Speech Technologies with Neural Networks

Jiatong Shi

jiatongs@cs.cmu.edu

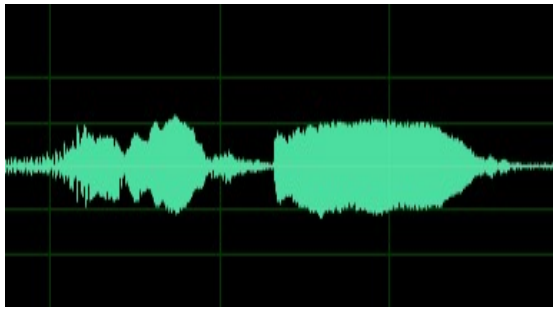
Content

INTRODUCE NEURAL NETWORKS TO SPEECH (ASR AS AN EXAMPLE)

END-TO-END SPEECH APPLICATIONS

Automatic speech recognition

- Target: to recognize the linguistic information in speech

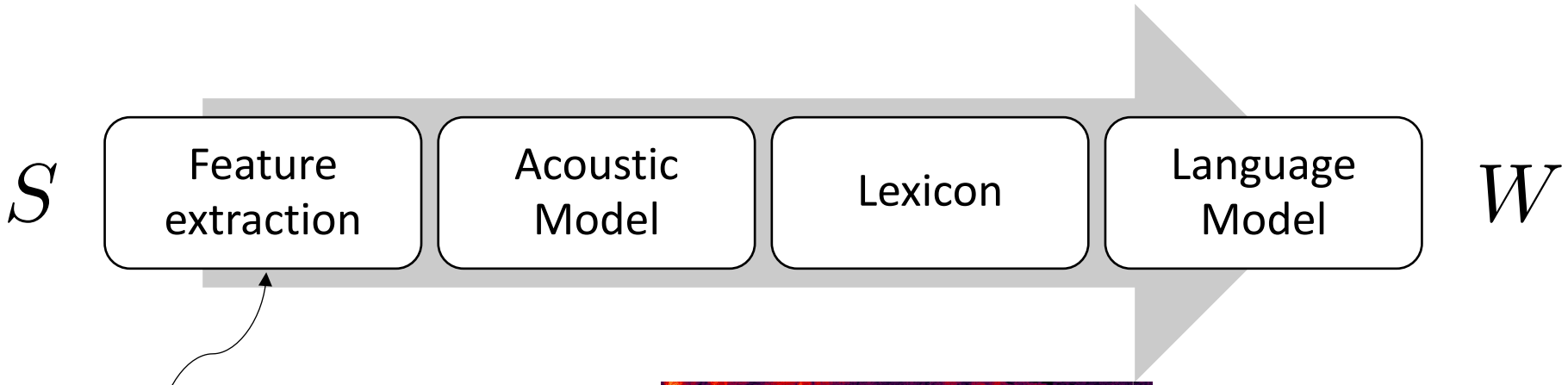


I'm not you

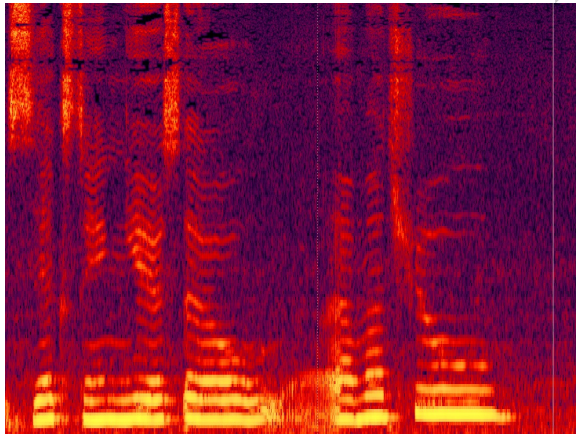
$$S = \{s_n \in \mathbb{Z} | n = 1, ..., N\}$$

$$W = \{w_l \in \mathcal{V} | l = 1, ..., L\}$$

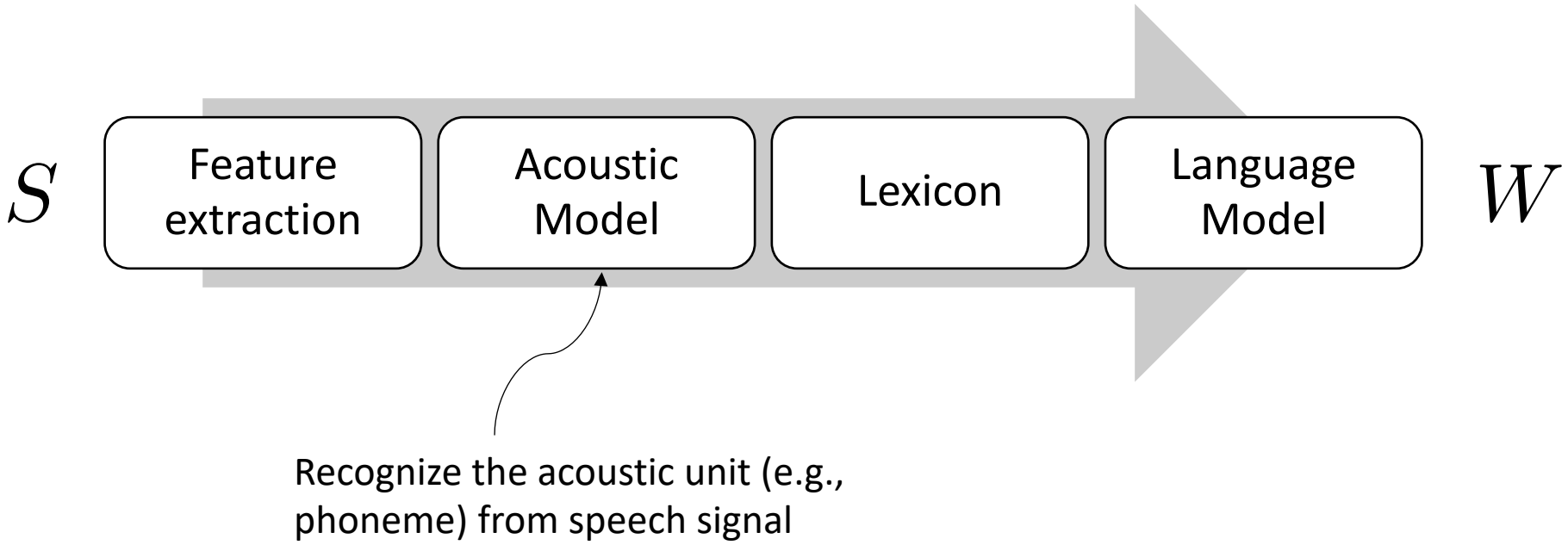
Conventional ASR



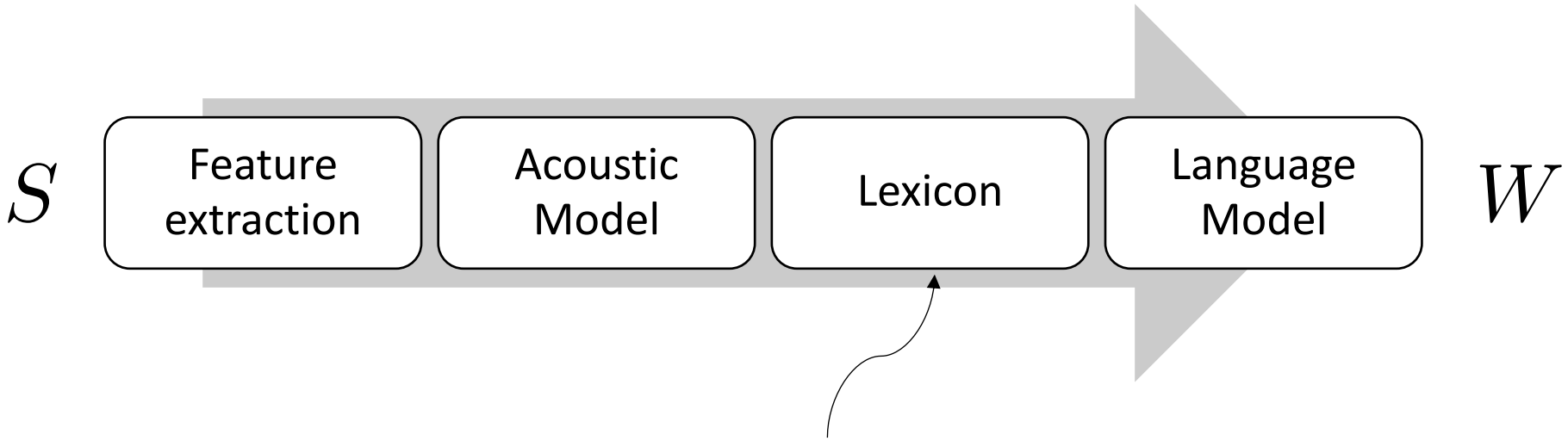
Raw signal S is a 1D long sequence, which is difficult to process, so normally we apply a psychoacoustic-inspired feature extraction.



