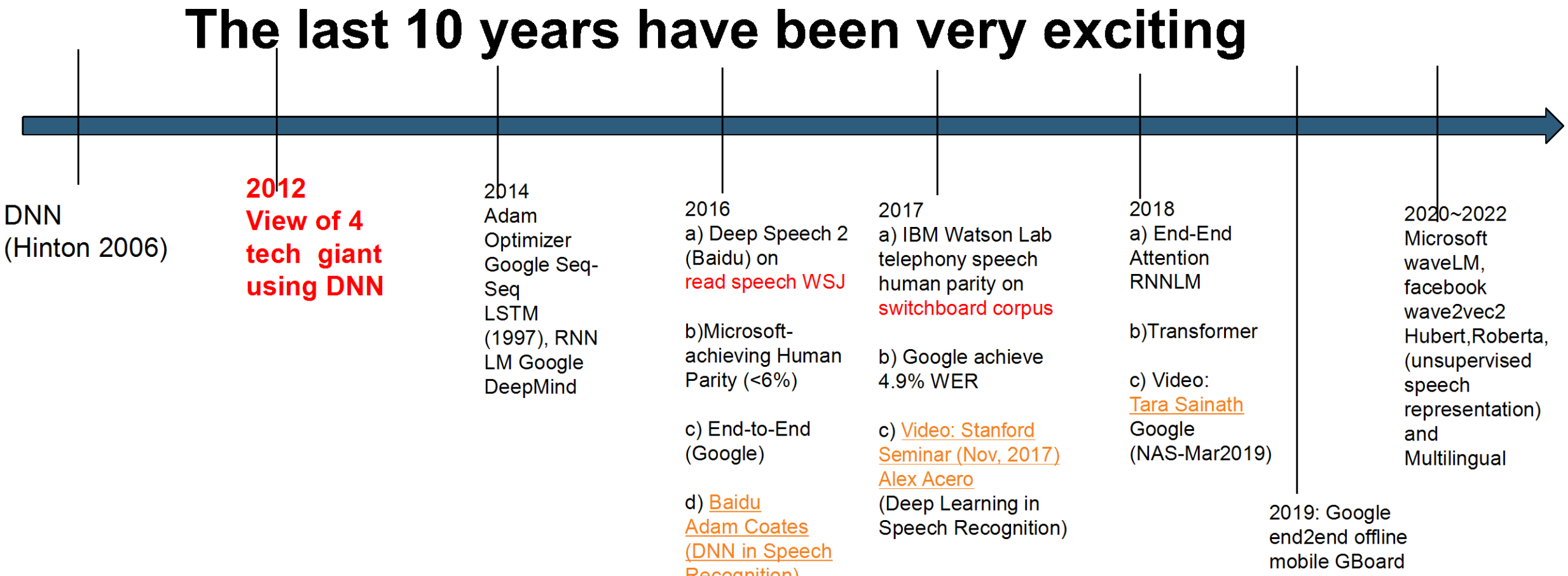
History of Speech Recognition (2)

5. **1990s - Continuous Speech and Large Vocabulary**:
   - Recognizing continuous speech (rather than isolated words). System can recognize a large vocabulary.
   - Dragon Systems released "Dragon NaturallySpeaking" in 1997, the first general-purpose continuous speech recognition program.
6. **2000s - Data-Driven Approaches**:
   - With the rise of the internet and the availability of large datasets, data-driven approaches became more feasible.
   - Voice-controlled assistants like Apple's Siri (introduced in 2011) and Google's Voice Search utilized cloud-based processing to improve accuracy.
7. **2010s - Deep Learning Revolution**:
   - The resurgence of neural networks, particularly deep learning, brought significant improvements to speech recognition.
   - Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs) and Transformer architectures, became popular for modeling sequential data like speech.
   - Google, Microsoft, and other tech giants reported achieving human-parity or near-human parity in certain speech recognition tasks.
8. **2020s and Beyond**:
   - Making models more efficient, understanding context, handling multiple languages and accents, and operating in noisy environments.
   - Edge computing allowed powerful speech recognition on-device without always needing a cloud connection.
   - Growing emphasis on preserving user privacy and understanding the ethical implications of voice data collection and usage.

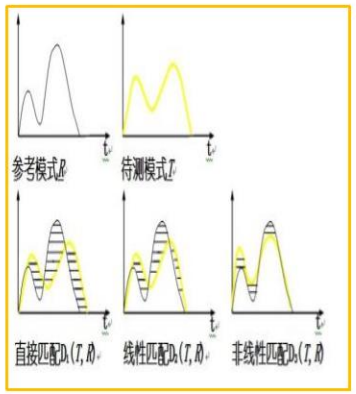
ASR in recent years

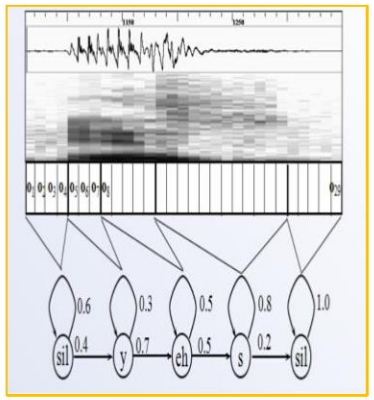


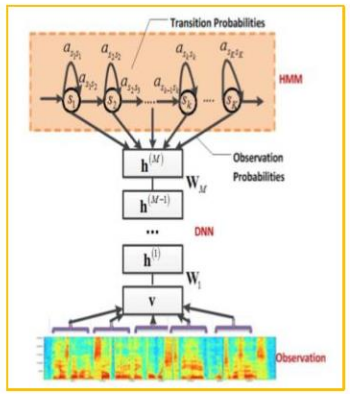
Deep Learning ASR - Recent TimeLine

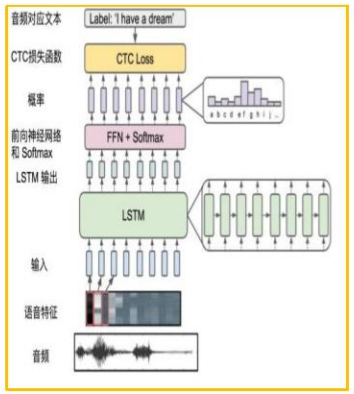


Evolution of ASR methods





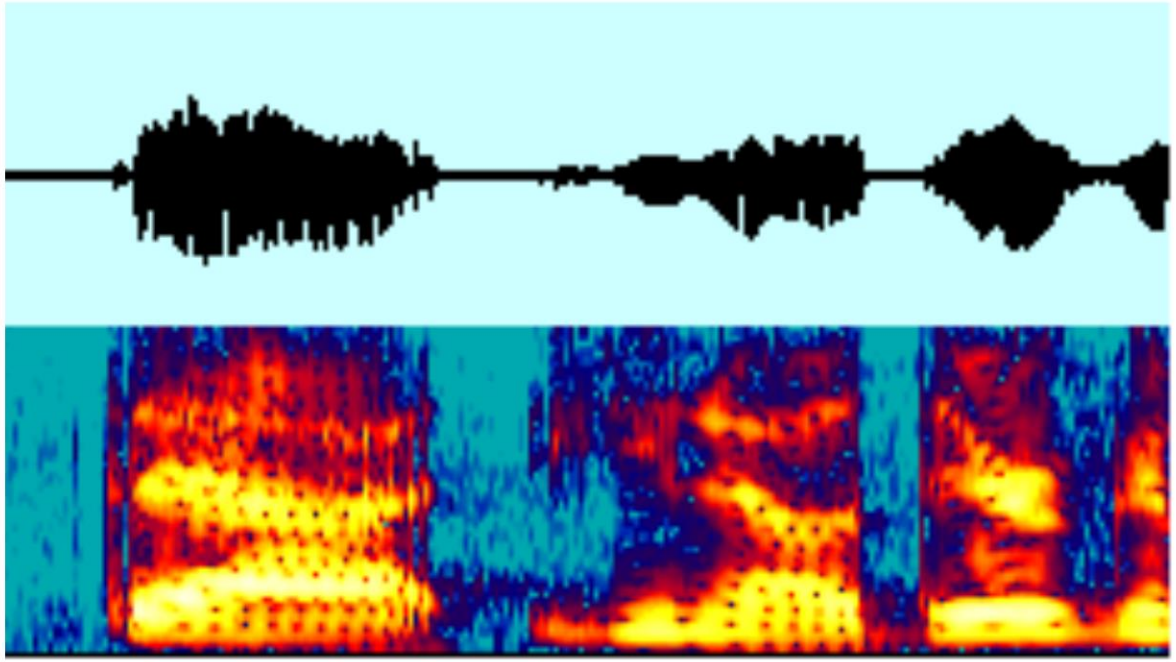




Pattern matching	HMM models	Deep learning	End-to-End ASR
• Since 1952 • Isolated words	• Since 1980 • GMM-HMM • Continuous speech • Limited accuracy	• Since 2009 • DNN-HMM • Huge language model • Natural speech	• Since 2017 • Neural network • Huge speech database • Natural speech

Feature Calculation

Input:



Output:

-0.1	0.2	0.2	-6.1
0.3	0.1	0.0	-2.1
1.4	1.2	1.2	3.1
-1.2	-1.2	-1.2	2.4
2.3	4.4	4.4	1.0
2.6	2.2	2.2	2.2
…	…	…	…

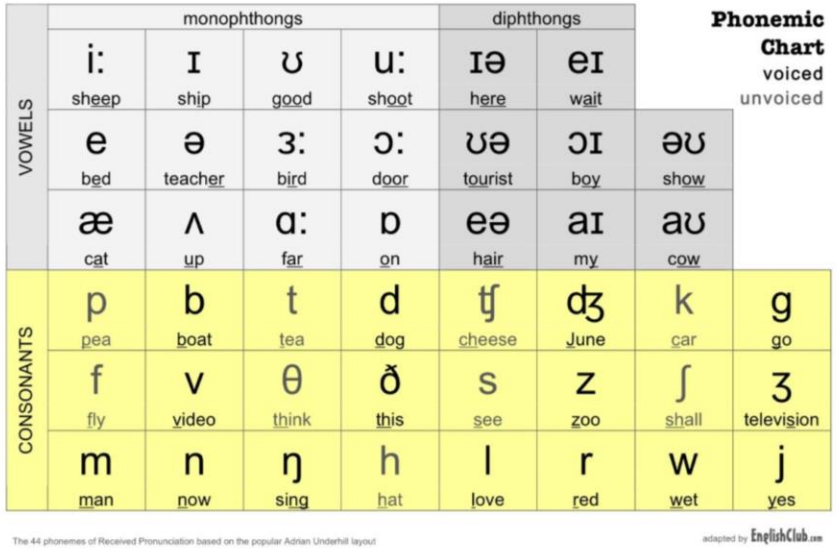
…

Typical speech features:

- 39-dim MFCC
- 80-dim filter bank output
- 400 sample points

Pronunciations

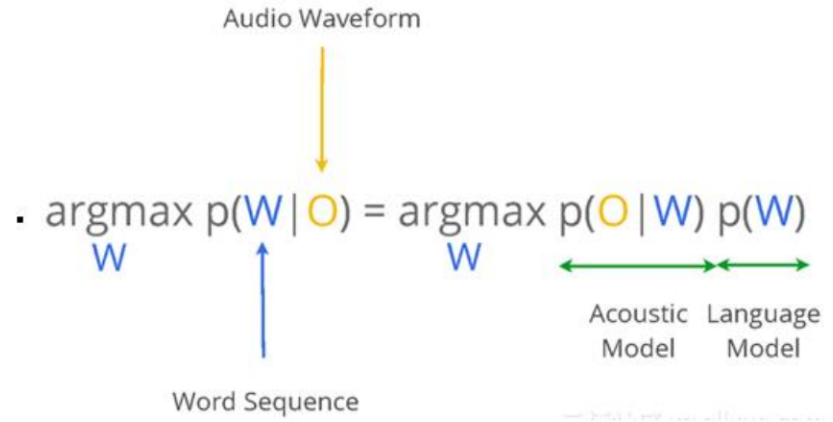
- **Phoneme:**
  - Smallest pronunciation unit to represent meanings
  - E.g. cat: K AE T K, good: G UH D G
  - Phone: The actual pronunciations of phoneme
  - There are about 44 phonemes in English



