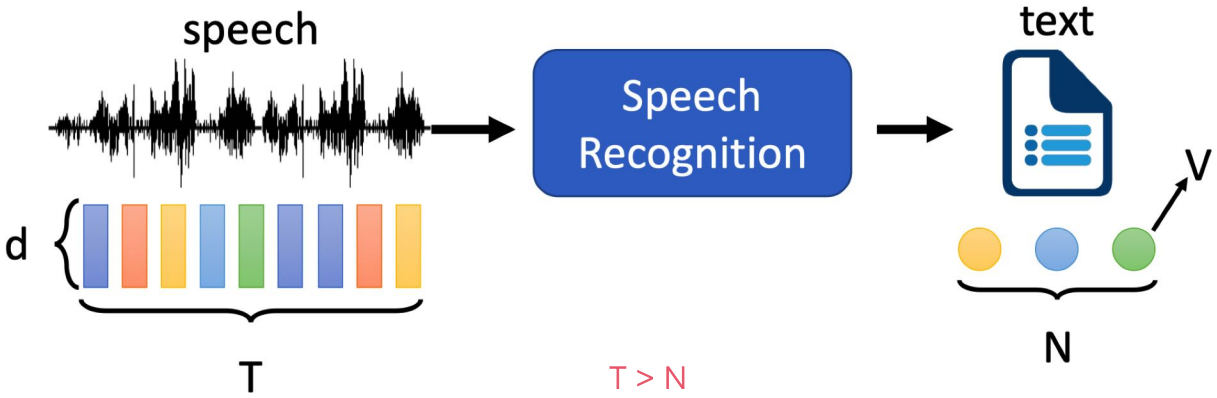
Intro to ASR

吴金龙

2020.11

ASR

- **Speech In, Text Out**



Speech: a sequence of vector (length T, dimension d)

Text: a sequence of token (length N, V different tokens)

ASR

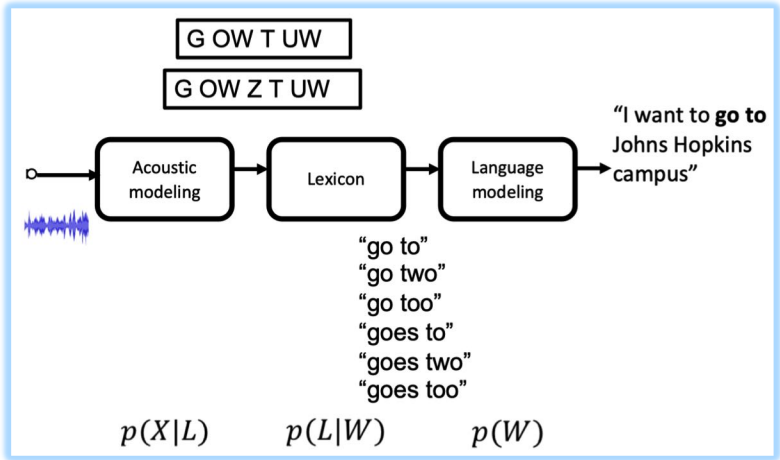
- **Classical method** (1970-)

$$\arg\max_{W} p(W|X) = \arg\max_{W} p(X|W)p(W)$$
$$\approx \arg\max_{W,L} p(X|L)p(L|W)p(W)$$

X: Speech sequence
W: Text sequence
L: Phoneme sequence

- **Speech recognition**
  - $p(X|L)$: Acoustic model (Hidden Markov model)
  - $p(L|W)$: Lexicon
  - $p(W)$: Language model (n-gram)