Conventional ASR

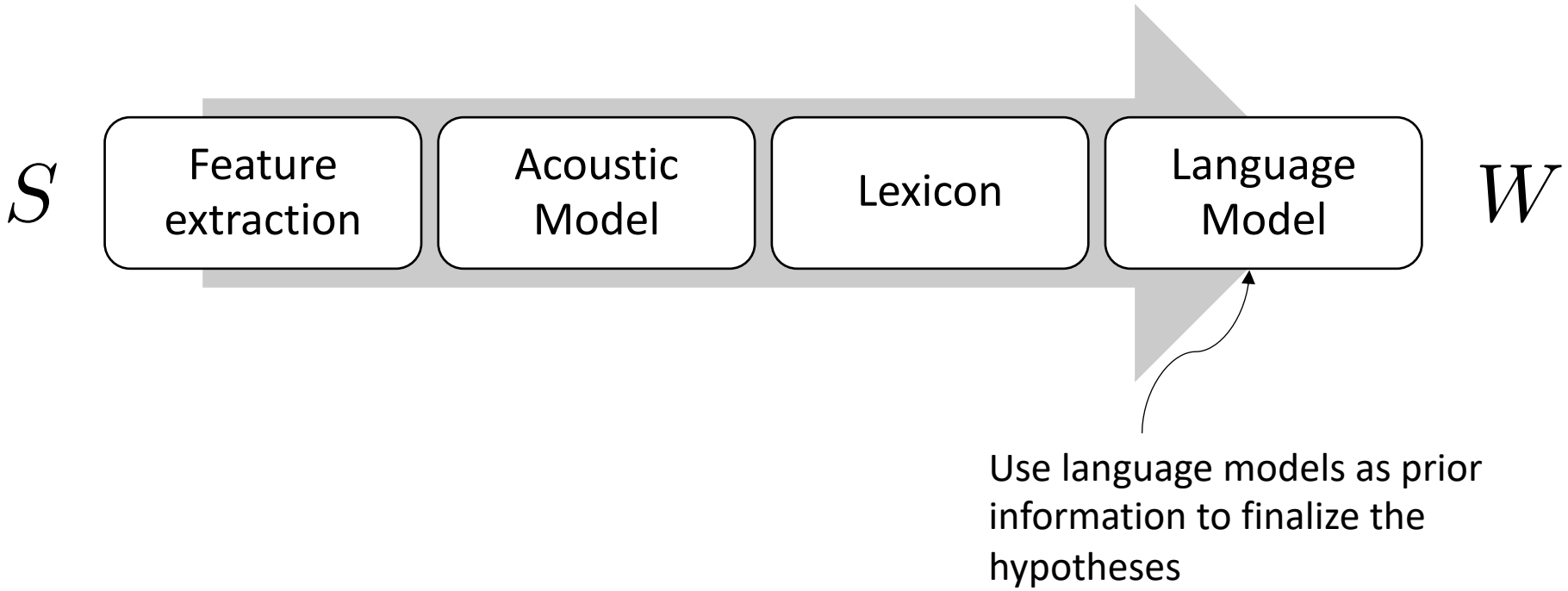


Conventional ASR



Convert acoustic units (e.g., phoneme) into words based on a dictionary (i.e., lexicon)

Conventional ASR



Mathematical formulation

- We note
  - Extracted speech features as $O = \{o_t | t = 1, ..., T\}$
  - Phoneme sequence as $Q = \{q_i \in \mathcal{Q} | i = 1, ..., I\}$

$$O = \mathrm{FeatureExtract}(S)$$

Mathematical formulation (Cont'd)

$$
\begin{aligned}
\mathrm{argmax}_W p(W|O) &= \mathrm{argmax}_W \sum_Q p(W, Q|O) \\
&= \mathrm{argmax}_W \sum_Q \frac{p(W, Q, O)}{p(O)} \\
&= \mathrm{argmax}_W \sum_Q \frac{p(O|W, Q)p(Q|W)p(W)}{p(O)} \\
&= \mathrm{argmax}_W \sum_Q p(O|W, Q)p(Q|W)p(W) \\
&\approx \mathrm{argmax}_W \sum_Q \boxed{p(O|Q)}\boxed{p(Q|W)}\boxed{p(W)}
\end{aligned}
$$

Acoustic model Lexicon Language model

Mathematical formulation (Cont'd)

$$
\begin{aligned}
\mathrm{argmax}_W p(W|O) &= \mathrm{argmax}_W \sum_Q p(W, Q|O) \\
&= \mathrm{argmax}_W \sum_Q \frac{p(W, Q, O)}{p(O)} \\
&= \mathrm{argmax}_W \sum_Q \frac{p(O|W, Q)p(Q|W)p(W)}{p(O)} \\
&= \mathrm{argmax}_W \sum_Q p(O|W, Q)p(Q|W)p(W) \\
&\approx \mathrm{argmax}_W \sum_Q \boxed{p(O|Q)}\boxed{p(Q|W)}\boxed{p(W)}
\end{aligned}
$$

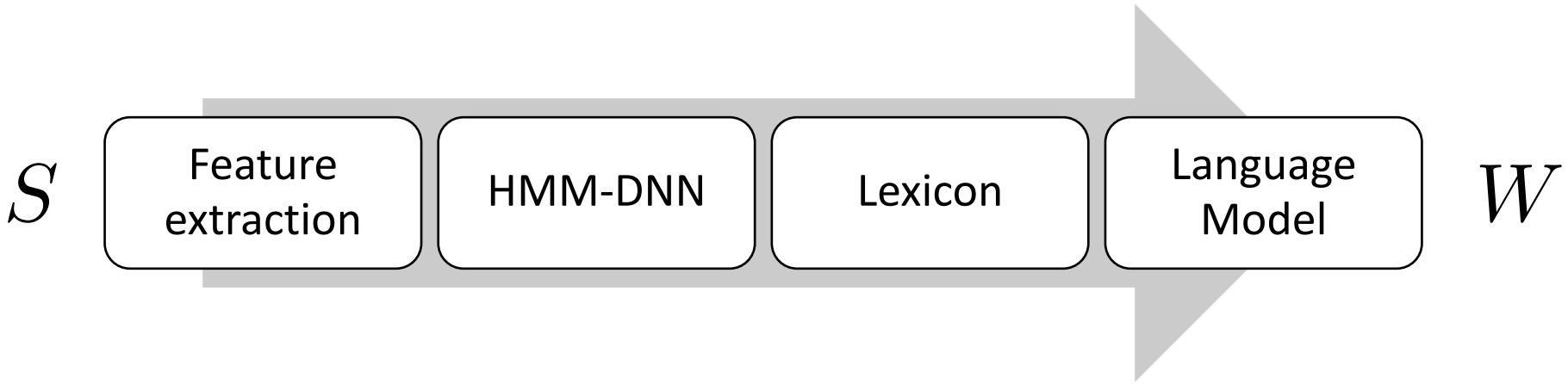
Acoustic model -> HMM Lexicon -> Dictionary Language model -> N-gram

Introduce neural networks

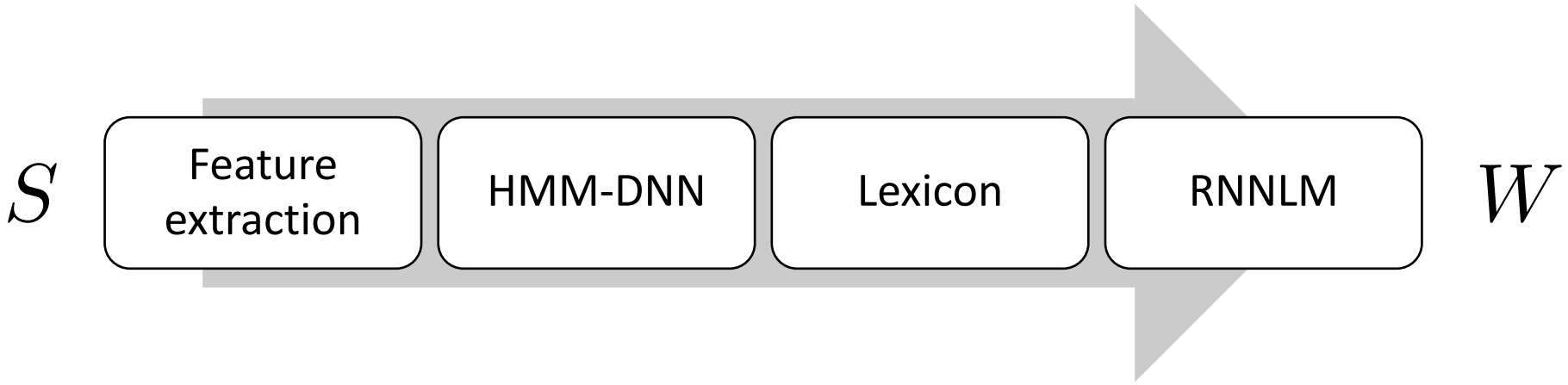
- The initial step - Incorporating DNN to HMMs
  - Tandem
    - Training a neural network with a bottleneck layer first and then use the bottleneck features instead of spectral features (e.g., MFCC)
  - Hybrid
    - By using conditional independence, we factorize the likelihood into pseudo likelihood estimated by networks (note that we still need to use HMM to get alignment for networks)

$$
\begin{aligned}
p(O|Q) &= \prod_{t=1}^{T} p(o_t|o_1, ..., o_{t-1}, Q) \\
&\approx \prod_{t=1}^{T} p(o_t|q_t) \\
&= \prod_{t=1}^{T} \frac{p(q_t|o_t)p(o_t)}{p(q_t)}
\end{aligned}
$$