Source: https://www.englishclub.com/

PART 2. TRADITIONAL METHODS

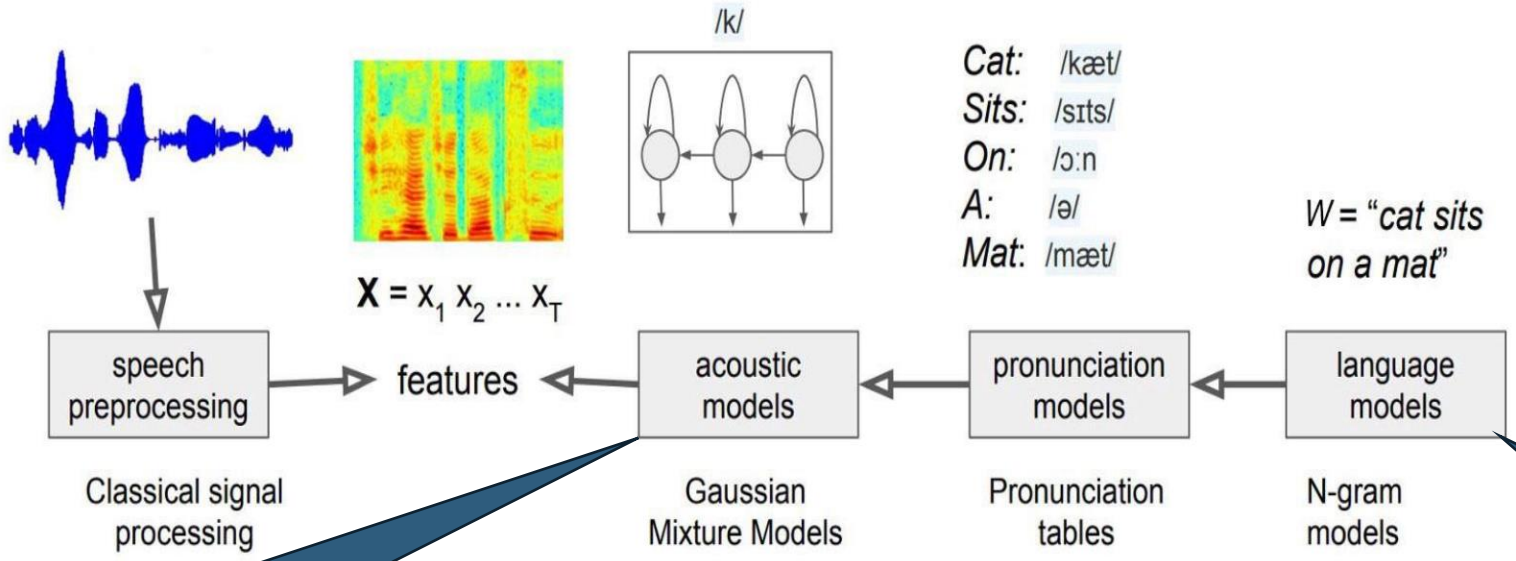
Speech Recognition Framework



$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

- **Acoustic model:**
  - To evaluate speech features against pronunciations
  - How likely the speech signal sounds like the word sequence.
- **Language model:**
  - To evaluate whether the word sequence is reasonable
  - How likely the word sequence like a correct sentence

ASR System



Evaluate how likely a given speech signal corresponds to a sequence of sound units

$$W^* = \arg\max_W p(\mathbf{X}|\mathbf{W})\, p(\mathbf{W})$$

acoustic model

language model

Evaluate how likely a certain word sequence is correct

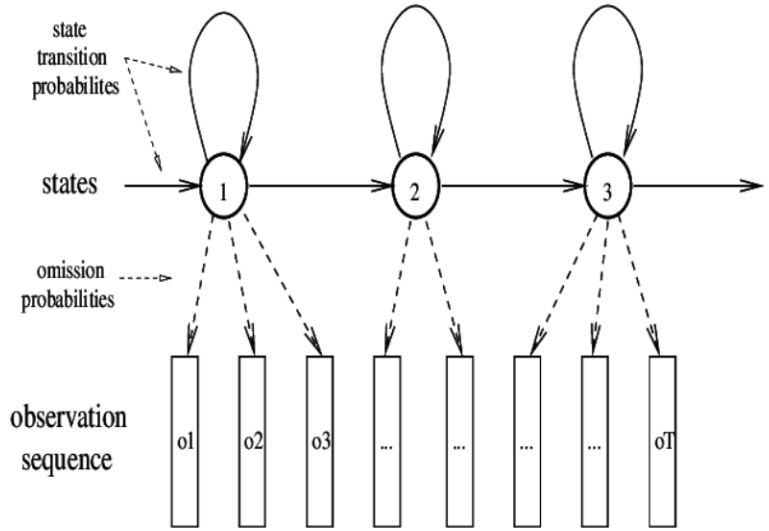
Image source: https://heartbeat.fritz.ai/the-3-deep-learning-frameworks-for-end-to-end-speech-recognition-that-power-your-devices-37b891ddc380

Acoustic Model

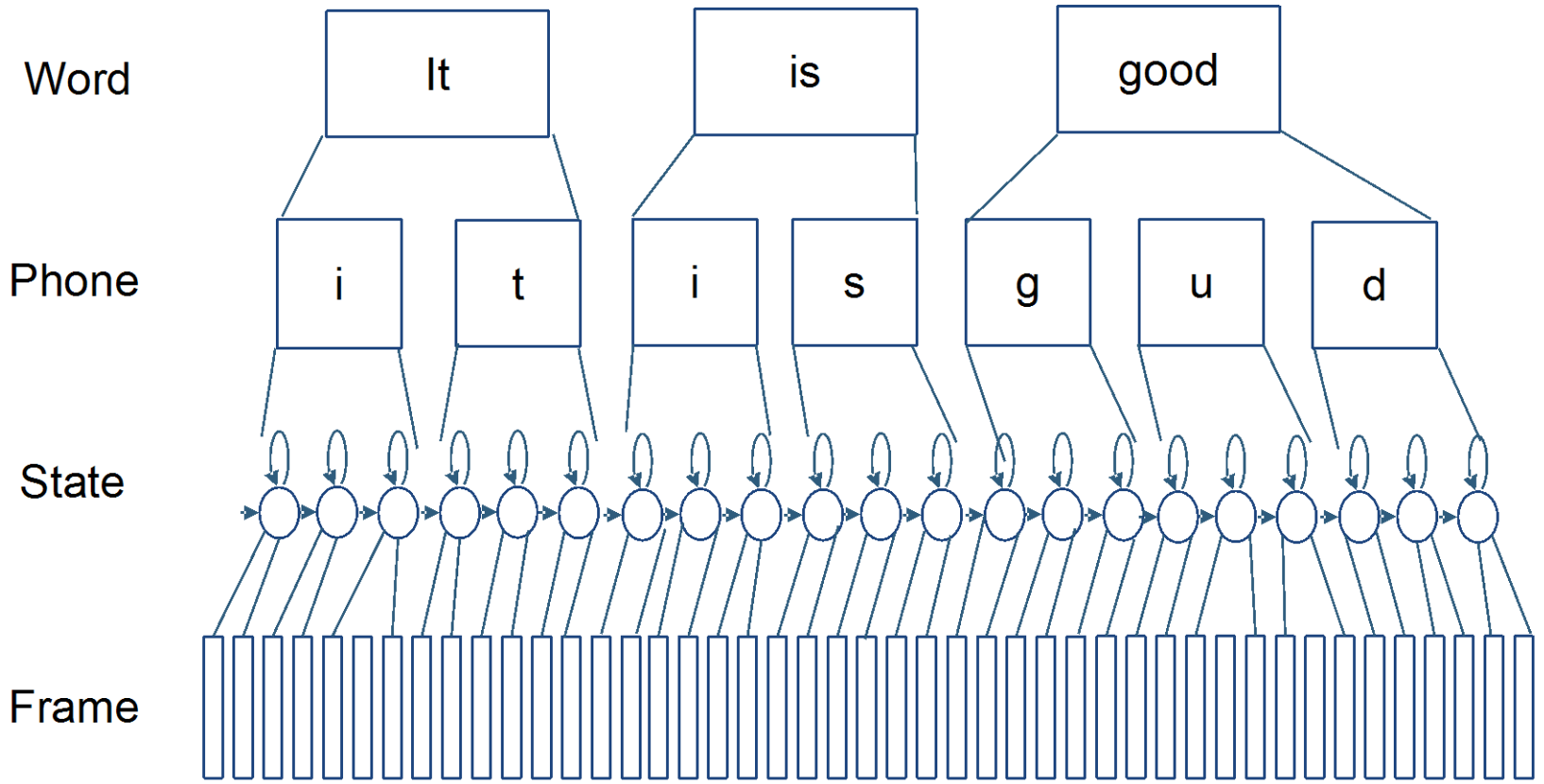
- Most common method: Hidden Markov Models

- Each phone is defined as a HMM model
- Each model contains 3-5 states.
- Speech frames are mapped to state

- Speech Recognition: To calculate **P(O|W)**

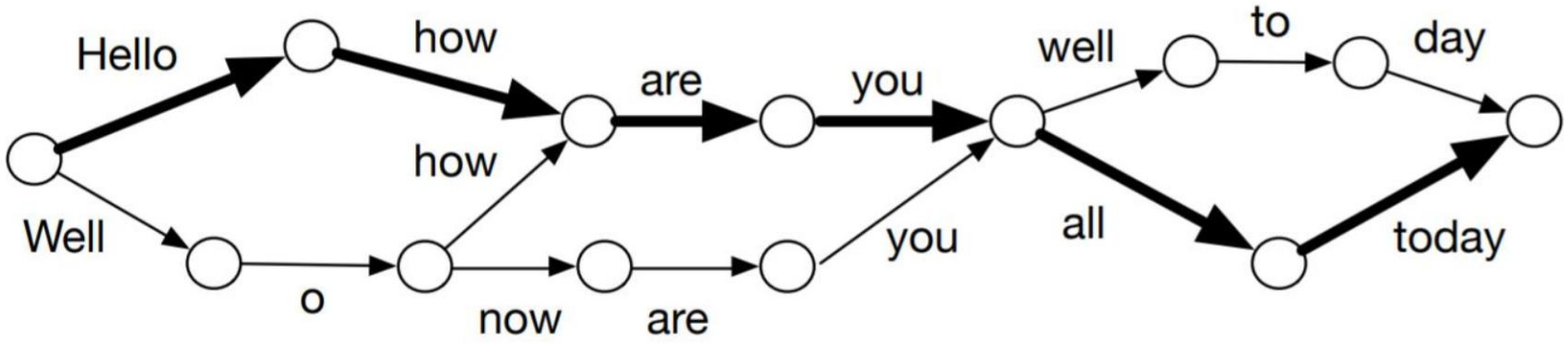


HMM for Sentence



Language Model

- To find the most like word sequence from all the possibilities.
- To calculate P(W)



NN-Grams: Unifying Neural Network and n-Gram Language Models for Speech Recognition, Babak Damavandi, Shankar Kumar, author Antoine Bruguier, INTERSPEECH 2016

Language Model

$$\text{Unigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n)$$

$$\text{Bigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_{n-1})$$

$$\text{Trigram LM} : p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_{n-2}, w_{n-1})$$

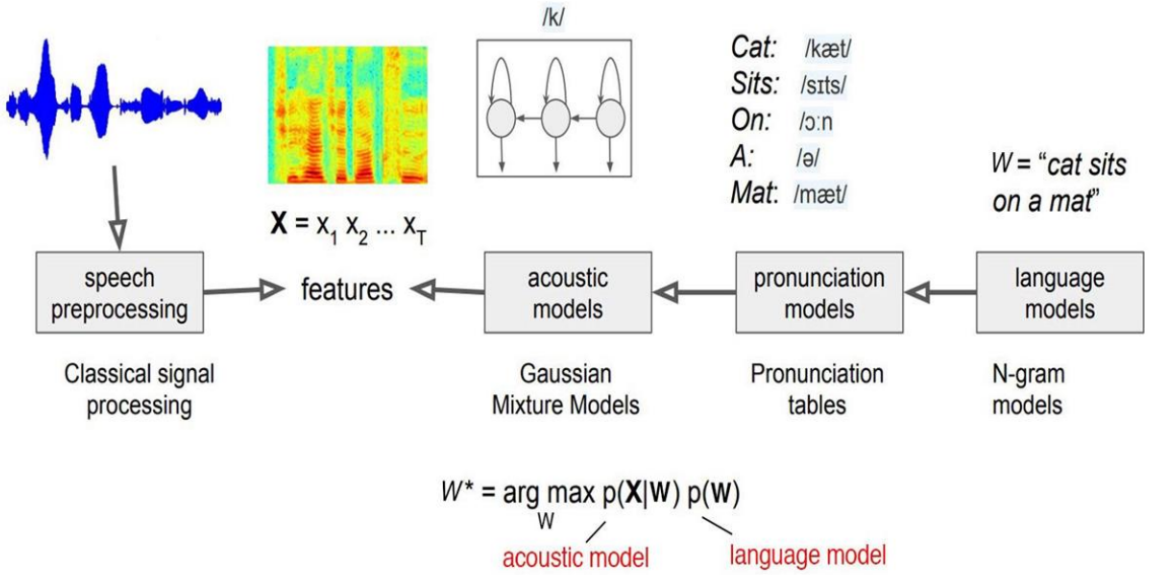
- **N-gram models calculate the likelihood of a word given previous word(s)**
- **Language models are trained with text corpus.**

PART 3. END-TO-END SPEECH RECOGNITION

Hybrid vs. End-to-End (E2E) Modeling

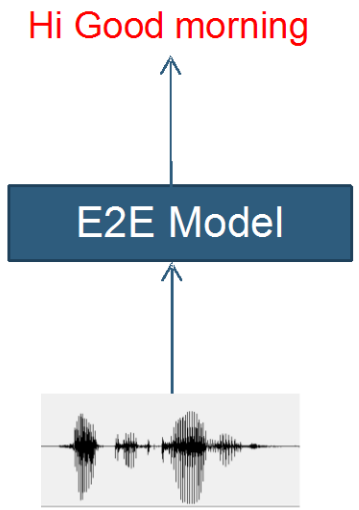
Hybrid

Each model is trained separately, and then the models are integrated together.



E2E

A single model is used to directly map the speech waveform into the target word sequence.