Modular-based integration



ASR

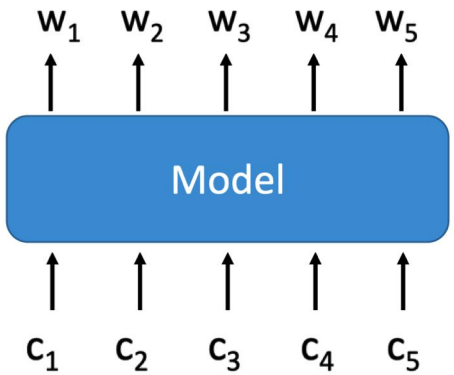
- **Metrics**
  - **CER**: Character Error Rate
  - **WER**: Word Error Rate

    WER= S+I+D/N

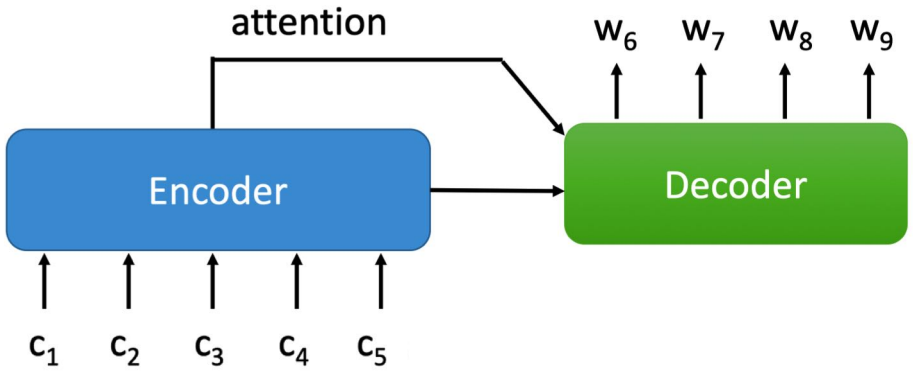
    1. **S** stands for substitutions,
    2. **I** stands for insertions,
    3. **D** stands for deletions,
    4. **N** is the number of words in the reference (that were actually said).
  - **SER**: Sentence Error Rate