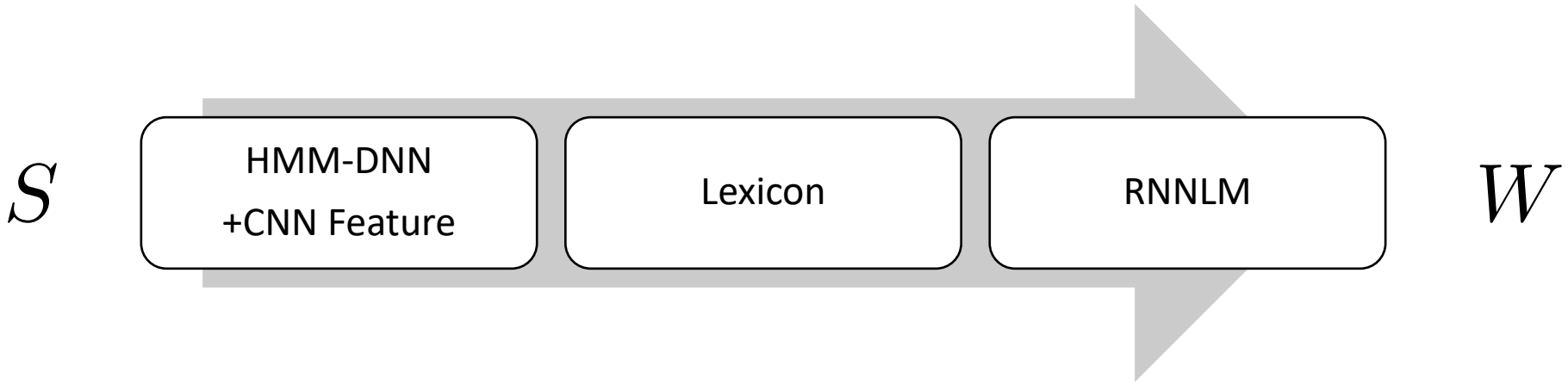
HMM-DNN ASR



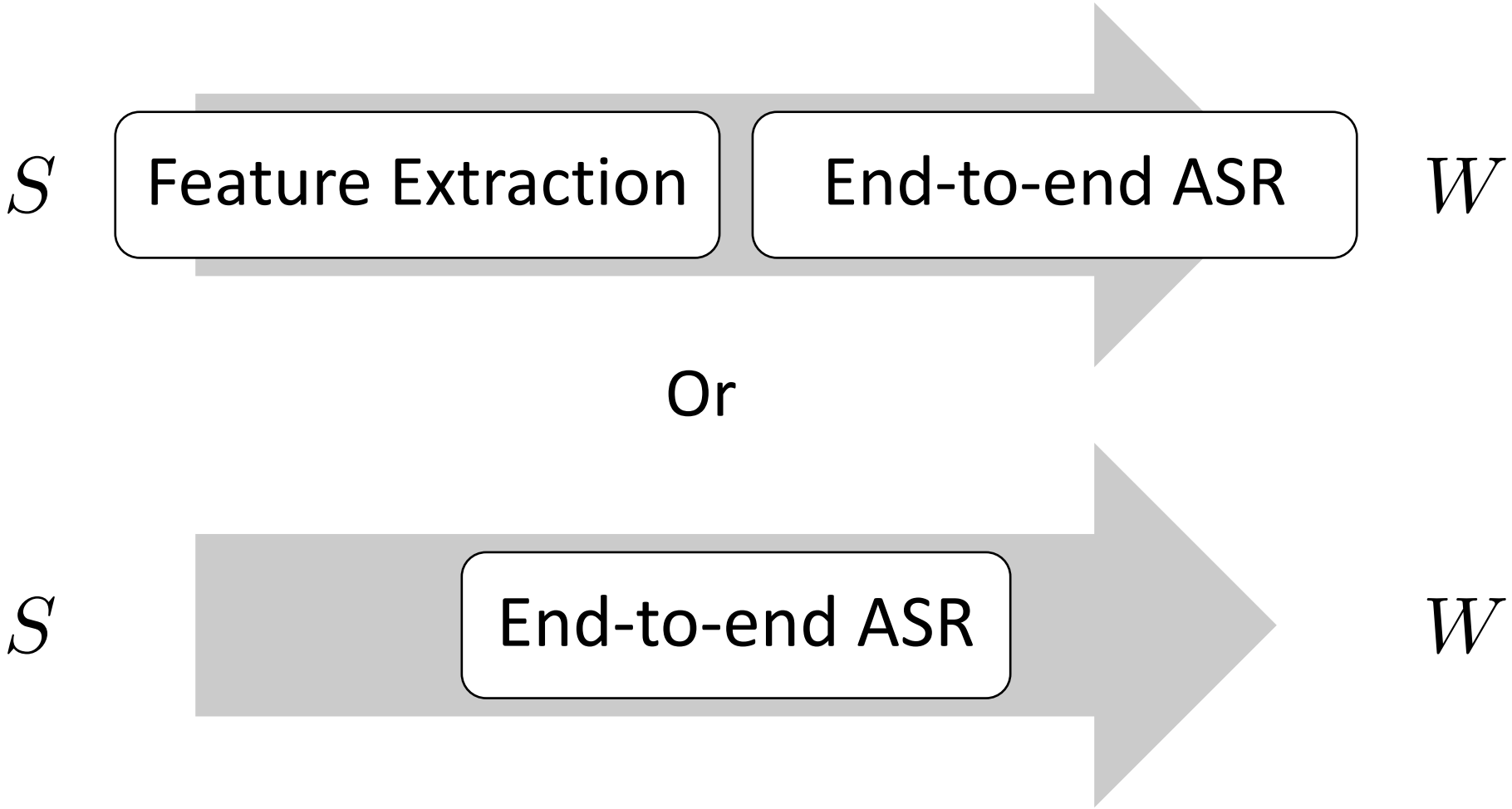
Neural network-based LM



Convolutional networks for feature extraction



Can we do end-to-end?



Sequence-to-sequence ASR

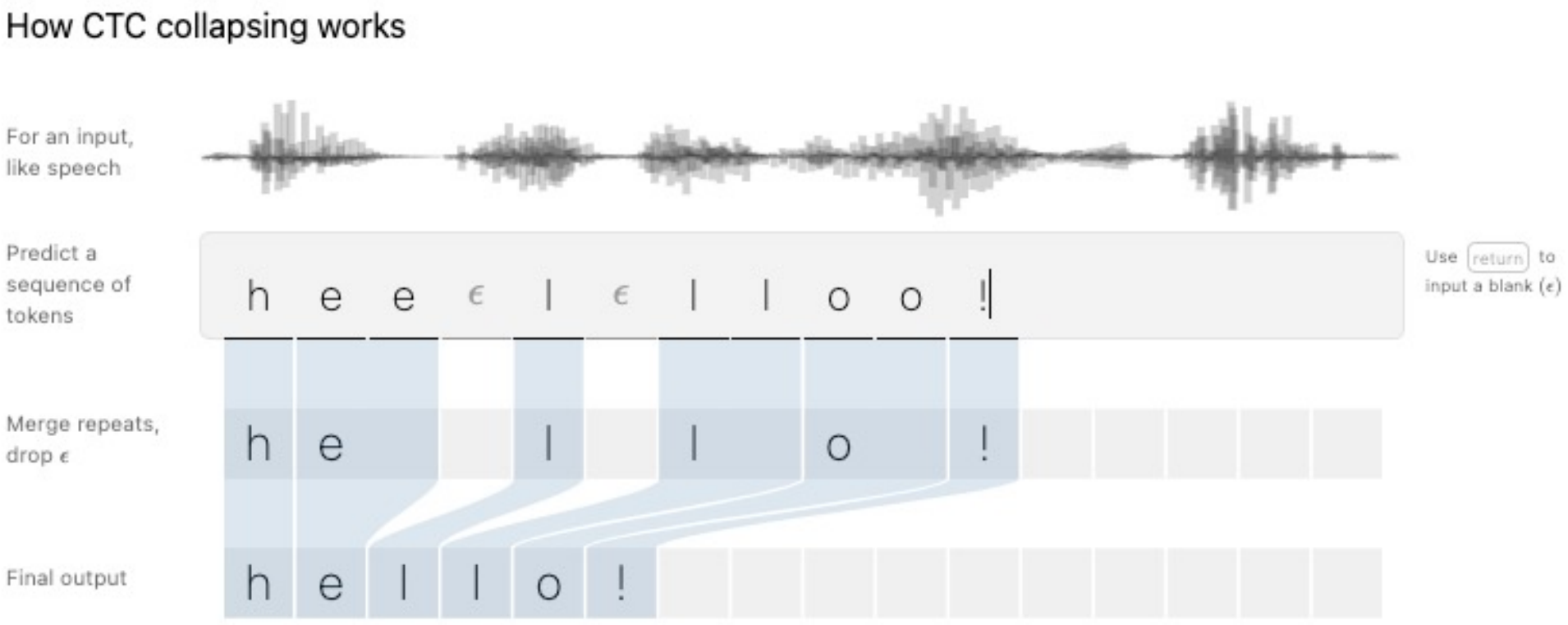
- How to achieve that?
  - HMM-based pipeline
  - Connectionist temporal classification (CTC)
  - Attention-based encoder decoder
  - RNN transducer

Connectionist temporal classification (CTC)

- Method
  - Predicting frame-based labels (the vocabulary and a special blank symbol)
  - Find alignment between speech features and transcription via dynamic programming

- Two requirements:
  - Input should be longer than output (always the case for speech processing)
  - Input and output should be monotonical aligned (the exact case for ASR)

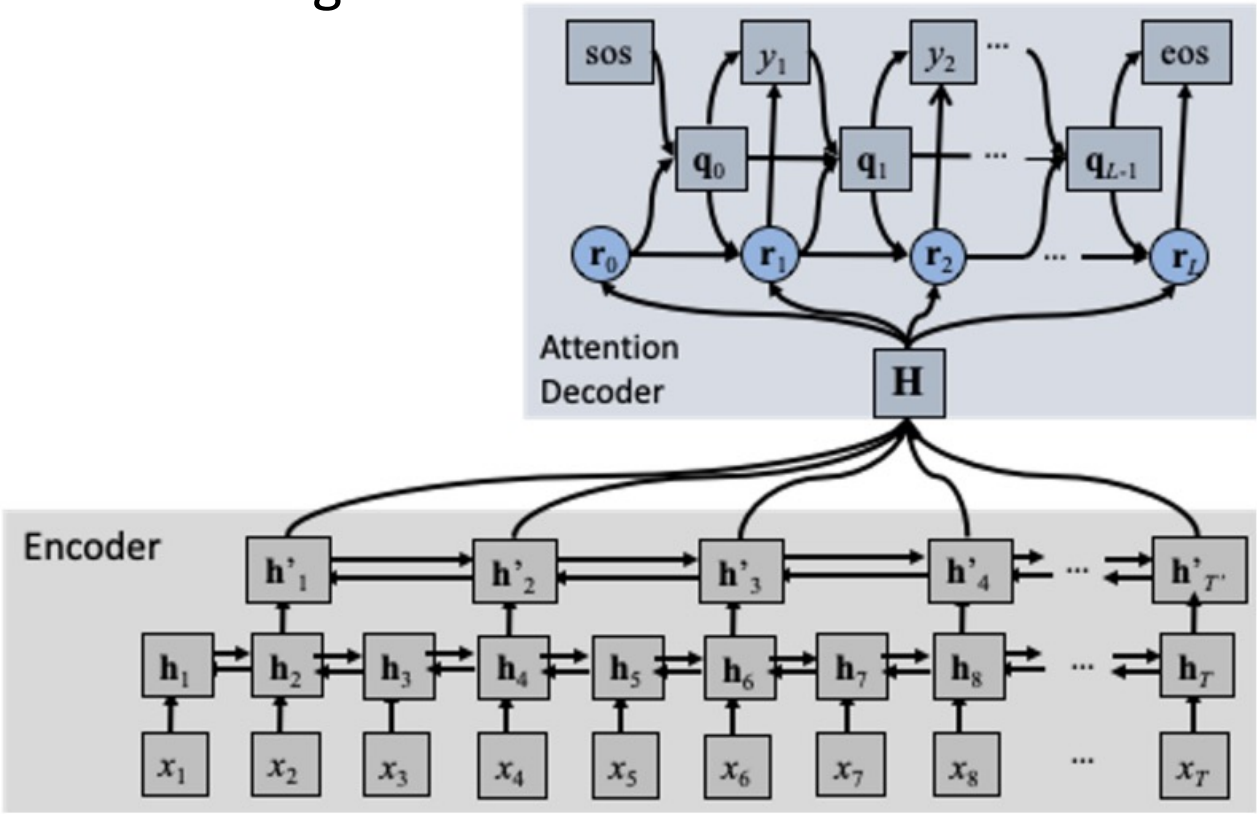
Connectionist temporal classification (CTC)