Advantages of E2E Models

- **E2E models use a single objective function which is consistent with the ASR objective**
- **E2E models directly output characters or even words, greatly simplifying the ASR pipeline**
- **E2E models are much more compact than traditional hybrid models -- can be deployed to devices with high accuracy and low latency**

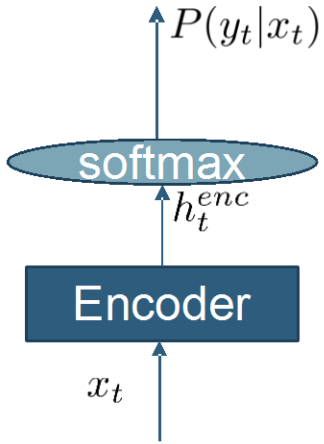
Graves and Jaitly, “Towards end-to-end speech recognition with recurrent neural networks” PMLR, 2014.
Hannun et al., “Deep speech: Scaling up end-to-end speech recognition,” in arXiv preprint, 2014.

Current Status

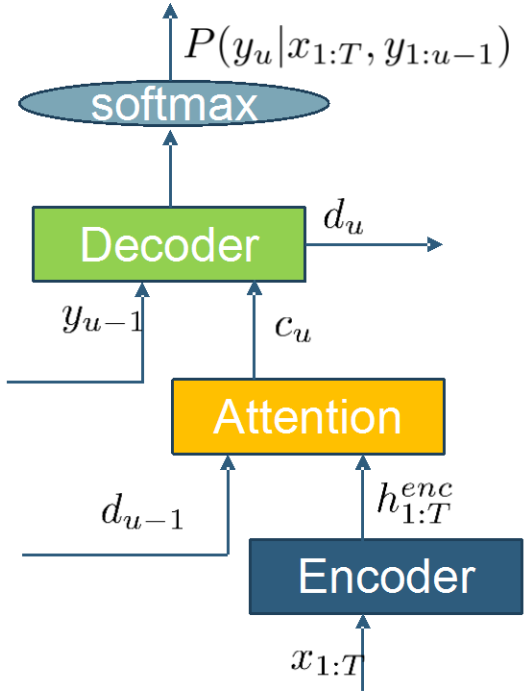
- E2E models achieve the state-of-the-art results in most benchmarks in terms of ASR accuracy.
- Practical challenges such as streaming, latency, adaptation capability etc., have been also optimized in E2E models.
- E2E models are now the mainstream models not only in academic but also in industry.

Li, Jinyu. "Recent advances in end-to-end automatic speech recognition." *APSIPA Transactions on Signal and Information Processing*, 2022.

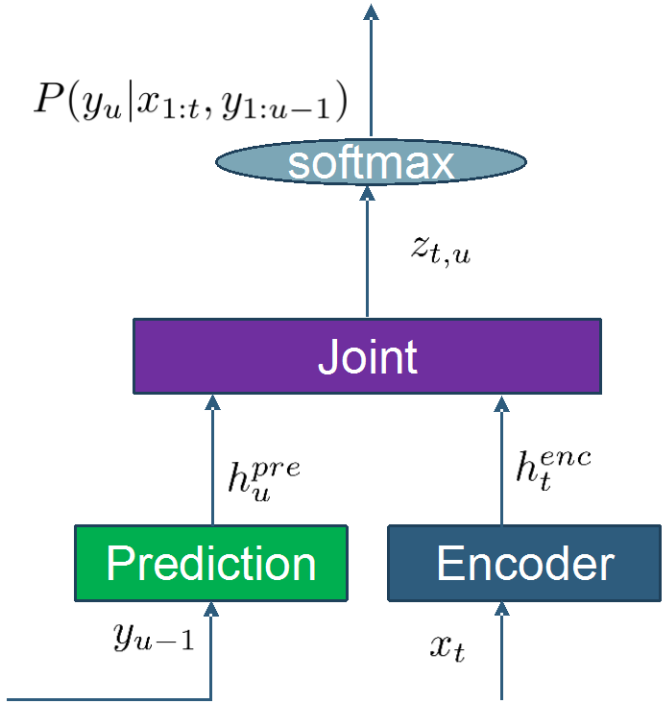
E2E Models



Connectionist Temporal Classification (CTC)



Attention-based encoder decoder (AED)



RNN-Transducer (RNN-T)

End-to-End Speech Recognition

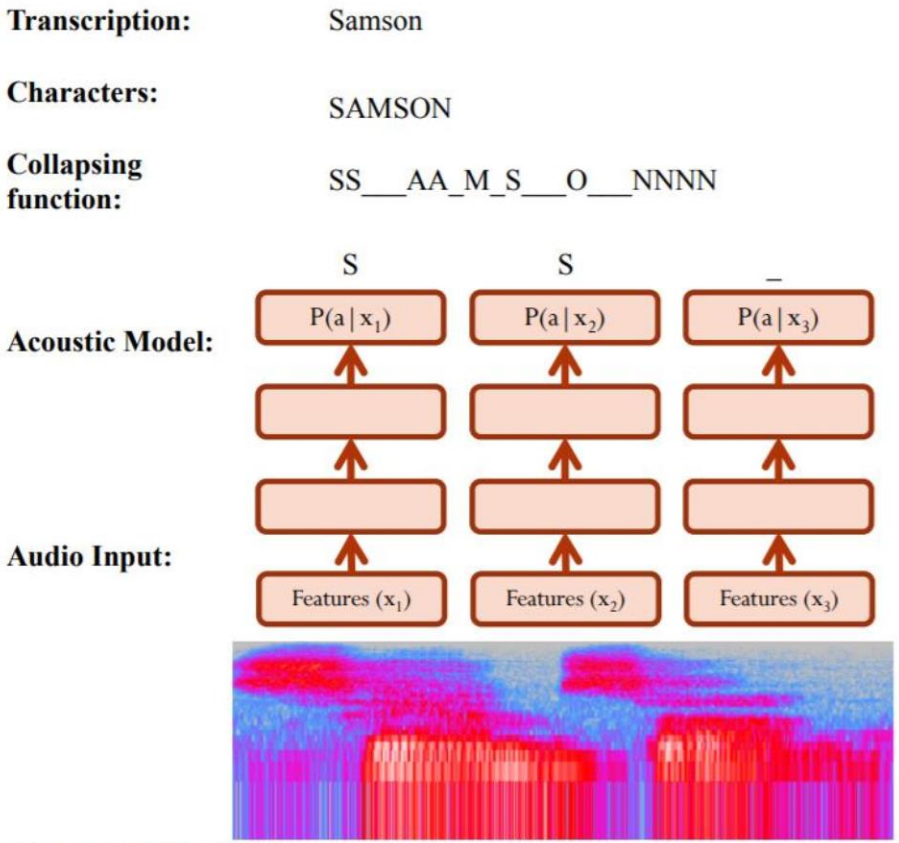
- **Connectionist Temporal Classification (CTC) method**
  - Generate a letter sequence first (with duplicated characters). Then to convert the letter sequence into text.

- **Listen, Attend, and Spell (LAS) method (AED)**
  - Encoder/decoder with attention. Directly generate sequence.

- **RNN Transducer (RNN-T)**
  - Integrate language model into prediction. Do not rely on the full sequence.

CTC-based End-to-end ASR

- Multiple modules are merged into one network for joint training.
- Sequence to Sequence model.
- It directly maps the input acoustic feature sequence to the text result sequence
- Use DNN to approximate distribution over characters.
- Simple collapsing function to generate results



(Graves & Jaitly. 2014)

CTC

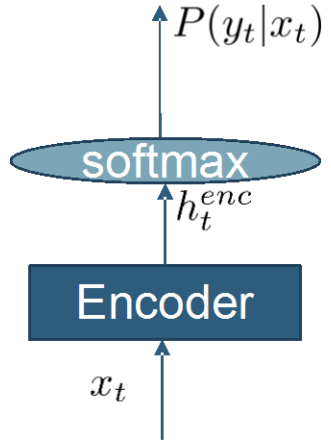
- **The first and simplest E2E ASR model.**
- **To solve the challenge that target label length is smaller than the speech input length:**
  - Inserts blank and allows label repetition to have the same length of CTC path and speech input sequence.

$$P(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{q}\in\mathbf{B}^{-1}(\mathbf{y})} P(\mathbf{q}|\mathbf{x})$$

- **Frame independence assumption**

$$P(\mathbf{q}|\mathbf{x}) = \prod_{t=1}^{T} P(q_t|\mathbf{x})$$

- **Revives with the Transformer encoder and the emerged self-supervised learning technologies**



Graves et al., "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in Proc. ICML, 2006.

CTC ASR Example Results

YET A REHBILITATION CRU IS ONHAND IN THE BUILDING LOOGGING BRICKS PLASTER AND BLUEPRINS FOUR FORTY TWO NEW BETIN EPARTMENTS

YET A REHABILITATION CREW IS ON HAND IN THE BUILDING LUGGING BRICKS PLASTER AND BLUEPRINTS FOR FORTY TWO NEW BEDROOM APARTMENTS

THIS PARCLE GUNA COME BACK ON THIS ILAND SOM DAY SOO

THE SPARKLE GONNA COME BACK ON THIS ISLAND SOMEDAY SOON

TRADE REPRESENTIGD JUIDER WARANTS THAT THE U S WONT BACKCOFF ITS PUSH FOR TRADE BARIOR REDUCTIONS

TRADE REPRESENTATIVE YEUTTER WARNS THAT THE U S WONT BACK OFF ITS PUSH FOR TRADE BARRIER REDUCTIONS

TREASURY SECRETARY BAGER AT ROHIE WOS IN AUGGRAL PRESSED FOUR ARISE INTHE VALUE OF KOREAS CURRENCY

TREASURY SECRETARY BAKER AT ROH TAE WOOS INAUGURAL PRESSED FOR A RISE IN THE VALUE OF KOREAS CURRENCY

Andrew Mass, End-to-end neural network speech recognition, 2022

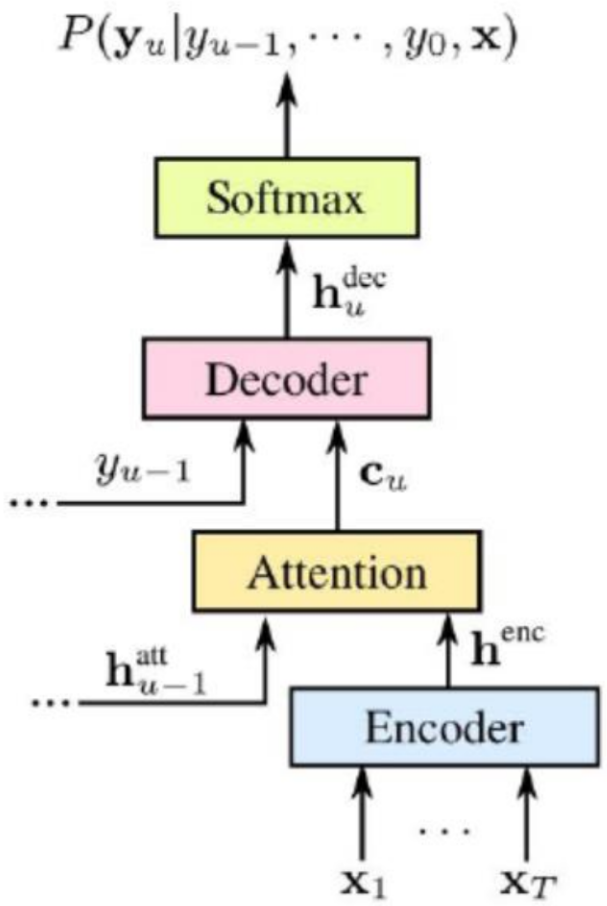
Attention-based End-to-End ASR

- **LAS method: Listen, Attend and Spell**
- **Listen: Encoder**

  Transforms input speech into higher-level representation
- **Attend: Attention**

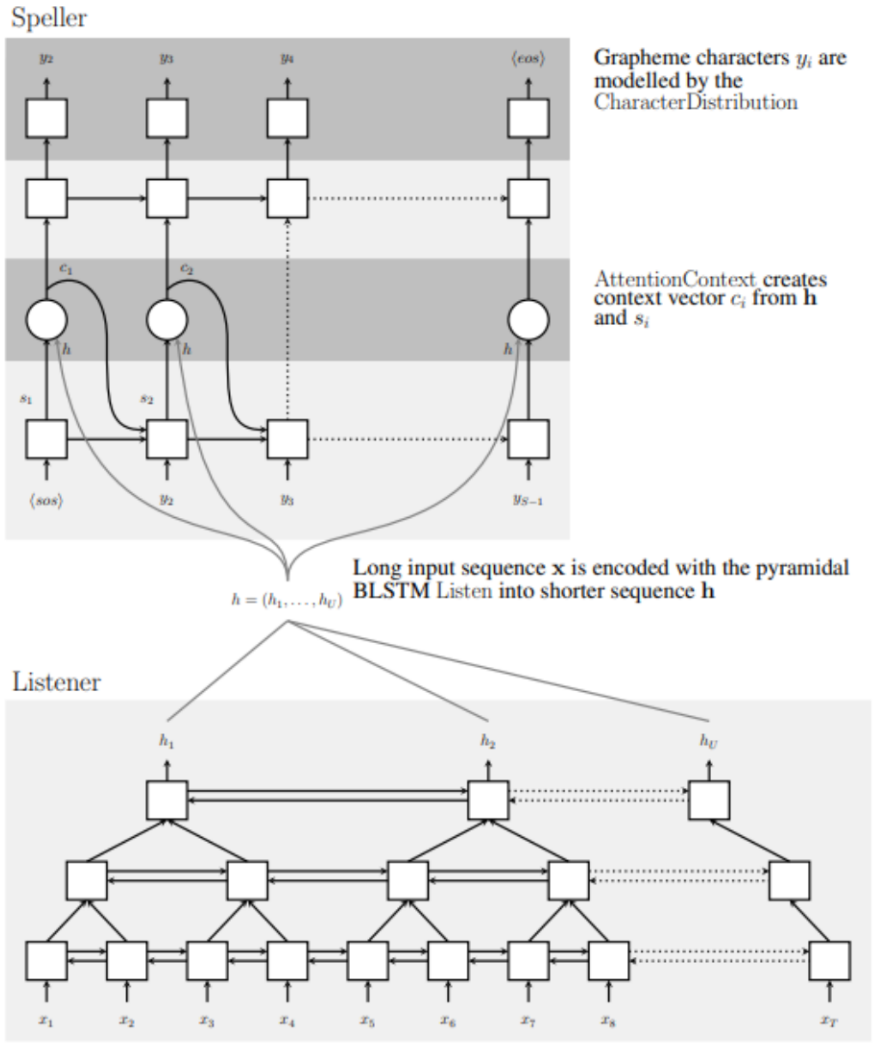
  Identifies encoded frames that are relevant to producing current output
- **Spell: Decoder**

  Operates autoregressively by predicting each output token as a function of the previous predictions



LAS model: Listen, Attention and Spell

- The listener is a pyramidal BLSTM encoding our input sequence x into high level features h.
- The speller is an attention-based decoder generating the y characters from h.