ASR

- End-to-End Neural Network：两种基本框架

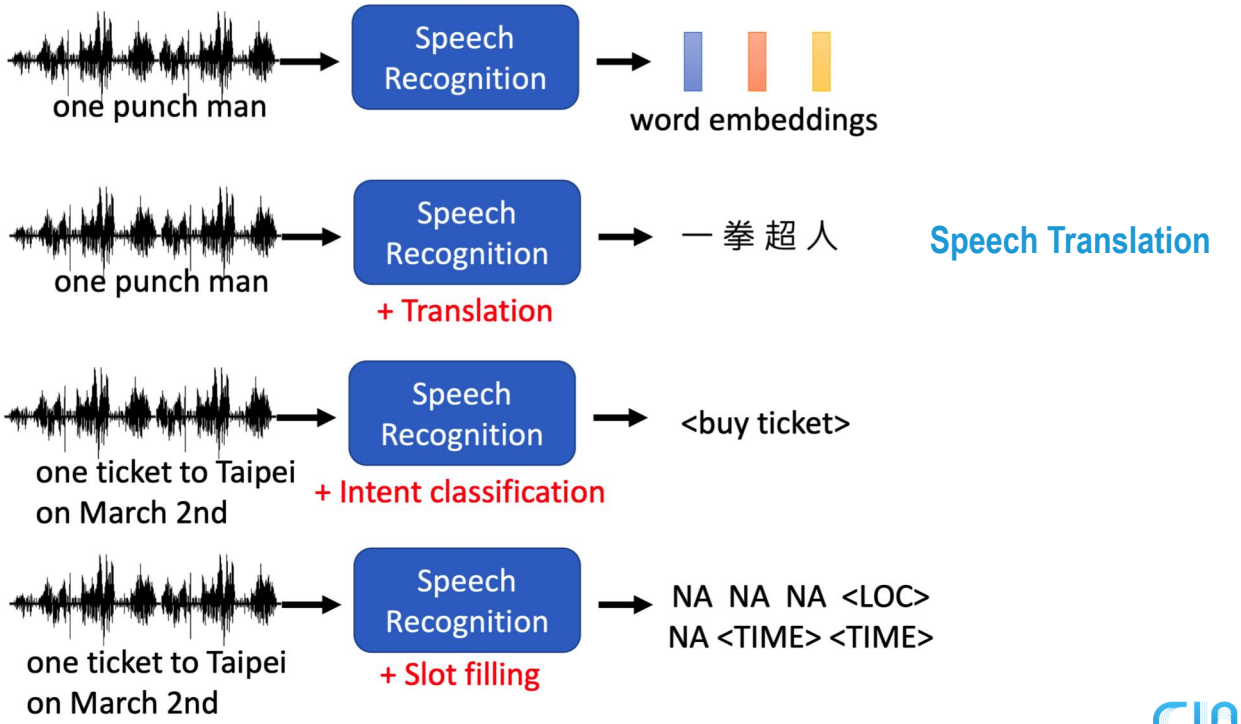


Sequence Labeling



Seq2seq

More Applications



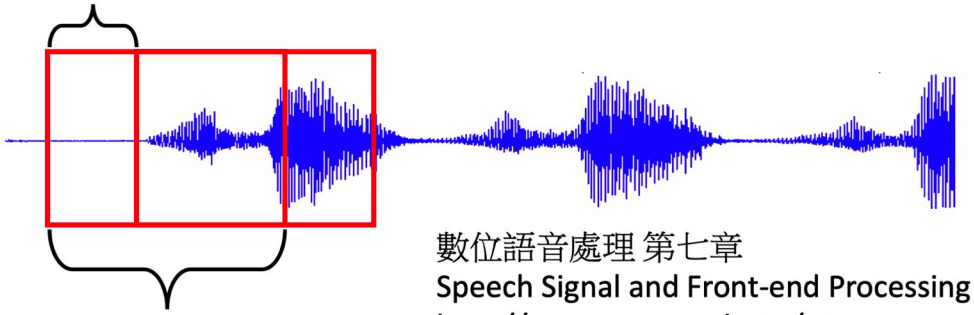
输入的语音特征（声学特征）

Acoustic Feature

length T, dimension d

10ms

1s → 100 frames



25ms

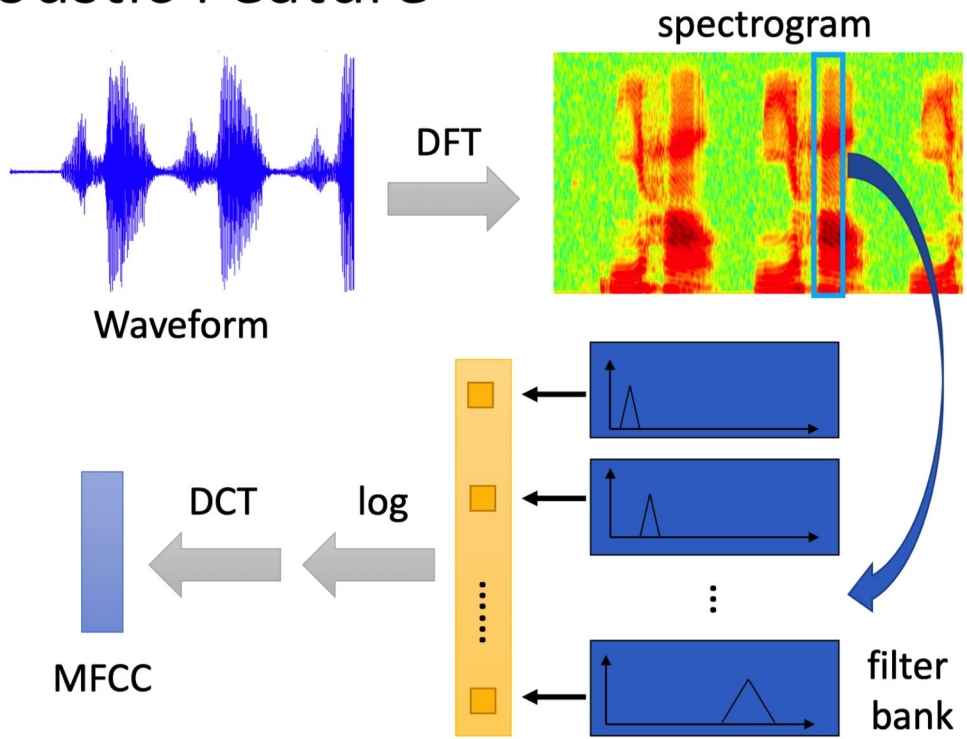
數位語音處理 第七章
Speech Signal and Front-end Processing
http://ocw.aca.ntu.edu.tw/ntu-ocw/ocw/cou/104S204/7