https://distill.pub/2017/ctc/

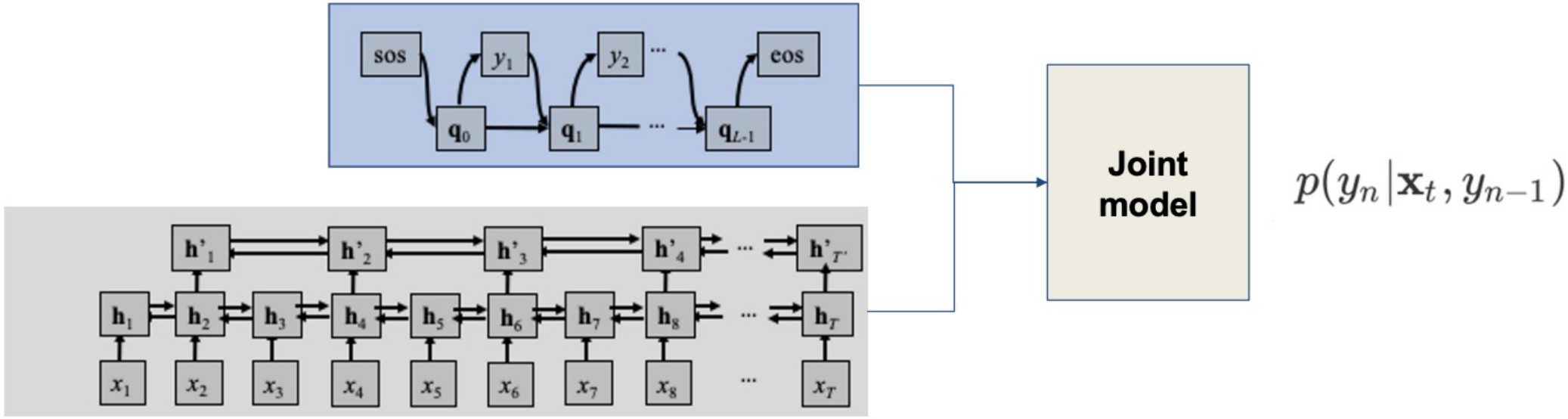
Attention-based encoder decoder

- Naturally sequence-to-sequence setups, but the alignment is too flexible to hold monotonic alignment
  - Why is that?
- Does it has other problems?



RNN transducer

- Good alignment and also be able to use in online mode
- But most of time, lower performances than attention-based methods



ESPnet project

- A comprehensive research-oriented end-to-end speech processing toolkit maintained at CMU.
- Support all three above architectures (and more)
- Demonstration in ASR:
  - https://colab.research.google.com/github/espnet/notebook/blob/master/espnet2_asr_realtime_demo.ipynb

End-to-end speech applications

- You already know
  - Speech recognition

  - Speech synthesis

Figures from https://developer.nvidia.com

End-to-end speech applications (Cont'd)

- Speech recognition
- Speech synthesis
- Voice Conversion
- Speaker verification/recognition
- Speech emotion recognition
- Speaker diarization
- Speech enhancement
- Spoken language identification
- Audio event recognition
- Speech separation
- Spoken language understanding
- Spoken dialogue systems
- Speech translation
- Multimodal processing
- Computer assisted pronunciation training

Voice conversion

- Convert a voice from a speaker to another
- A typical process
  - Disentangle source features into content, speaker information, and others
  - Keep the source features but with a changed speaker information
  - Resynthesis speech
- Some examples?
  - https://arkhamimp.github.io/TTL-VC/

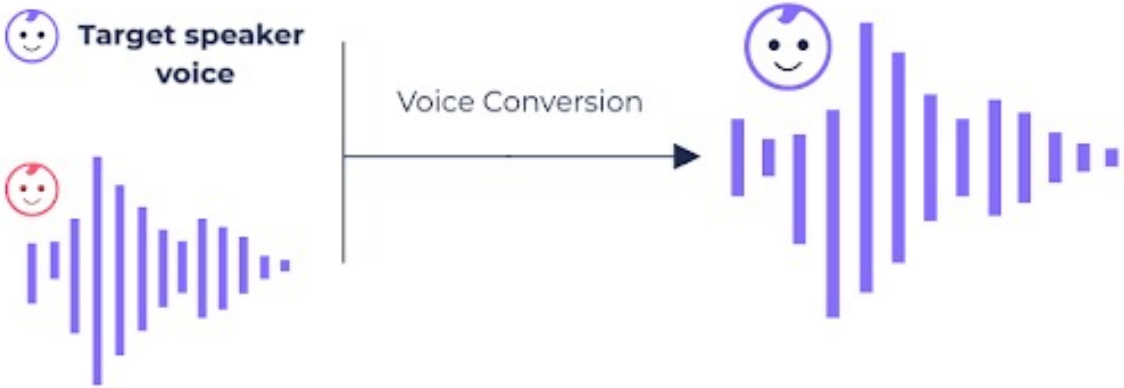
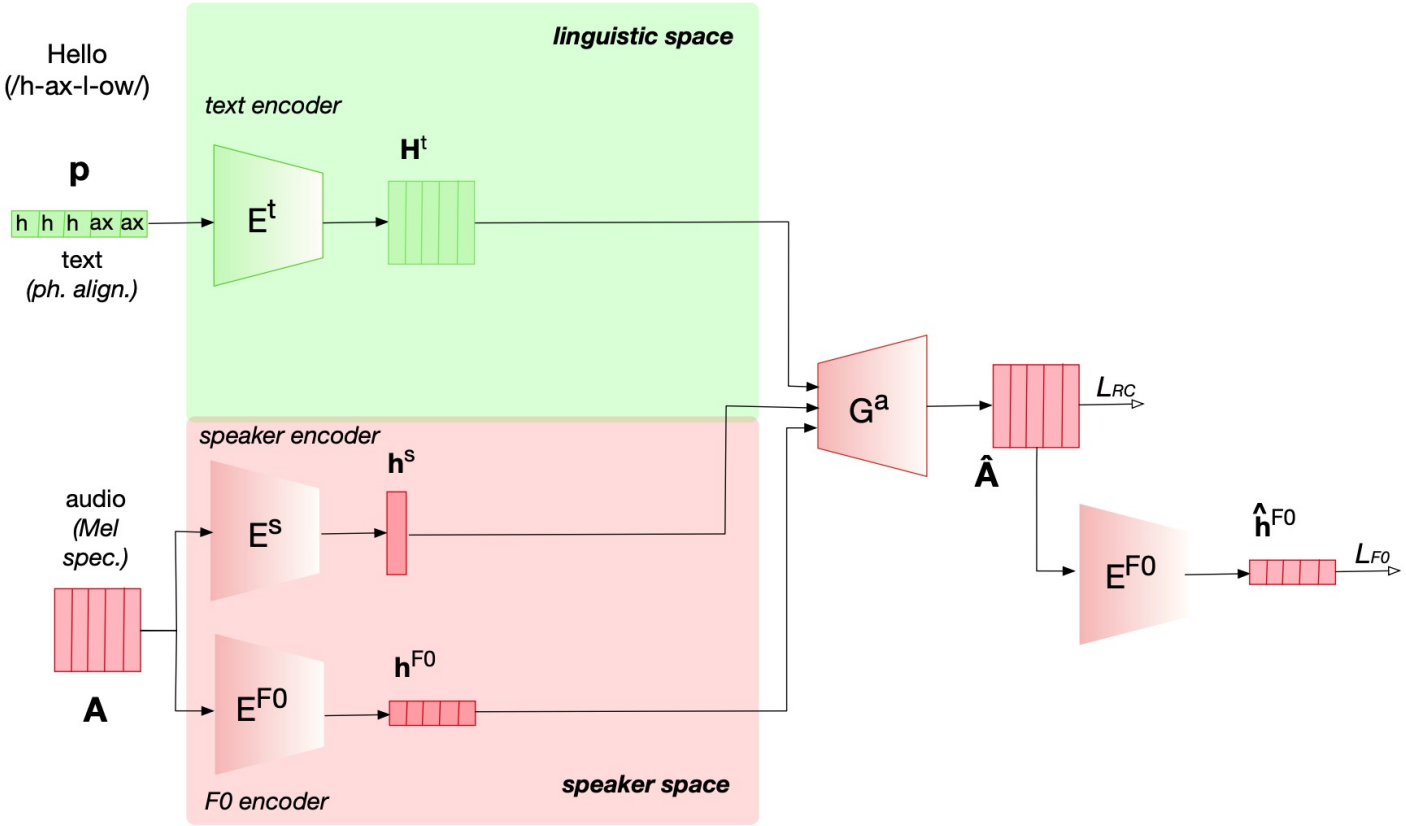


Figure from https://dasha.ai/en-us/blog/multi-speaker-text-to-speech

Voice conversion in neural networks



SEQUENCE-TO-SEQUENCE VOICE CONVERSION USING F0 AND TIME CONDITIONING AND ADVERSARIAL LEARNING (Bous et al. 2021)