LAS model: Performance

Model	Clean WER	Noisy WER
CLDNN-HMM [22]	8.0	8.9
LAS	14.1	16.5
LAS + LM Rescoring	10.3	12.0

2000 hours

[Chan, et al., ICASSP'16]

Exp-ID	Model	VS/D	1st pass Model Size
E8	Proposed	**5.6/4.1**	**0.4 GB**
E9	Conventional LFR system	6.7/5.0	0.1 GB (AM) + 2.2 GB (PM) + 4.9 GB (LM) = 7.2GB

12500 hours

[Chiu, et al., ICASSP, 2018]

It works well when trained with a large dataset.

Streaming

- **Most commercial setups need the ASR systems to be streaming with low latency: ASR system produces the recognition results at the same time as the user is speaking.**

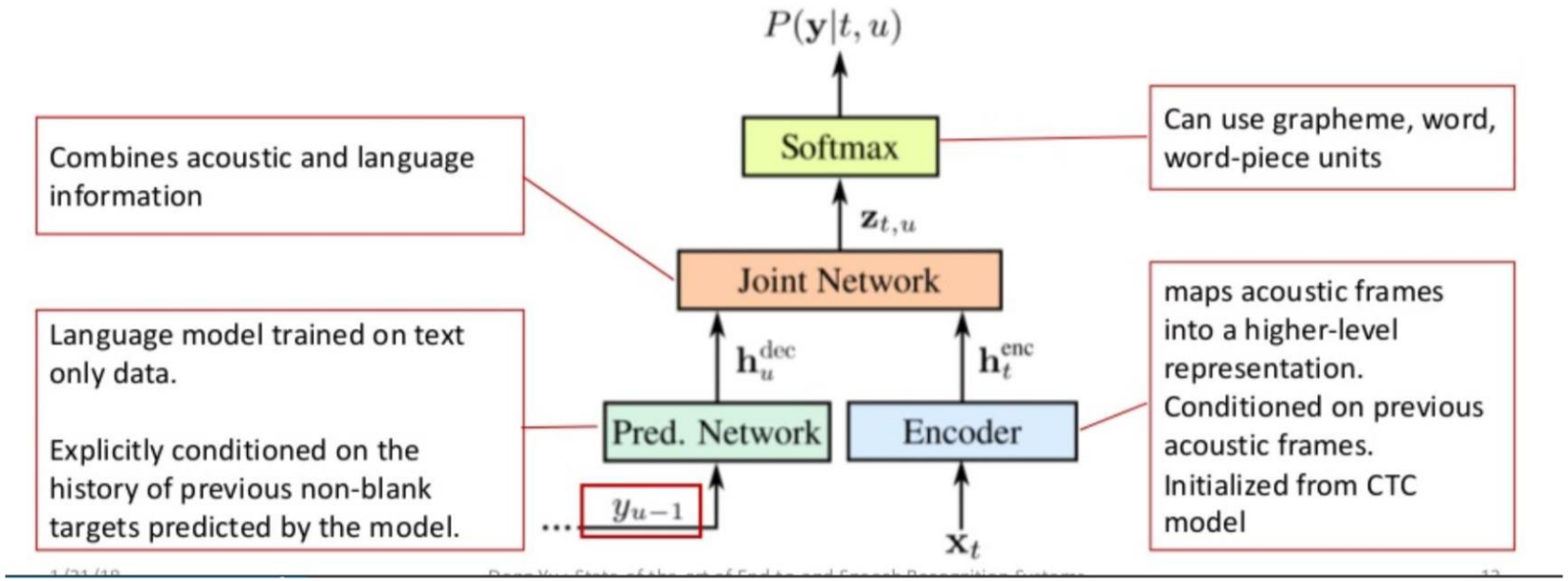
- **Full attention in AED may not be ideal to ASR because the decoder does not have access to future signals.**
  - Streaming AED (MOCHA, MILK etc.): apply attention on chunks of input speech.
  - Not a natural design for streaming.

- **RNN-T provides a natural way for streaming ASR and becomes the most popular E2E model.**

Chiu and Raffel, “Monotonic chunkwise attention,” in Proc. ICLR, 2018.
Arivazhagan et al., “Monotonic infinite lookback attention for simultaneous machine translation,” in Proc. ACL, 2019.

RNN Transducer (RNN-T)

- A streaming, all-neural, sequence-to-sequence architecture
- Jointly learns acoustic and language model components



Source: https://www.slideshare.net/BillLiu31/state-of-art-e2e-speech-recoginition-system-by-dong-yu

RNN Transducer (RNN-T)

- Properties of RNN-T

  - Do not need to process the entire input sequence to produce an output.
  - Continuously processes input samples and streams output symbols, good for speech dictation.
  - Outputs characters one-by-one, as you speak, with white spaces where appropriate.
  - Feeds predicted symbols to predict the next symbols. (Language model integrated)

E2E Models

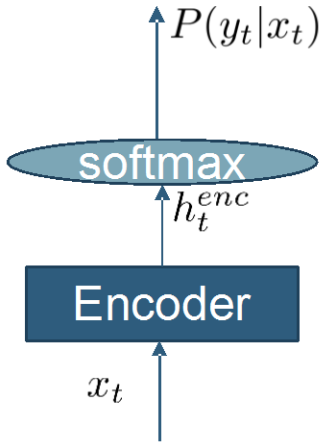
or Transformer Transducer

	CTC	AED	RNN-T
Independence assumption	Yes	No	No
Attention mechanism	No	Yes	No
Streaming	Natural	Additional work needed	Natural
Ideal operation scenario	Streaming	Offline	Streaming
Long form capability	Good	Weak	Good

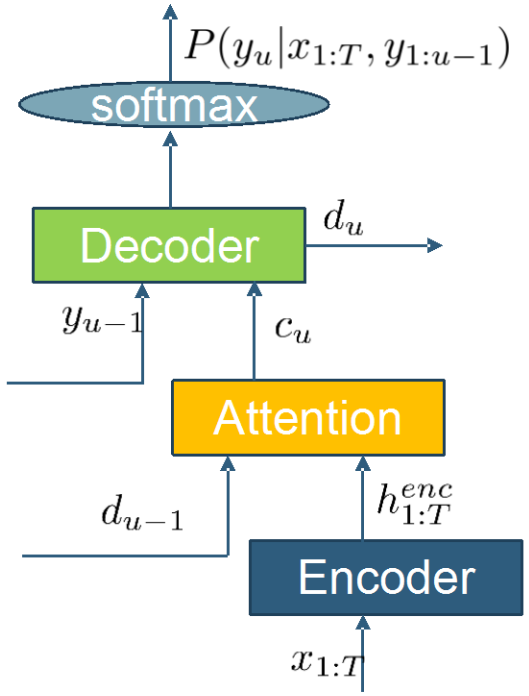
RNN-T is the most popular E2E model in industry which requires streaming ASR.

Sainath et al. “A streaming on-device end-to-end model surpassing server-side conventional model quality and latency” in Proc. ICASSP, 2020
Li et al., “Developing RNN-T Models Surpassing High-Performance Hybrid Models with Customization Capability,” *in* Proc. Interspeech, 2020.

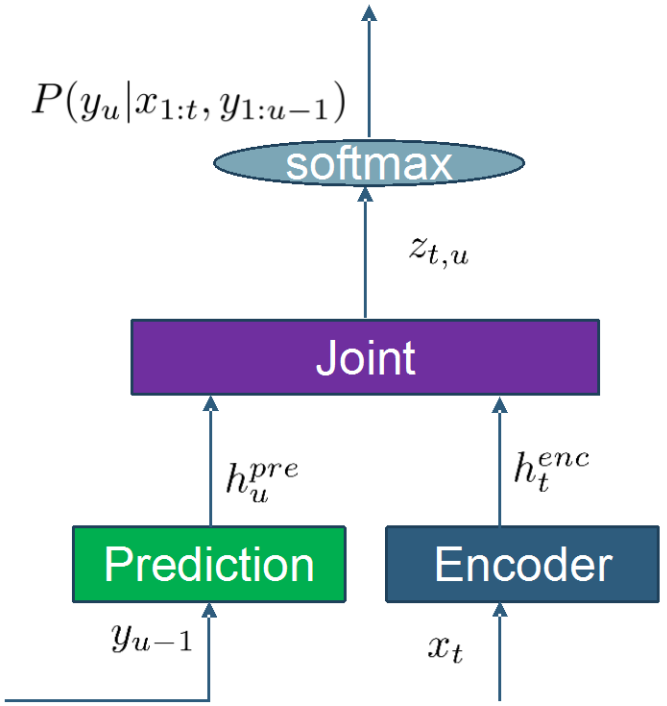
Encoder is the Most Important Component



Connectionist Temporal Classification (CTC)

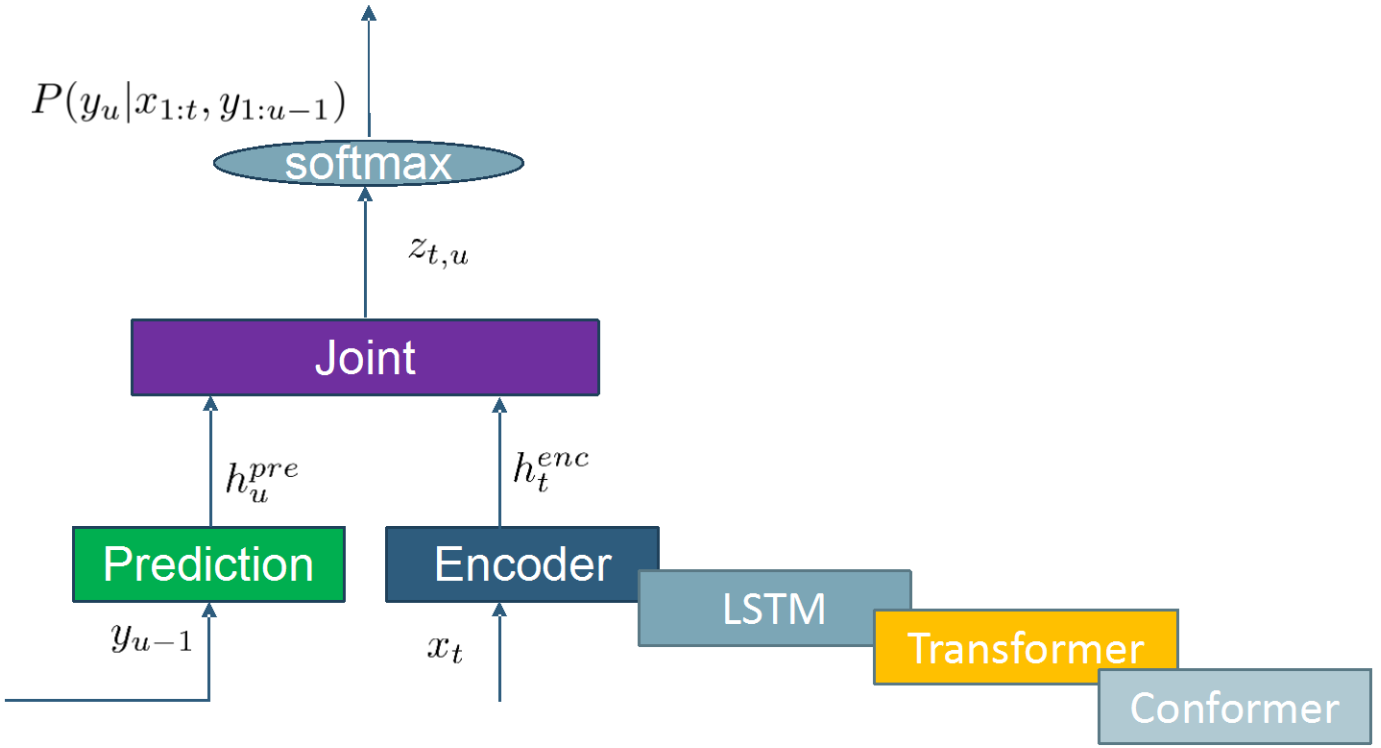


Attention-based encoder decoder (AED)