frame

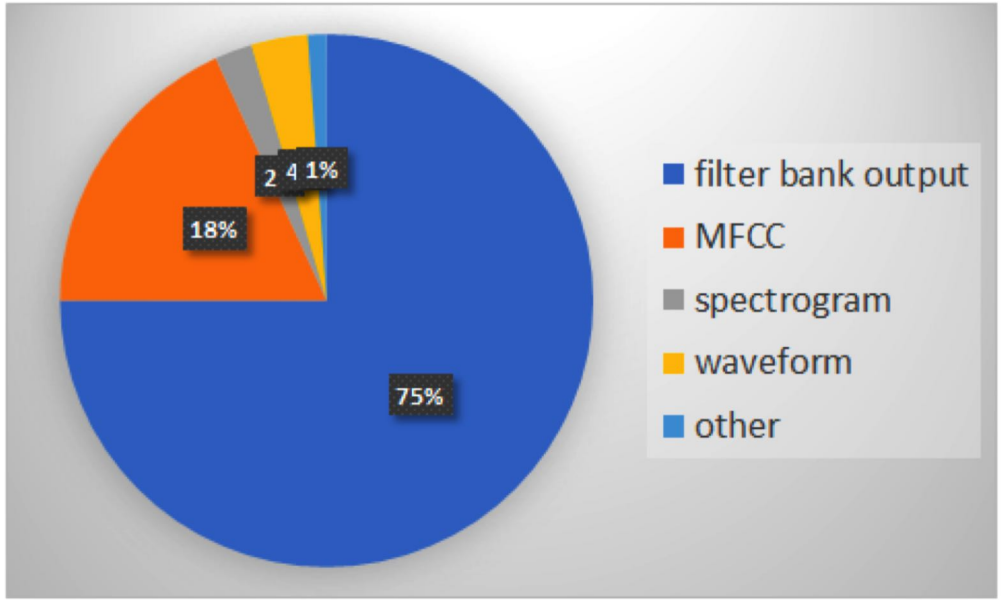
400 sample points (16KHz)
39-dim MFCC
80-dim filter bank output

输入的语音特征（声学特征）

Acoustic Feature



输入的语音特征（声学特征）



- Go through more than 100 papers in INTERSPEECH'19, ICASSP'19, ASRU'19
- 模型越发彪悍了，特征越发原始了

输出的文字 token

Token

Phoneme: a unit of sound

W AH N	P AH N CH	M AE N
one	punch	man

Lexicon: word to phonemes

cat ⟶ K AE T
good ⟶ G UH D
man ⟶ M AE N
one ⟶ W AH N
punch ⟶ P AH N CH

Grapheme: smallest unit of a writing system

Lexicon free!

one_punch_man
N=13, V=26+?

“一”，“拳”，“超”，“人”
N=4, V≈4000

26 English alphabet
+ { _ } (space)
+ {punctuation marks}

Chinese does not need “space”

输出的文字 token

Token

Word: one punch man ➡ N=3, usually V>100K

“一拳” “超人” ➡ N=2, V=???

For some languages, V can be too large!