Speaker verification / recognition

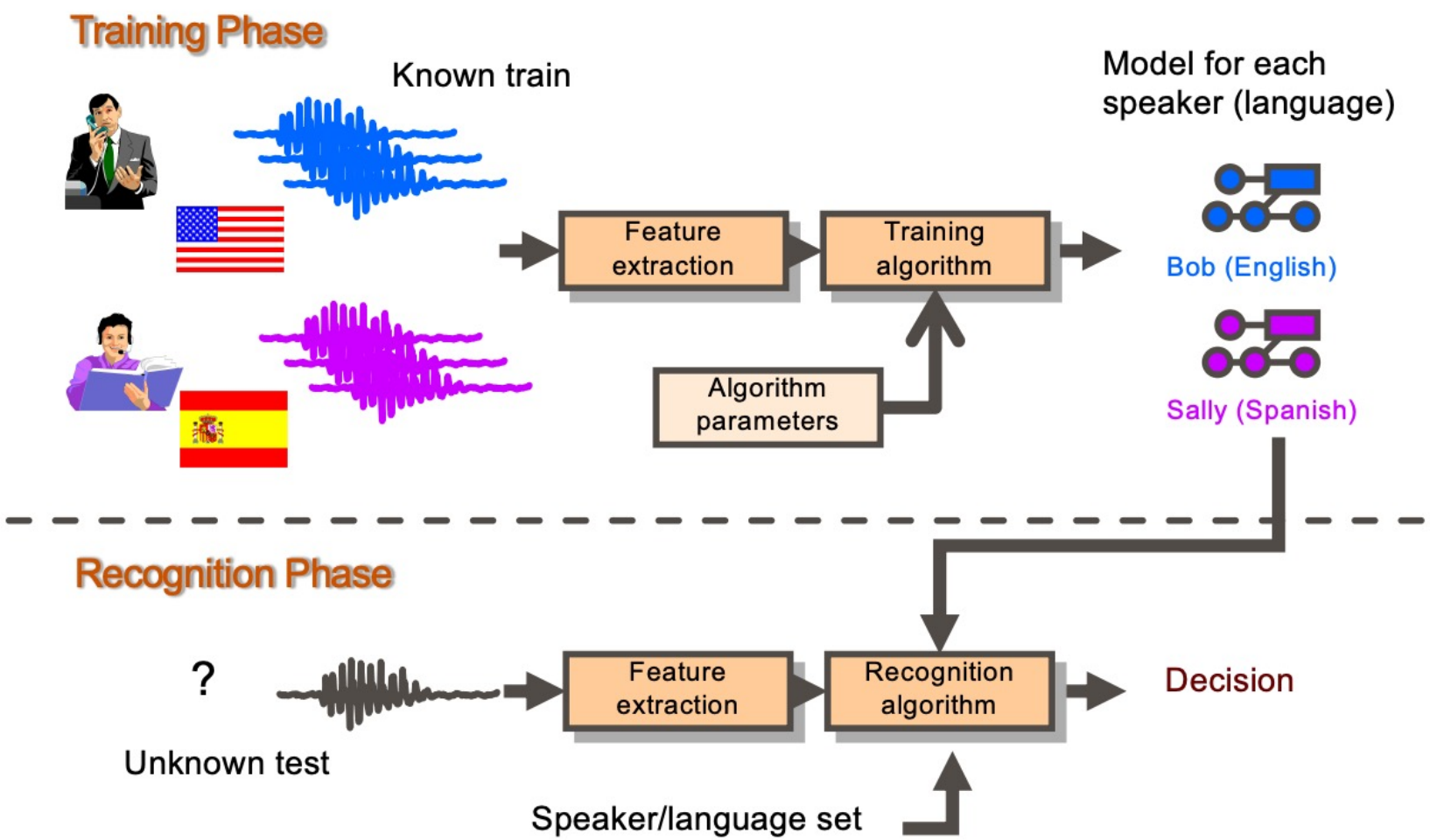
- What’s the difference of verification and recognition?

Speaker verification / recognition

- What’s the difference of verification and recognition?
  - Who are you?
  - Are you exactly you?

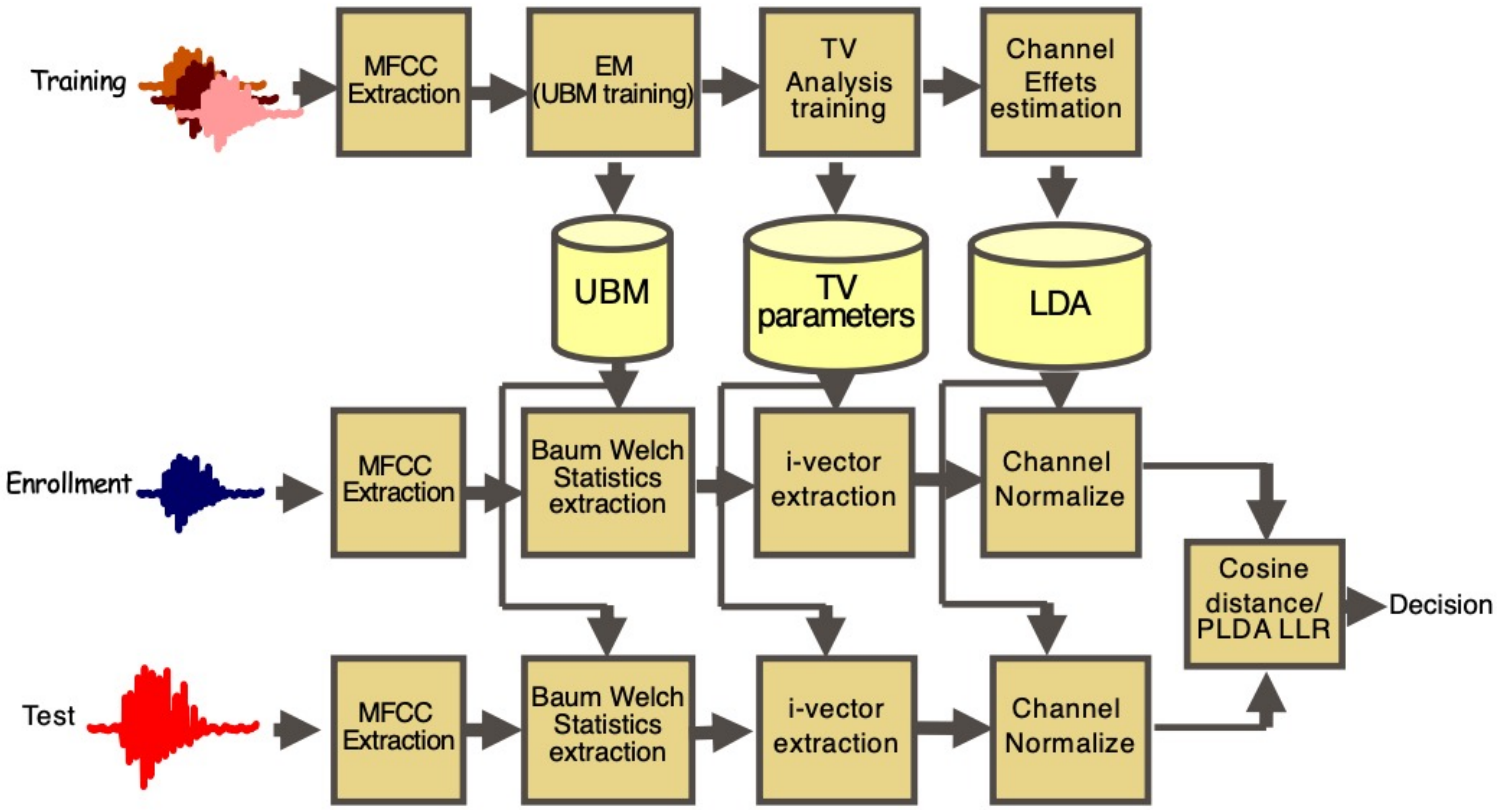
- Speaker verification:
  - Given some of a target speaker speech, verify if a new speech is from him/her
- Speaker recognition:
  - Recognize who is speaking

Recognition pipeline



Figures borrow from https://jhu-intro-hlt.github.io/slides/speaker-id.pdf

Verification pipeline (i-vector pipeline)



Speech translation

- Translate a speech from a source language into another language
- Two major types:
  - Speech-to-text translation
    - ASR + MT or end-to-end system?
  - Speech-to-speech translation
    - ASR + MT + TTS or end-to-end system?

- Compare with ASR setup, what could be the differences?

End-to-end speech-to-speech translation

- An emerging field with many interesting problems
  - Evaluation of speech-to-speech translation system
  - Simultaneous speech-to-speech translation
  - Speaker retain

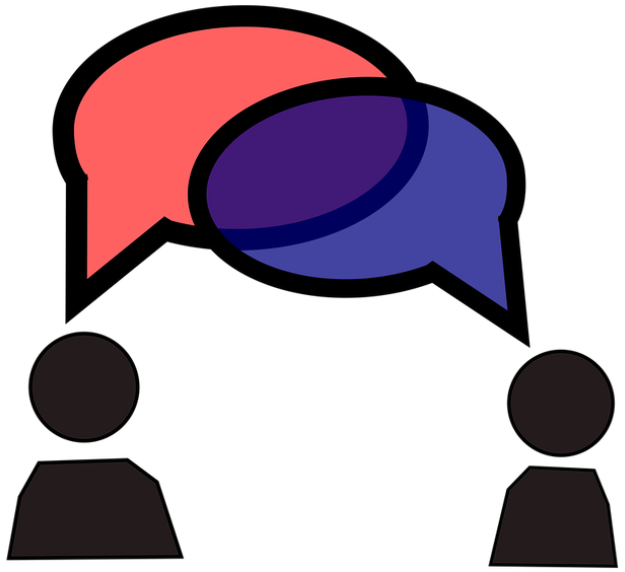
End-to-end speech-to-speech translation

- Evaluation of speech-to-speech translation system
  - For ASR, we have word error rate; For MT/S2T, we have BLEU (wil covered later): how we evaluate s2st system?

End-to-end speech-to-speech translation

- Simultaneous speech-to-speech translation
  - Many sub-modules are involved in the system, making it quite difficult to achieve streaming

- Why streaming?
  - Let say conversation



End-to-end speech-to-speech translation

- Speaker retain
  - How to keep the source speaker identity?