RNN-Transducer (RNN-T) or Transformer Transducer

Encoder for RNN-T



Transformer

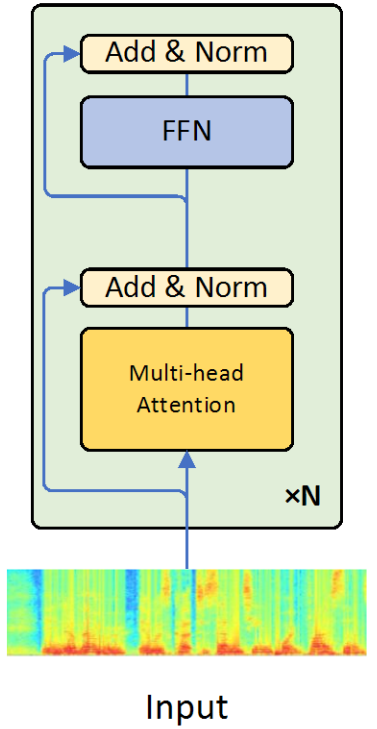
- **Self-attention: computes the attention distribution over the input speech sequence**

$$\alpha_{t,\tau} = \frac{\exp(\beta(\mathbf{W}_q\mathbf{x}_t)^T(\mathbf{W}_k\mathbf{x}_\tau))}{\sum_{\tau'}\exp(\beta(\mathbf{W}_q\mathbf{x}_t)^T(\mathbf{W}_k\mathbf{x}_{\tau'}))}$$

- **Attention weights are used to combine the value vectors to generate the layer output**

$$\mathbf{z}_t = \sum_\tau \alpha_{t\tau}\mathbf{W}_v\mathbf{x}_\tau = \sum_\tau \alpha_{t\tau}\mathbf{v}_\tau$$

- **Multi-head self-attention: applies multiple parallel self-attentions on the input sequence**



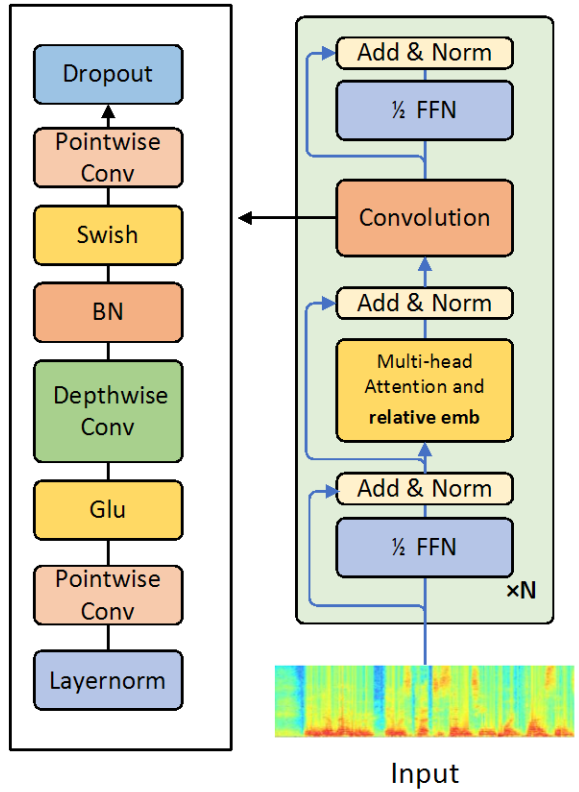
Vaswani et al. “Attention is all you need” NIPS 2017

Zhang et al., “Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss.” in Proc. ICASSP 2020.

A good tutorial: https://towardsdatascience.com/transformers-explained-visually-part-1-overview-of-functionality-95a6dd460452

Conformer

- **Transformer: good at capturing global context, but less effective in extracting local patterns**
- **Convolutional neural network (CNN): works on local information**
- **Conformer: combines Transformer with CNN**

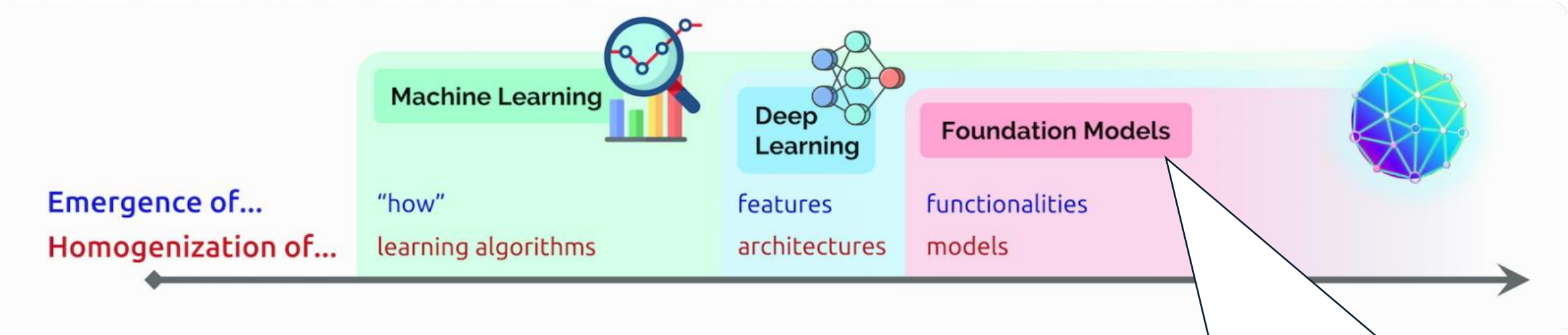


Gulati et al. "Conformer: Convolution-augmented Transformer for Speech Recognition," in Proc. Interspeech, 2020.

PART 4A. EMERGING ADVANCED SPEECH PROCESSING

- SPEECH FOUNDATION MODELS

Paradigm Shift



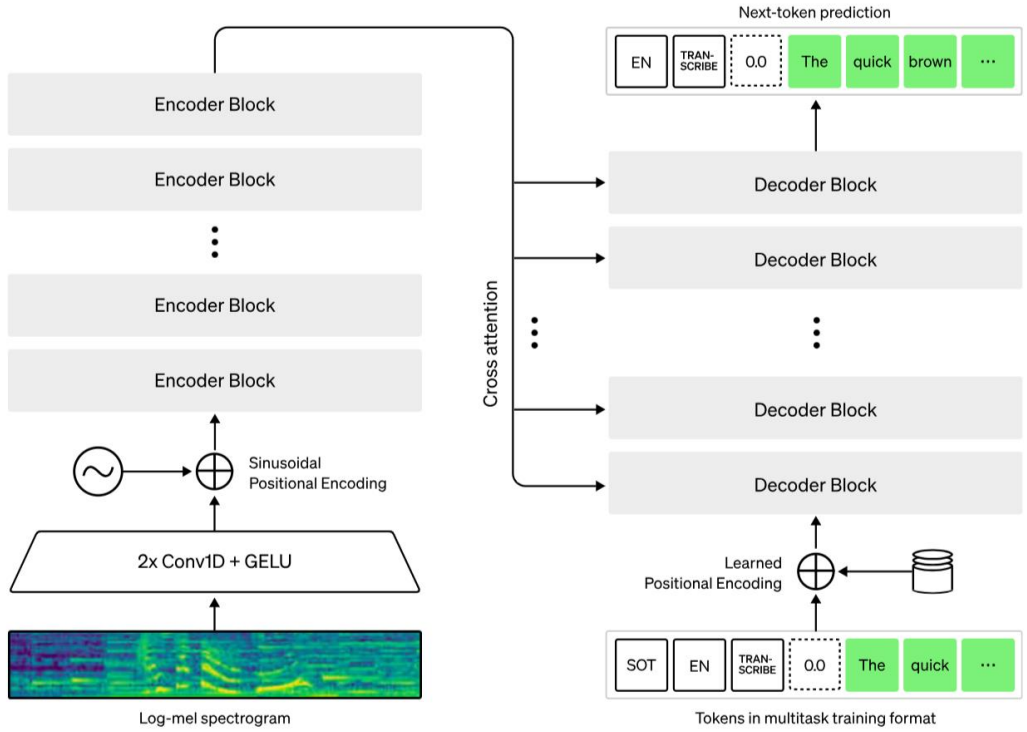
- serves as a foundational layer upon which specialized applications are built.
- involves extensive training data
- being adapted to wide range of downstream tasks

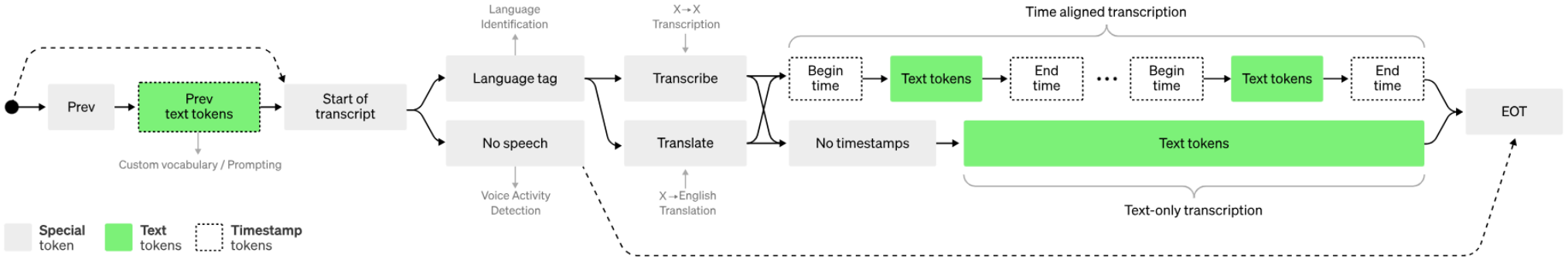
Types of Learning

- **Supervised Learning:**
  - Need human labeled data. Very expensive to build databases.
- **Alternative solutions:**
  - Unsupervised Learning: No human labels needed. Easy to build the databases. Discover patterns from unlabeled data
  - Semi-supervised learning: Use a small amount of labelled data.
  - Self-supervised learning: Use information from input data as the label to learn representations.

Whisper Model

- Trained on 680k hours of multilingual and multitask supervised data
- Three tasks: speech recognition, speech translation, and language recognition
- #model parameters: 39M, 74M, 244M, 769M, 1550M





Source: https://openai.com/research/whisper

Popular Simultaneous Speech Translation (ST) Methods

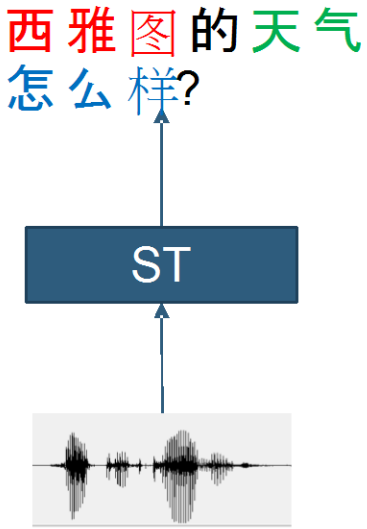
- **Re-Translation: re-translate partial ASR results from beginning**
  - Cost is very high because machine translation (MT) needs to be called multiple times
  - Stability is an issue because the outputs of different MT calls are independent

- **Wait-K: start to translate ASR results after waiting for K words.**
  - The read-write operation is interleaving, not flexible
  - K is pre-determined

- **AED models for E2E ST**
  - Streaming AED is still a challenge

Can We Build a Simultaneous Direct ST System?

- Treating ST as an ASR problem – we already have the success in streaming E2E ASR.
- We directly use streaming Transformer Transducer to build streaming ST.



Xue et al. "Large-Scale Streaming End-to-End Speech Translation with Neural Transducers." in Proc. Interspeech, 2022.

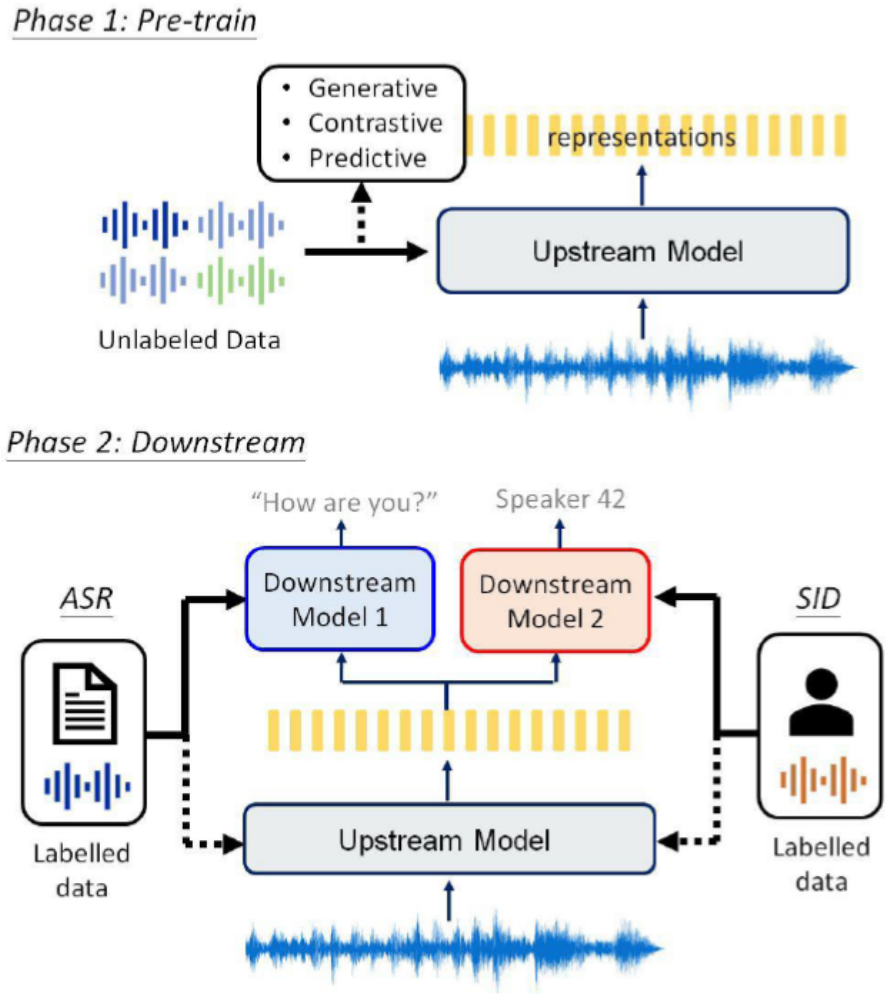
Streaming Multilingual Speech Model (SM^2)

- **Multilingual data is pooled together to train a streaming model to perform both ST and ASR functions.**
- **ST training is totally weakly supervised without using any human labeled parallel corpus.**
- **The model is very small, running on devices.**

Xue et al. "A Weakly-Supervised Streaming Multilingual Speech Model with Truly Zero-Shot Capability." *arXiv preprint,* 2022.