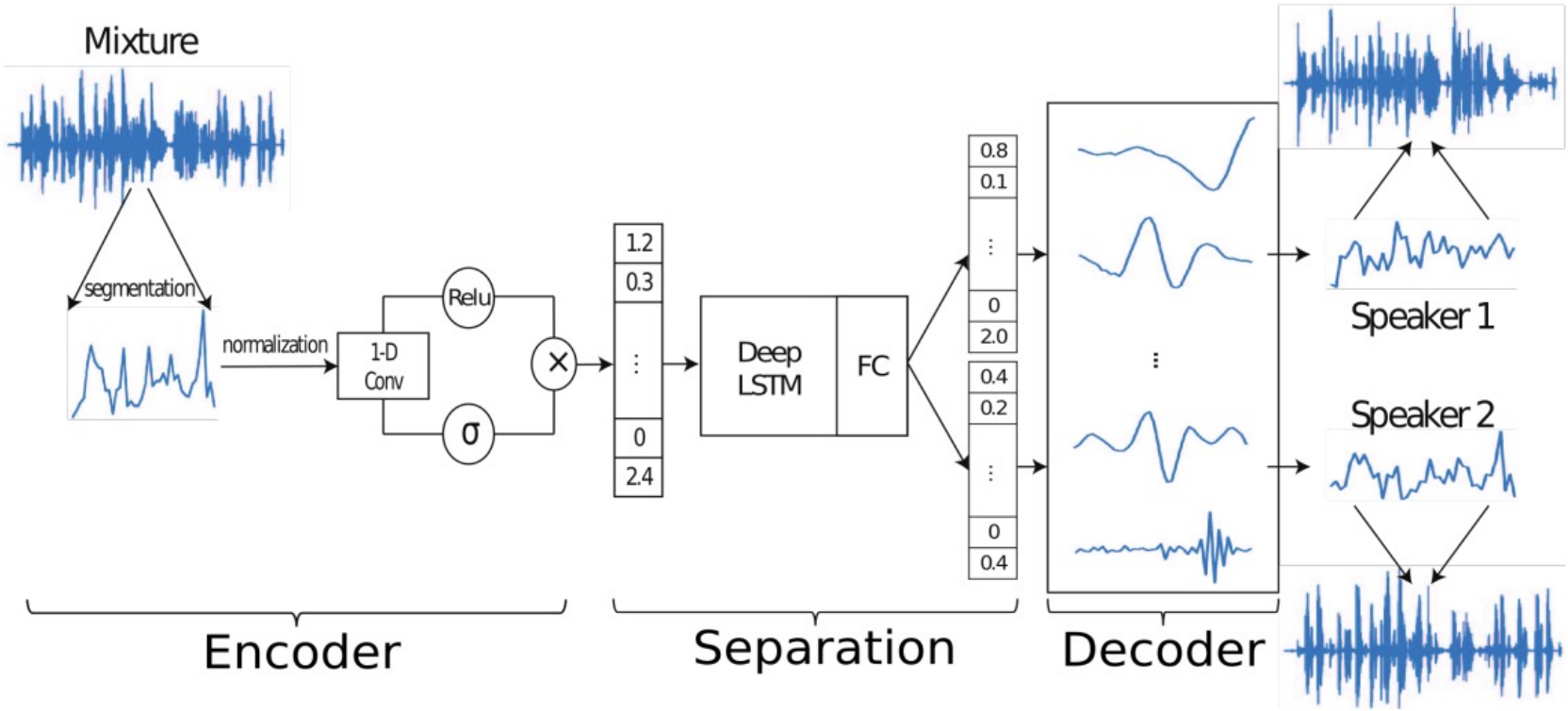
This Photo by Unknown Author is licensed under CC BY-SA-NC

Speech enhancement & separation

- Enhance the target speech / separate mixture of speech

- Very important feature since COVID-19
- https://youtu.be/NzZDnRni-8A (a very interesting example)

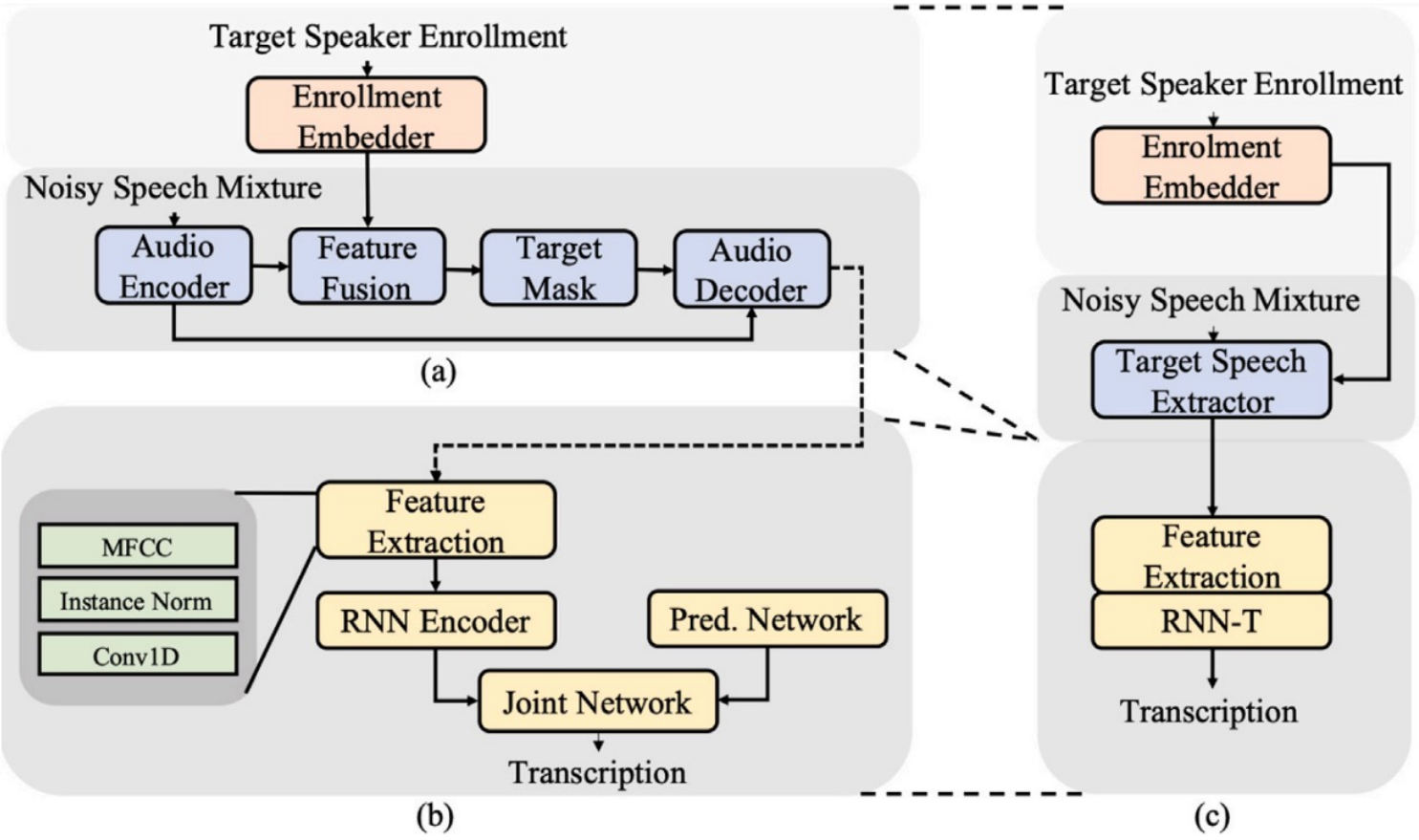
Speech enhancement & separation



TaSNet: Time-Domain Audio Separation Network for Real-Time, Single-Channel Speech Separation (Luo et al. 2018)

Speech enhancement as a front-end

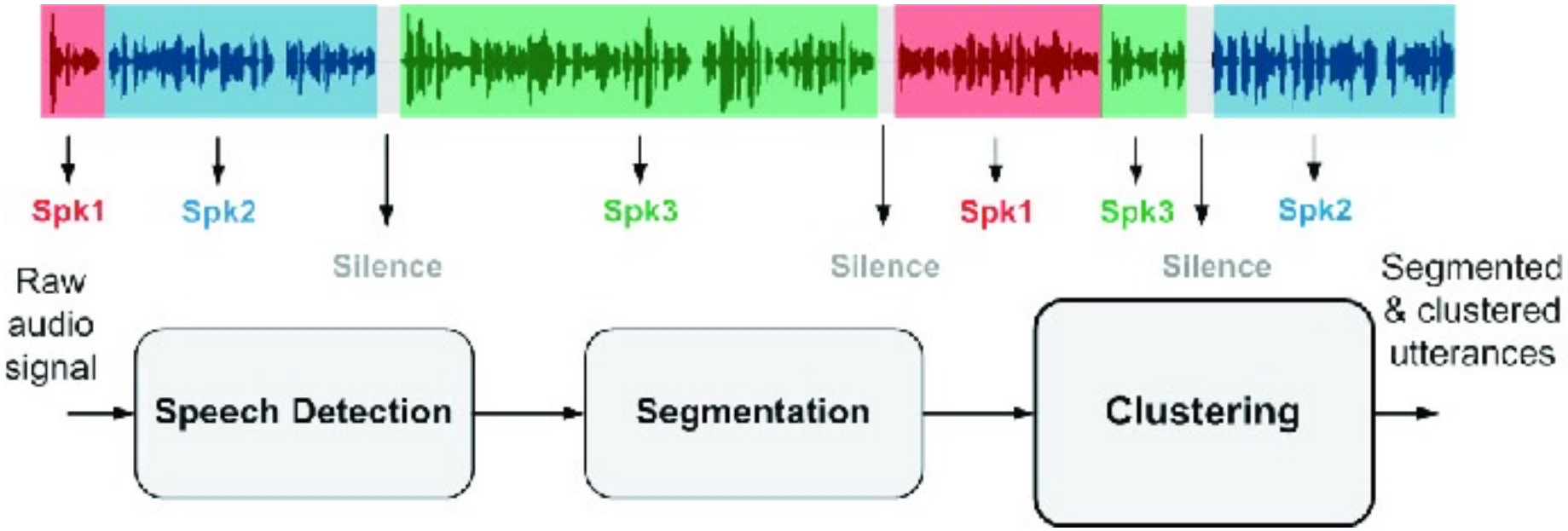
- SE + ASR
- SE + ST
- SE + SV
- SE + …



An investigation of neural uncertainty estimation for target speaker extraction equipped RNN transducer (Shi et al. 2022)