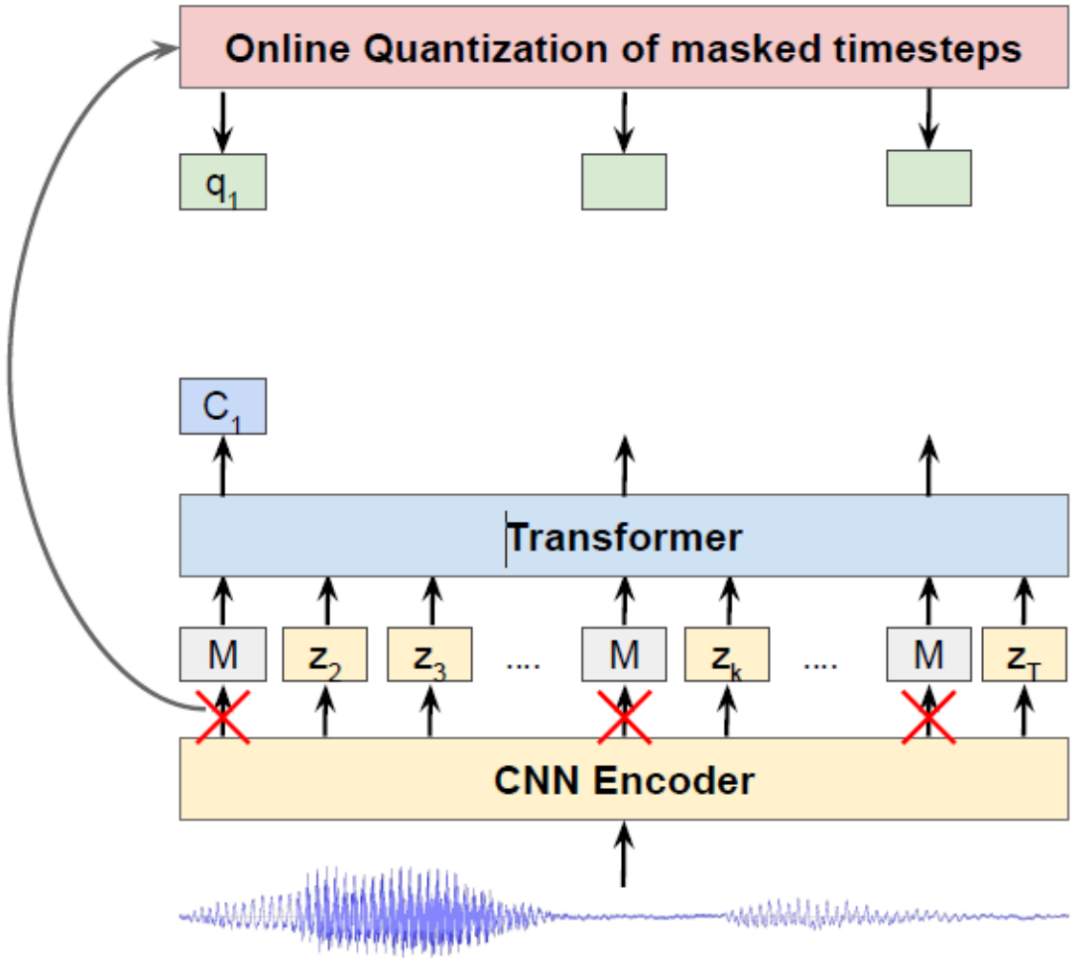
Wav2Vec2.0: Self Supervised Learning

- **Two stages:**
  - 1. Use SSL to pre-train an upstream model
  - 2. Downstream task uses the learned representation from a pre-trained model (frozen) or fine-tune the pre-trained model using supervised data



Wav2Vec 2.0

- A method to train speech representation.
- Quantization is used to create targets in self-supervised learning.
- Training is to maximize the similarity between the learned contextual representation and the quantized input features.

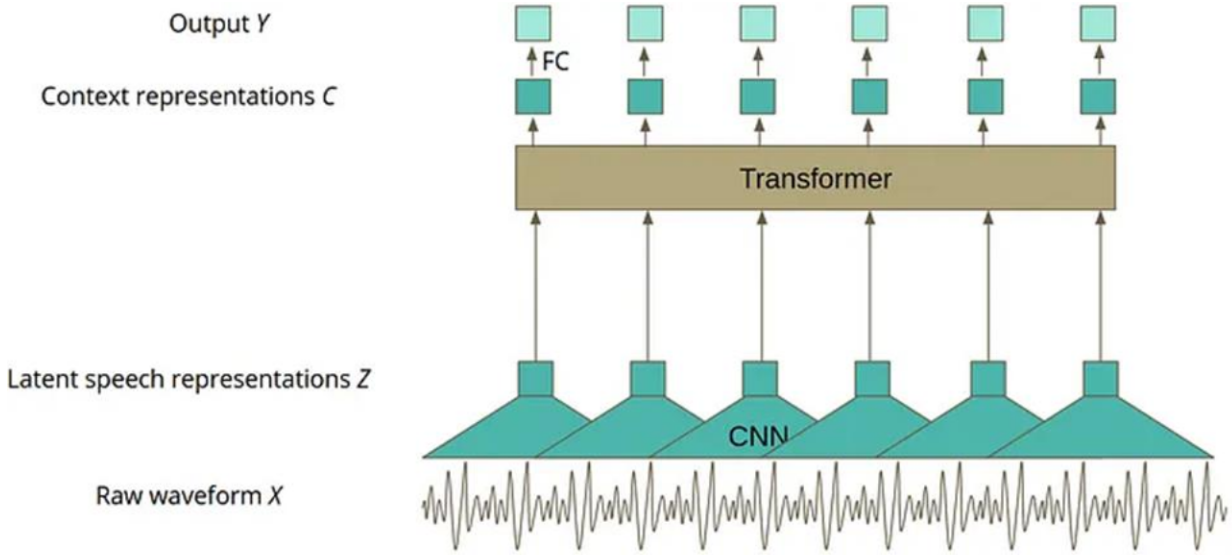


A. Baevski, H. Zhou, A. Mohamed and M. Auli, wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations (2020), CoRR

Wav2Vec 2.0

- Good performance on low resource languages
- Achieved 8.2%WER when fine-tuned with 10 minutes data. This is normally achieved with thousands hours speech data.

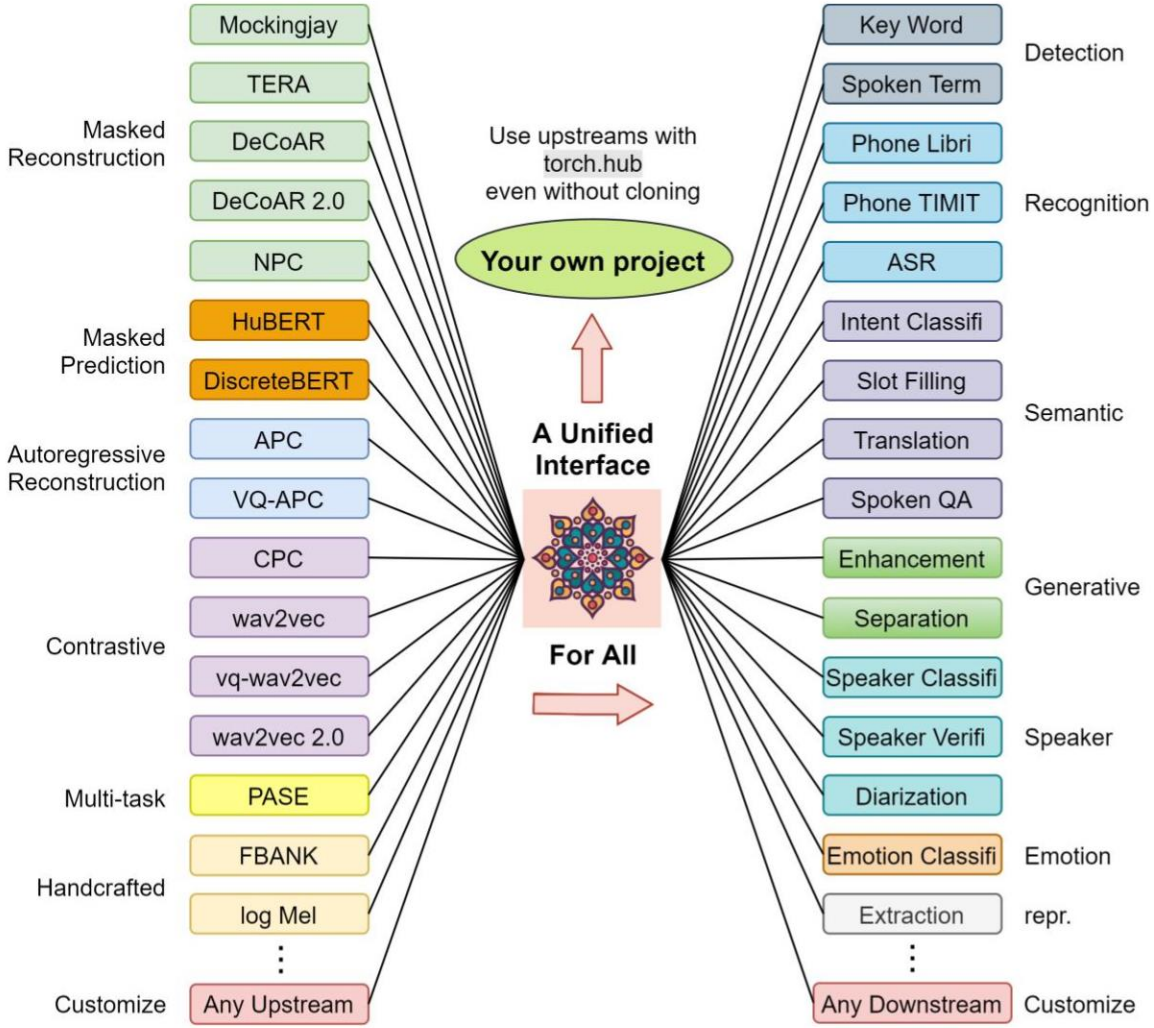
Results of Wav2Vec 2.0



Fine-tuned Wav2Vec 2.0 model architecture

Sample Representations and Applications

- Many speech representations are created by self-supervised learning.
- They can be used in many downstream applications.



Source: https://github.com/s3prl/s3prl

Toolkits

- **FAIRSEQ:**
  - wav2vec, vq-wav2vec, wav2vec 2.0, data2vec
  - HuBERT, wav2vec U, wav2vec U 2.0, GSLM, pGSLM…
- **S3PREL:**
  - The most comprehensive library for pre-trained models
- **ESPNet:**
  - Pre-train HuBERT Task
- **SpeechBrain:**
  - A Fine-tuned Wav2vec 2.0/HuBERT Benchmark

Summary about Self Supervised Learning

- Learn from unlabelled data without human annotation, which is an low cost solution.
- Use contextual information to improve the accuracy.
- Self learned representation and fine-tuning together can improve accuracy for low resource speech recognition.
- It showed good performance on many downstream tasks like speaker recognition, emotion recognition, speech separation, etc.

PART 4B. EMERGING ADVANCED SPEECH PROCESSING

- LARGE LANGUAGE MODELS WITH SPEECH RECOGNITION CAPABILITIES

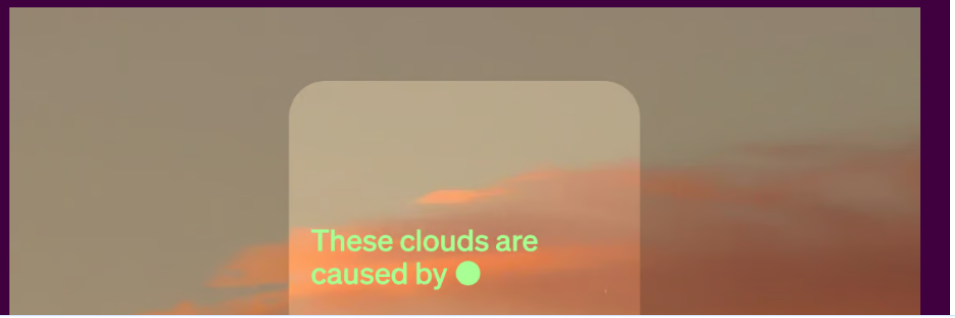
ChatGPT: LLM with Speech Capabilities

OpenAI Research API ChatGPT Safety Company Search Log in ↗ Try ChatGPT ↗

Blog

ChatGPT can now see, hear, and speak

We are beginning to roll out new voice and image capabilities in ChatGPT. They offer a new, more intuitive type of interface by allowing you to have a voice conversation or show ChatGPT what you're talking about.



openai.com/blog/chatgpt-can-now-see-hear-and-speak

OpenAI

To get started with voice, head to Settings → New Features on the mobile app and opt into voice conversations. Then, tap the headphone button located in the top-right corner of the home screen and choose your preferred voice out of five different voices.