Speaker diarization

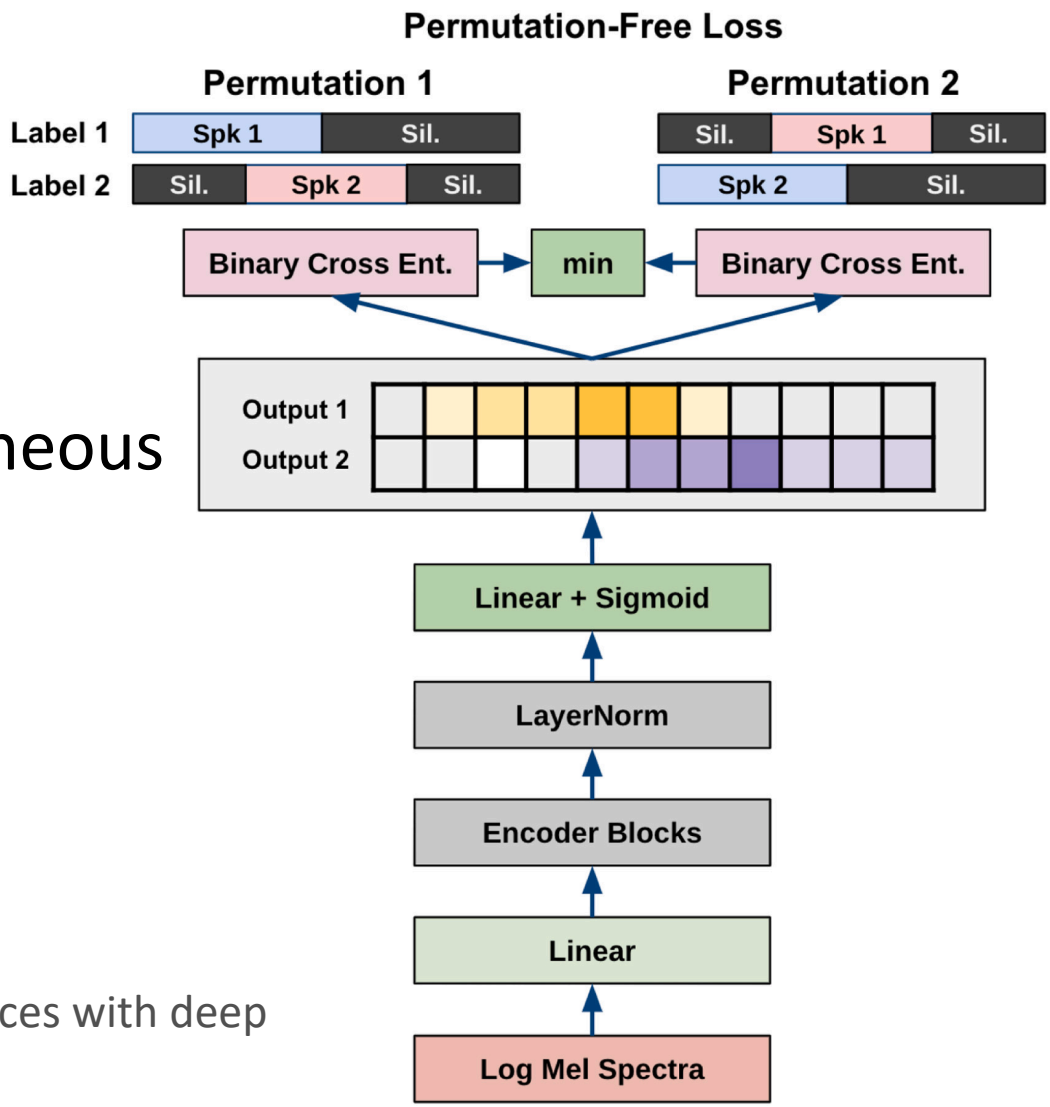
- Target the problem of “when and who speaks”



Partially Supervised Speaker Clustering (Tang et al. 2011)

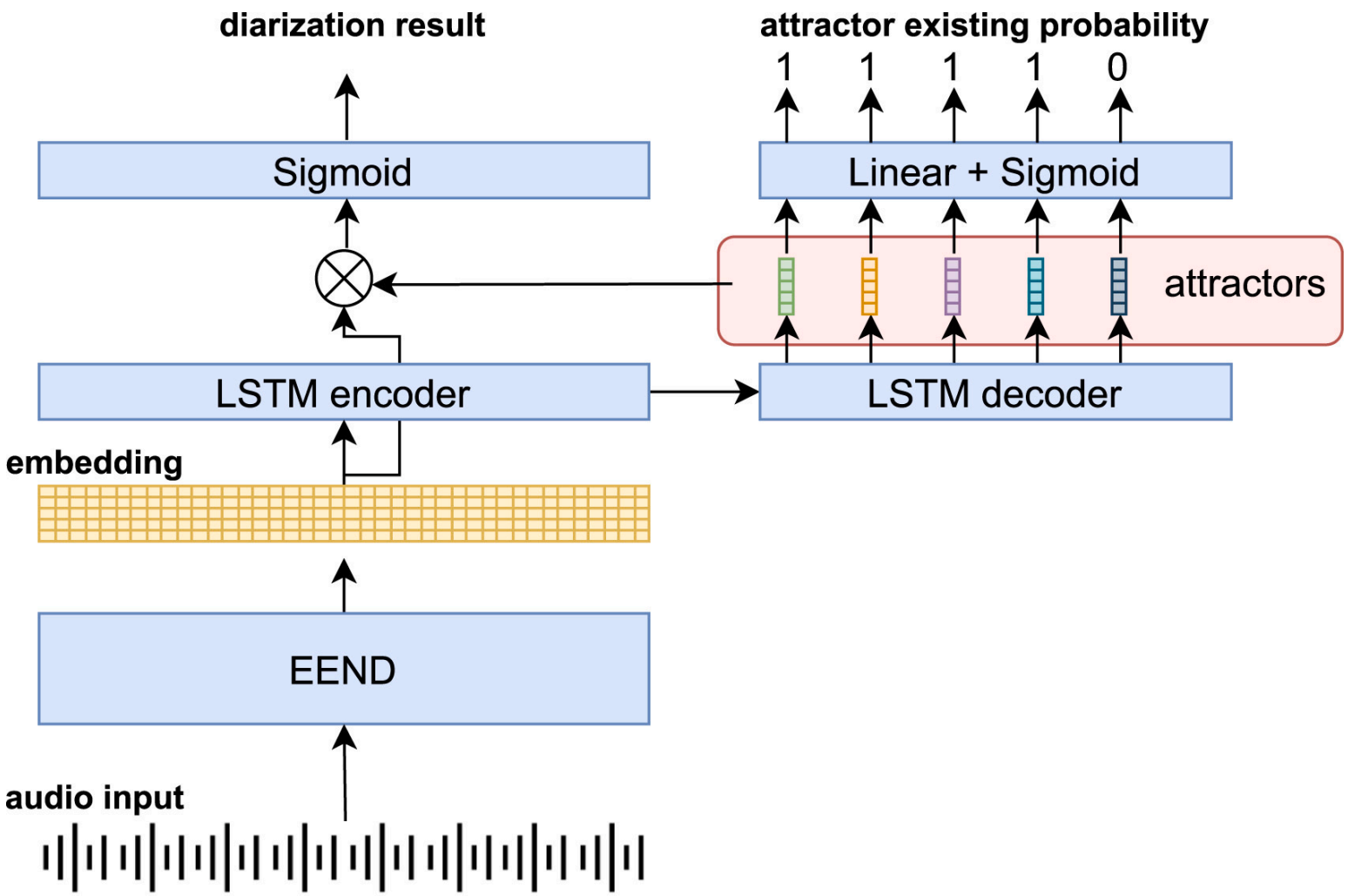
Neural diarization system

- Issues with previous segmentation:
  - Cannot handle overlap
- EEND predict speaker occurances simultaneous
  - Can handle overlap cases
  - But need to fix speaker number ahead
  - (How to solve it?)



A review of speaker diarization: Recent advances with deep learning (Park et al. 2021)

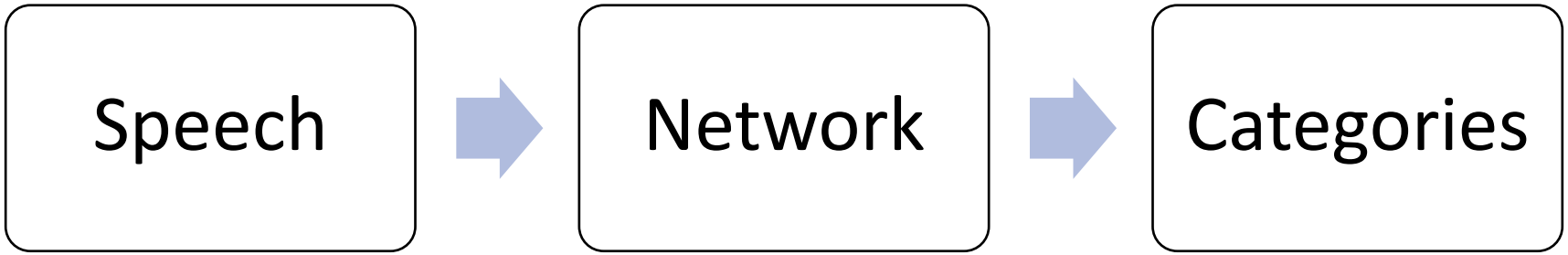
Neural diarization system (Cont'd)