The new voice capability is powered by a new text-to-speech model, capable of generating human-like audio from just text and a few seconds of sample speech. We collaborated with professional voice actors to create each of the voices. We also use Whisper, our open-source speech recognition system, to transcribe your spoken words into text.

1. Speech LLM

- **How it works**
  - By directly prepending a sequence of audial embeddings to the text token embeddings, the LLM can be converted to an automatic speech recognition (ASR) system, and be used in the exact same manner as its textual counterpart
- **What is the performance**
  - Incorporating a conformer encoder into the open sourced LLaMA-7B allows it to outperform monolingual baselines by 18% and perform multilingual speech recognition despite LLaMA being trained overwhelmingly on English text.

Prompting Large Language Models with Speech Recognition Abilities, Yassir Fathullah et al.

1. Speech LLM

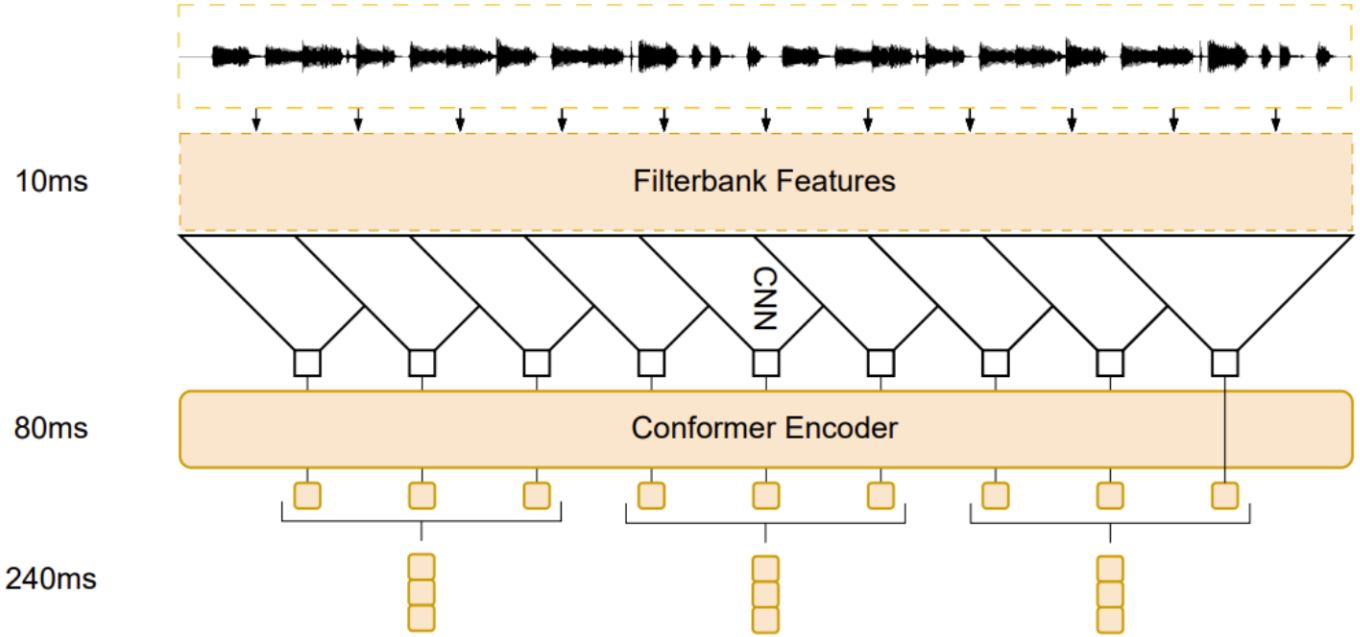


Figure 1: Audio encoder architecture. The initial conformer is trained on a CTC loss. Thereafter the outputs are stacked and projected to the dimension of the LLM to ensure compatibility. This figure showcases a stacking factor of 3 resulting in 240ms embeddings.

Prompting Large Language Models with Speech Recognition Abilities, Yassir Fathullah et al.

1. Speech LLM

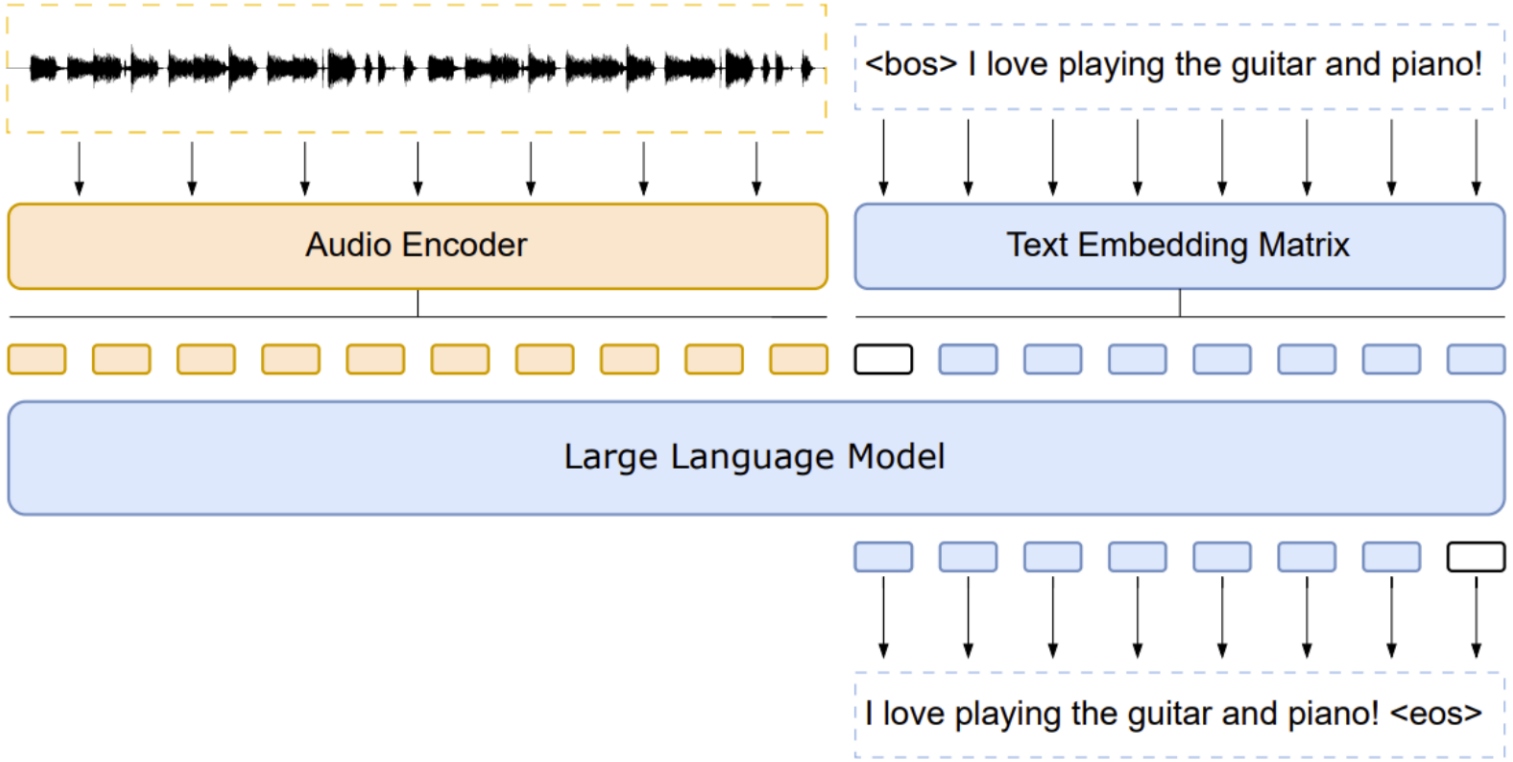


Figure 2: Model architecture. The embedding sequence generated from the audio encoder is directly prepended to the text embeddings sequence. This is directly fed into the decoder-only LLM, tasked with predicting the next token. The LLM can be frozen, adapted with parameter efficient approaches such as LoRA or fully finetuned. This work will investigate the former two.

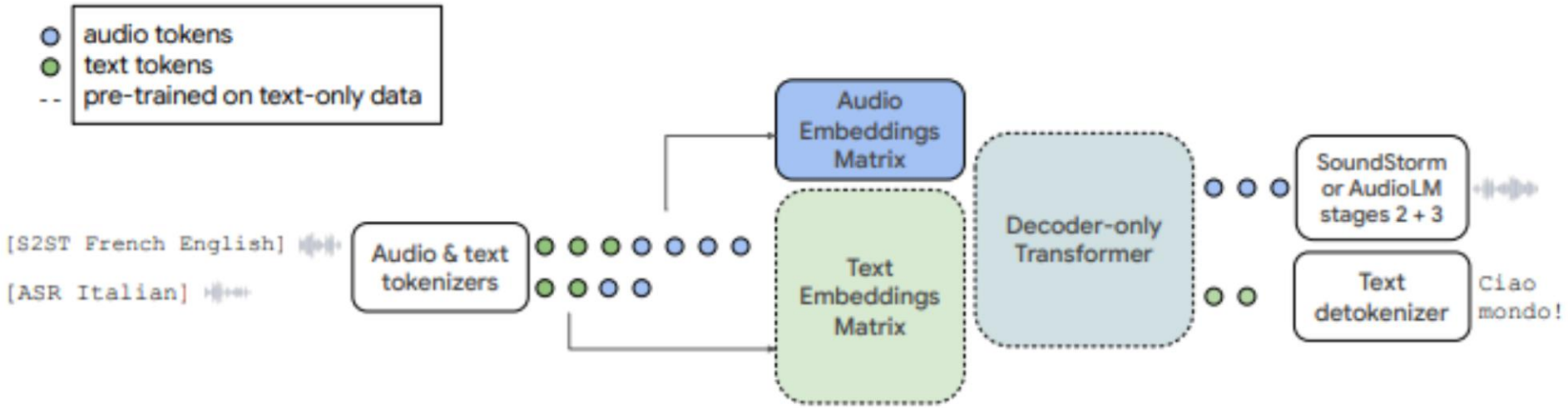
Prompting Large Language Models with Speech Recognition Abilities, Yassir Fathullah et al.

2. AudioPaLM

- **A Large Language Model That Can Speak and Listen**
  - AudioPaLM fuses text-based and speech-based language models, into a unified multimodal architecture
  - Can process and generate text and speech with applications including speech recognition and speech-to-speech translation
- **Performance**
  - The resulting model significantly outperforms existing systems for speech translation tasks and has the ability to perform zero-shot speech-to-text translation for many languages for which input/target language combinations were not seen in training

AudioPaLM: A Large Language Model That Can Speak and Listen, Paul K. Rubenstein et al

2. AudioPaLM



PART 5. ASR TRAINING ISSUES

How to Train an ASR system

- **Resources**
  - Speech with text transcription, big enough to cover the language, Should have word transcriptions
  - Pronunciation dictionary
  - Big text corpus
- **Models to train**
  - Acoustic models
  - Language model

Combine external LMs with E2E models via shallow fusion for domain adaptation

Guidelines for Acoustic Modeling Units

- Very phonetic languages (e.g., Spanish, German)
  - Letter-based or byte pair encoding (BPE) based acoustic modeling is effective.
- Reasonably phonetic languages (e.g., English)
  - BPE-based acoustic model is preferred.
  - Letter-based acoustic modeling is also viable if large training datasets are available.
- Non-phonetic or least phonetic languages (e.g., Chinese)
  - It is recommended to use a pronunciation dictionary to map words to modeling units.

Further reading on Acoustic Modeling

- Fine-Tune Whisper For Multilingual ASR with Transformers
  - https://huggingface.co/blog/fine-tune-whisper
- Espnet: End-to-end speech processing
  - https://github.com/espnet/espnet
- Kaldi: HMM-DNN acoustic modeling
  - https://kaldi-asr.org/
  - https://kaldi-asr.org/doc/tutorial.html
  - https://kaldi-asr.org/doc/kaldi_for_dummies.html
  - https://github.com/kaldi-asr/kaldi/tree/master/egs
- HMM-based monophone and context-dependent triphone:
  - https://jonathan-hui.medium.com/speech-recognition-asr-model-training-90ed50d93615

Further Reading on N-gram Language Models

- Overview
  - https://web.eecs.umich.edu/~wangluxy/courses/eecs498_wn2021/slides_eecs498_wn21/lm.pdf
- LM training with SRILM
  - https://cmusphinx.github.io/wiki/tutoriallmadvanced/
- LM linear interpolation/Building large n-gram LMs with SRILM
  - https://joshua.apache.org/6.0/large-lms.html
- Morph n-gram model
  - http://research.spa.aalto.fi/speech/s895150/ex3.html

Training Process

- **Dataset**
  - Modelling system needs thousand hours of speech
  - Dataset is divided into training, development and testing sets.
- **Training & Testing**
  - Train the system
  - Test on development set
  - Tune the system and repeat the process
  - At the end, test on the testing set.

Data Quality and Quantity

- **Diverse Data:**
  - Ensure the dataset covers various accents, dialects, speaking rates, and noise levels.
- **Data Augmentation:**
  - Techniques like time-stretching, pitch-shifting, and adding background noise can artificially increase and diversify the dataset.
- **Transcription Accuracy:**
  - Labels should be as accurate as possible since errors can degrade model performance. Expensive to collect and transcribe speech data.
- **Data Quantity**
  - Morden systems are normally trained with thousands of hours speech data. Some recent ones using 60000 hours or more.