Extend ASR for other speech applications

- The model setup for ASR can be used for various speech applications

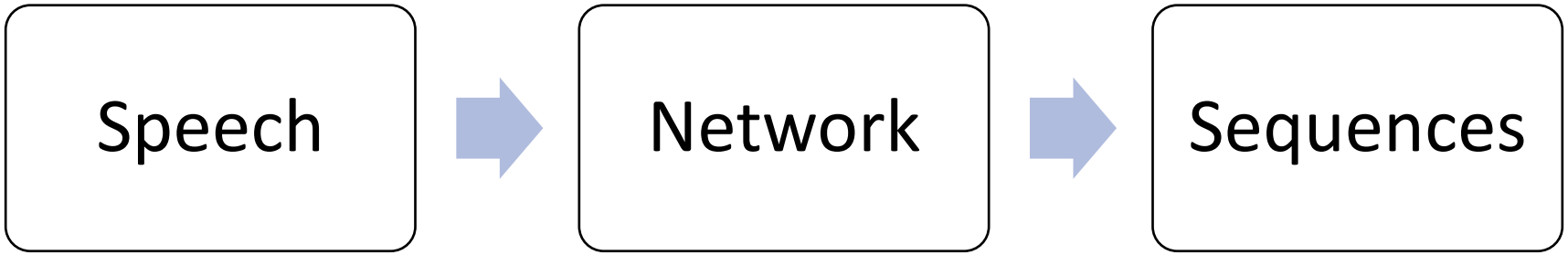
- For classification tasks, we can simply replace the ASR transcript with the corresponding labels
  - Speech emotion recognition -> use emotion category
  - Intent classification -> use intent category
  - Audio event recognition -> use audio event category



Extend ASR for other speech applications

- The model setup for ASR can be used for various speech applications

- For seq2seq tasks, we replace ASR transcript with corresponding sequence
  - Speech summarization -> summarized scripts
  - Audio caption -> caption of the speech

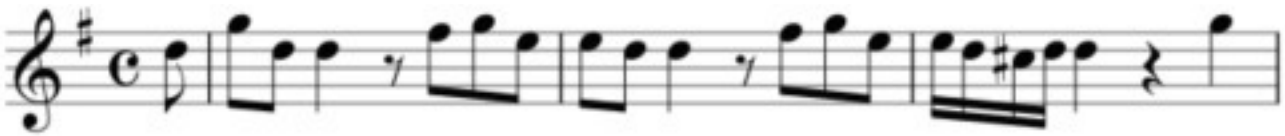


Extend speech technology to music

- Music has many similar problem setup as speech and could easily use some speech-based methods in many of its applications
  - Automatic music transcription
  - Singing voice synthesis
  - Many others

Figure from www.dreamstime.com

Automatic music transcription



CTC-friendly encoding

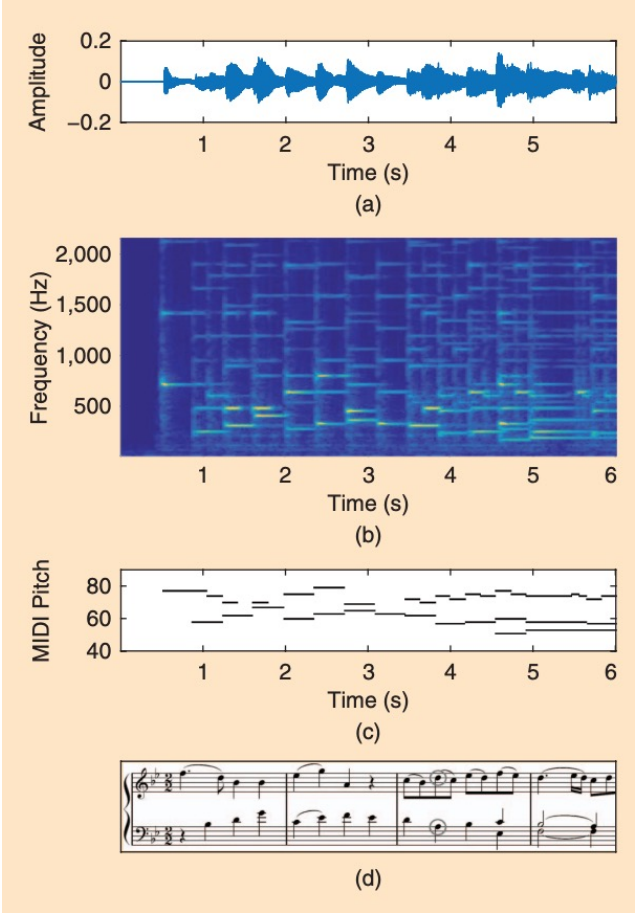
```
$ t ( d5- | g5- d5- d5! ,- f#5- g5- e5- | e5- d5- d5! ,- f#5- g5- e5- |
e5= d5= c#5= d5= d5! ,! g5! |
```

PAE-based encoding

```
G-2 xF 4/4 8''D / G D 4D 8- F G E / E D 4D 8- F G E / 6E D xC D 4D - G /
```

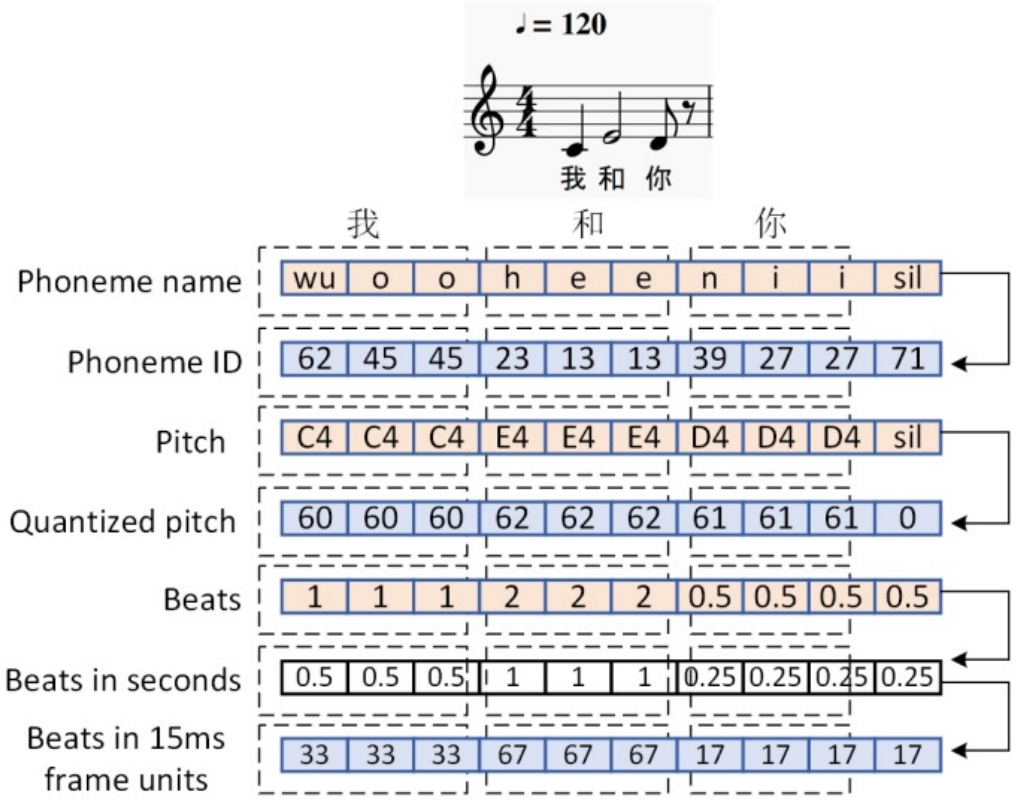
Kern-based encoding

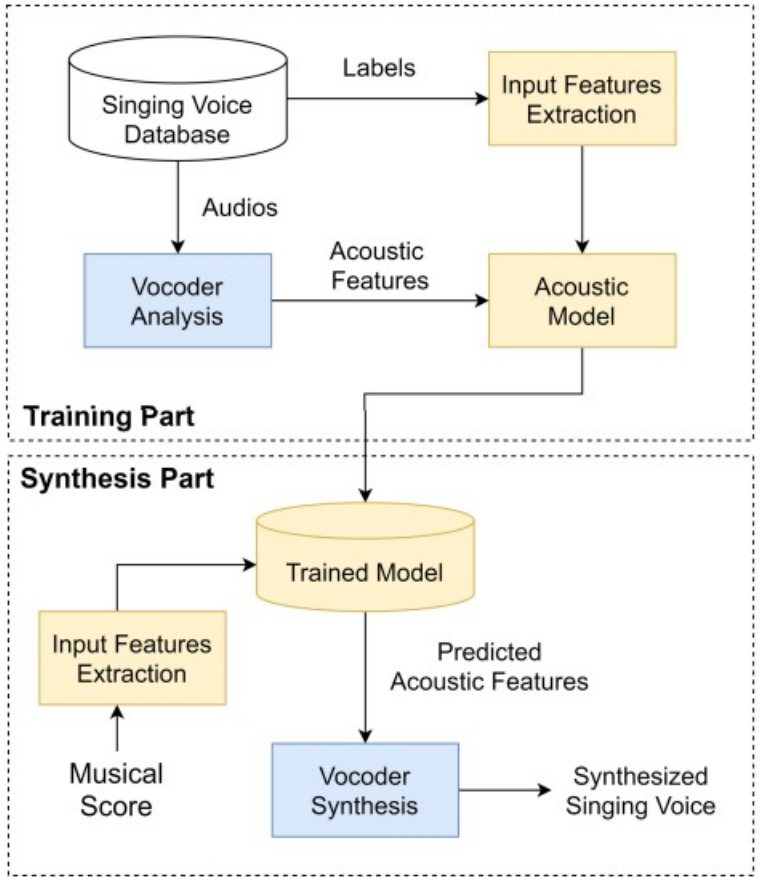
```
G-2 xF 4/4 8dd = 8gg 8dd 4dd 8r 8ff# 8gg 8ee = 8ee 8dd 4dd 8r 8ff# 8gg 8ee =
16ee 16dd 16cc# 16dd 4dd 4r 4gg =
```



Data representations for audio-to-score monophonic music transcription (Roman et al. 2020)
Automatic Music Transcription: an overview (Benetos et al. 2019)

Singing voice synthesis





XiaoiceSing: A High-Quality and Integrated Singing Voice Synthesis System (Lu et al. 2020)
A Survey on Recent Deep Learning-driven Singing Voice Synthesis Systems (Cho et al. 2021)

Muskits project

- A open-source music processing toolkit focus on end-to-end singing voice synthesis (also lead in CMU)
- Up to four singing voice synthesis models for four languages, ten corpora
- Singing voice synthesis demo:
  - https://colab.research.google.com/github/SJTMusicTeam/svs_demo/blob/master/muskit_svs_realtime.ipynb

Want to learn more in speech?

- 11-785 Deep learning
- 11-751 Speech recognition and understanding
- 11-492/692 Speech processing
- 11-737 Multilingual NLP