Accuracy Measure

- **Usually, ASR performance is judged by the word error rate**

ErrorRate = 100*(Subs + Ins + Dels) / Nwords

```
REF:  I  WANT  TO   GO  HOME  ***
REC:  *  WANT  TWO  GO  HOME  NOW
SC:   D   C     S   C    C    I
100*(1S+1I+1D)/5 = 60%
```

Character Error Rate (CER) or Syllable Error Rate (SER) may be used for some languages.

Accuracy Measure

- **Usually, ASR performance is measured by Word Error Rate**
  - This assumes that all errors are equal
  - Also, a bit of a mismatch between the optimization criterion and error measurement

- **Task-specific measures are sometimes used**
  - Task completion
  - Concept error rate

Speaker Dependent System

- **Speaker dependent/independent**
  - Current ASR systems are normally developed for any user.
  - But ASR system can be adapted to a particular speaker.
- **Speaker dependent system:**
  - System is built for a particular speaker only.
  - Relatively easy to achieve higher accuracy.
- **Speaker independent system:**
  - System works for any new speaker.
  - Difficult to achieve high accuracy

Other Considerations

- **Adaptability and Fine-tuning**:
  - **Domain Adaptation**: If the ASR system is intended for a specific domain (e.g., medical or legal), additional fine-tuning on domain-specific data can be beneficial.
  - **Online Learning**: For continually evolving applications, the model might need to adapt to new data over time.
- **Noise and Environment**:
  - **Robustness**: The model should be robust to different noise levels and types, especially for applications like voice assistants.
  - **Acoustic Models**: Different environments (e.g., indoors vs. outdoors) can affect acoustics, so it's essential to consider this during training.

PART 6. DEPLOYMENT ISSUES

Considerations in ASR Deployment

- **Free text or limited text**
  - Free text: large vocabulary free text recognition
    Examples: dialog system, dictation, etc
  - Voice commands: Limited number of commands. Examples: ASR for embedded system.
- **Cloud or embedded**
  - Cloud: Speech recognition is on cloud. Normally the recognition model is huge. Most of free text recognition engines are on cloud.
  - Embedded system: Small system data size.

Considerations in ASR Deployment

- **Distance**
  - Near field: Smartphone, PC, etc
  - Far field: Microphone is far away from speaker.
    Example: Echo, etc
- **Channel/sampling rate**
  - Digital system: Digital system like smartphone, sampling rate is 16KHz.
  - Telephony: The sampling rate is 8KHz for tranditional telephony systems.

Considerations in ASR Deployment

- **Multi-thread processing**
  - Real-time factor: Time needed to process 1 second speech. Real-time factor 0.1 means 1 CPU can support 10 users.
  - Memory: System data can be shared among running instances. E.g. model, lexicon, etc. Working data need additional memory for each user. (E.g. intermediate data generated for the user in recognition process)
- **Response time**
  - Time used for data transfer: data sent to server, result sent to client.
  - Delay in speech recognition engine for real time processing.

Examples of Speech Application

- Telephony system
- Medical records
- Court hearing transcription
- Speech analytics for call center
- Voice assistant/Smart speaker
- Voice control devices
- In-car application
- Air flight traffic control

PART 7. RESOURCES & PROGRAMMING

Speech Recognition Toolkit

- **Kaldi**
  - Kaldi is a toolkit for speech recognition,
  - For use by speech recognition researchers and professionals.
  - Support many of the advanced features.
- **HTK**
  - A proprietary software toolkit for handling HMMs
- **EPSnet**
  - End-to-end speech processing toolkit
- **Wenet**
  - An open-source speech recognizer developed for deployment
- **Whisper**
  - Open-source package by OpenAI. With pretrained model (Speech recognition and translation)

Integrated Speech Recognition

- Windows system
- MacOS
- iPhone
- Android system

Cloud Services

- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text

Open-source Python Speech Recognition

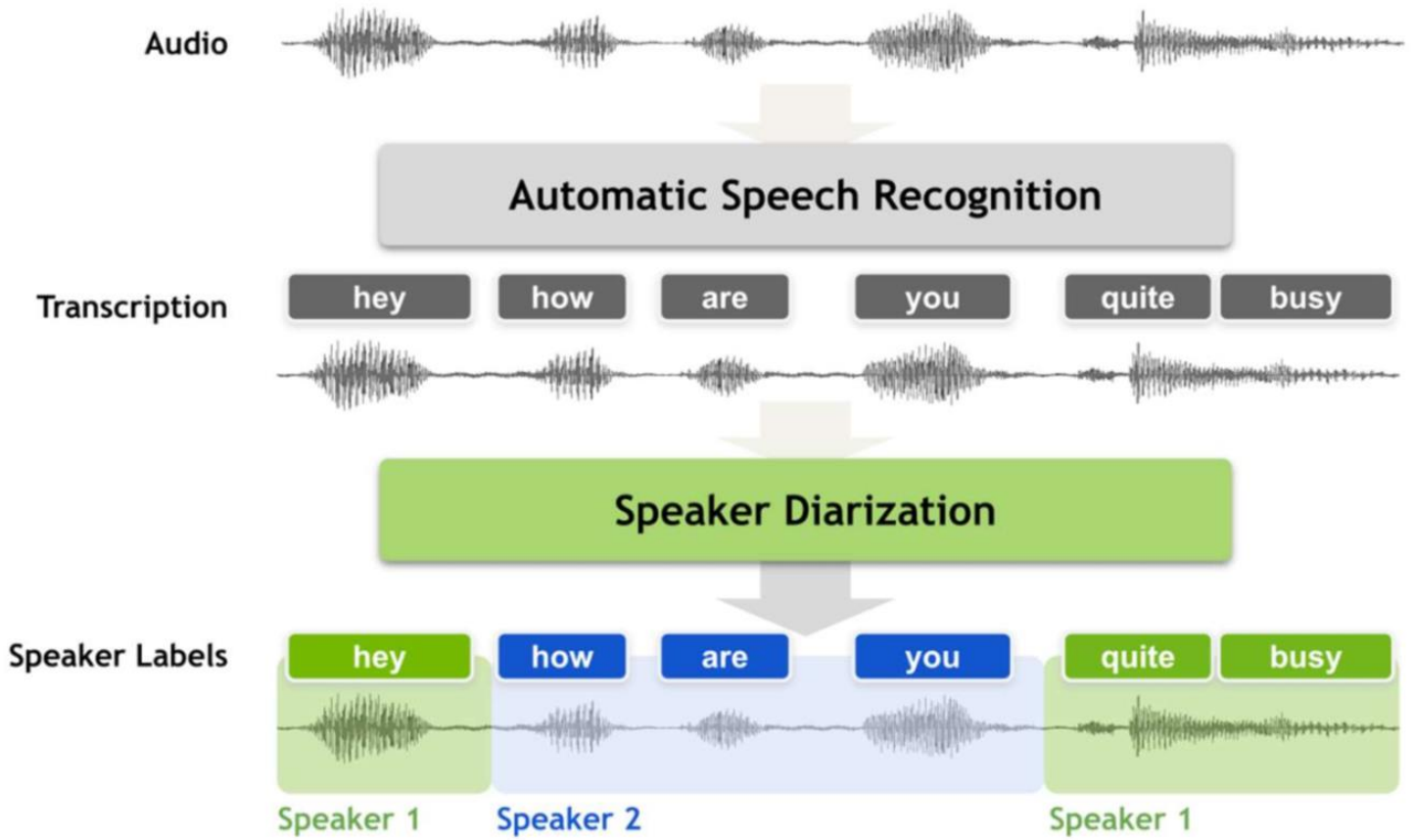
- Installation: pip install SpeechRecognition
- Library for performing speech recognition
- Supported engines:
  - CMU Sphinx (works offline)
  - Google Speech Recognition
  - Google Cloud Speech API
  - Wit.ai
  - Microsoft Bing Voice Recognition
  - Houndify API
  - IBM Speech to Text
  - Snowboy Hotword Detection (works offline)
  - OpenAI Whisper (works offline)

Open-source Python Speech Recognition

- Recognize speech input from the microphone
- Transcribe an audio file
- Save audio data to an audio file
- Calibrate the recognizer energy threshold for ambient noise levels
- Listening to a microphone in the background
- Website: https://pypi.org/project/SpeechRecognition/

Topic 3: Speaker Diarization

Speaker Diarization



Reference: https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/main/asr/speaker_diarization/intro.html

Speech Diarization

- This is the task of determining "who spoke when" in an audio recording.
- It involves both detecting when speech occurs and attributing the speech to specific speakers.
- The goal is to segment and cluster the audio such that all segments belonging to a particular speaker are grouped together.

Importance & Applications

- **Multi-participant Broadcasts, Meetings, or Interviews:**
  - In scenarios with multiple speakers, diarization helps in understanding the structure of the conversation and the contribution of each participant.
- **Enhancing Transcription Services:**
  - By labeling speakers, transcriptions become more informative and readable, especially in contexts like legal depositions or medical consultations where speaker identity is crucial.
- **Improving Voice Assistants:**
  - In environments where multiple people might be speaking, diarization allows voice assistants to distinguish between the commands of the primary user and background conversations.

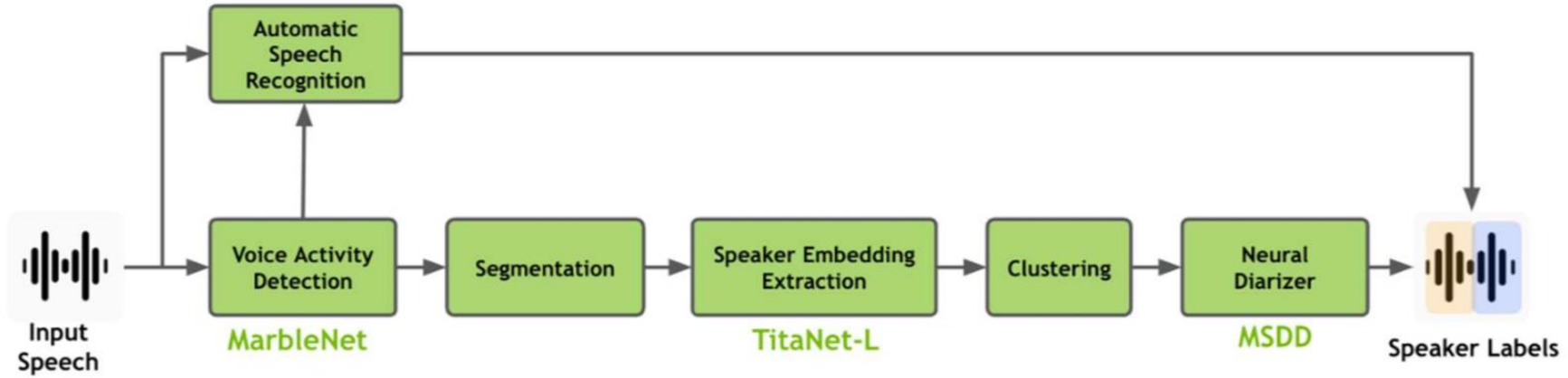
Key Challenges

- **Overlapping Speech:**
  - One of the most challenging aspects is when two or more speakers talk simultaneously. Traditional methods struggle in such scenarios, and advanced models are required.
- **Variability in Speech:**
  - A person's voice can vary due to emotions (e.g., anger, joy), health conditions (e.g., a cold), or environmental factors (e.g., background noise, room acoustics).
- **Short Speaker Turns:**
  - Rapid back-and-forth exchanges can make it challenging to get enough data to accurately identify and differentiate speakers.

Main Components

- **Voice Activity Detection (VAD):**
  - This is the first step, where non-speech segments (like silence or background noise) are filtered out, focusing the analysis only on speech segments.
- **Speaker Segmentation:**
  - The continuous speech is divided into smaller, homogeneous segments, ideally such that each segment contains speech from only one speaker.
- **Clustering:**
  - The homogeneous segments are then grouped together using clustering algorithms, ensuring that all segments from a single speaker fall into the same cluster.
- **Re-segmentation:**
  - An iterative process where segment boundaries are adjusted and refined to improve the accuracy of speaker clusters.

NeMo speaker diarization pipeline



NeMo speaker diarization system consists of the following modules:

- **Voice Activity Detector (VAD)**: A trainable model which detects the presence or absence of speech to generate timestamps for speech activity from the given audio recording.
- **Speaker Embedding Extractor**: A trainable model that extracts speaker embedding vectors containing voice characteristics from raw audio signal.
- **Clustering Module**: A non-trainable module that groups speaker embedding vectors into a number of clusters.
- **Neural Diarizer**: A trainable model that estimates speaker labels from the given features.

Reference: https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/main/asr/speaker_diarization/intro.html

Evaluation

- **Diarization Error Rate (DER):**
  - The sum of the percentage of three errors.
  - **Missed Speech (MISS):** The percentage of reference speaker speech that is not attributed to any speaker by the diarization system.
  - **False Alarm (FA):** The percentage of time that the diarization system identifies as speech but does not correspond to any reference speaker speech.
  - **Speaker Error (ERROR):** The percentage of reference speaker speech that is attributed to the wrong speaker by the diarization system.
- **A lower DER indicates better performance.**

NUS National University of Singapore | ISS

facebook.com/iss.nus

instagram.com/iss.nus

linkedin.com/company/iss_nus

youtube.com/@nus-iss

Thank you

www.iss.nus.edu.sg