Morpheme: the smallest meaningful unit (< word, > grapheme)

unbreakable → “un” “break” “able”

rekillable → “re” “kill” “able”

What are the morphemes in a language?

linguistic or statistic

输出的文字 token

- Token：拼音
  - yi1 quan2 chao1 ren2
  - 再把拼音转换为汉字，类似输入法

输出的文字 token

Token

Bytes (!): The system can be **language independent**!

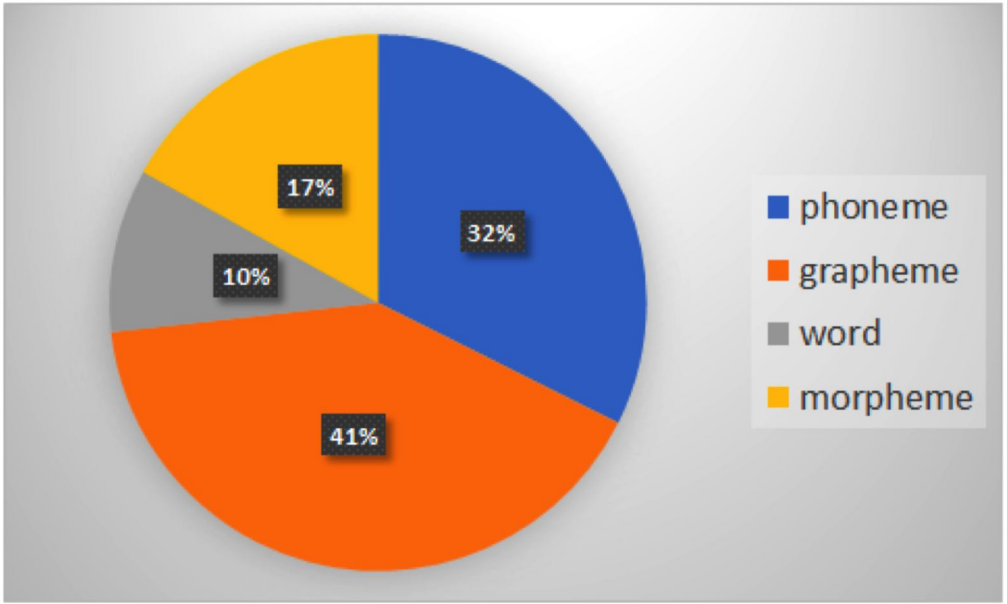
UTF-8

	Binary
$	00100100
¢	11000010 10100010
ह	11100000 10100100 10111001
€	11100010 10000010 10101100
한	11101101 10010101 10011100
𐍈	11110000 10010000 10001101 10001000

V is always 256

[Li, et al., ICASSP'19]

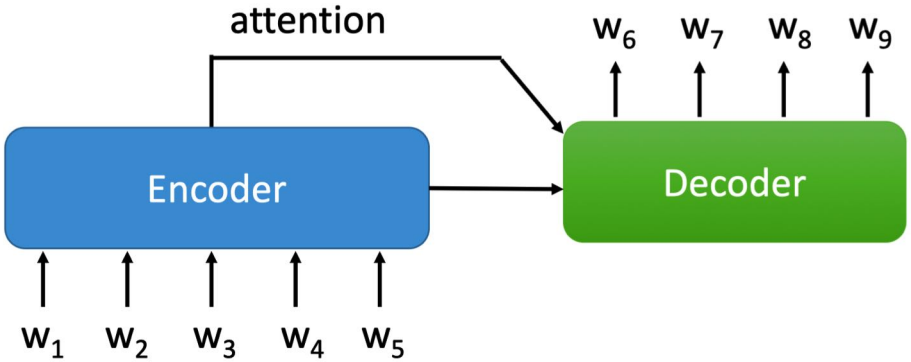
输出的文字 token



- Go through more than 100 papers in INTERSPEECH'19, ICASSP'19, ASRU'19
- 模型越发彪悍了，token 越发直白了

模型简介：Seq2seq

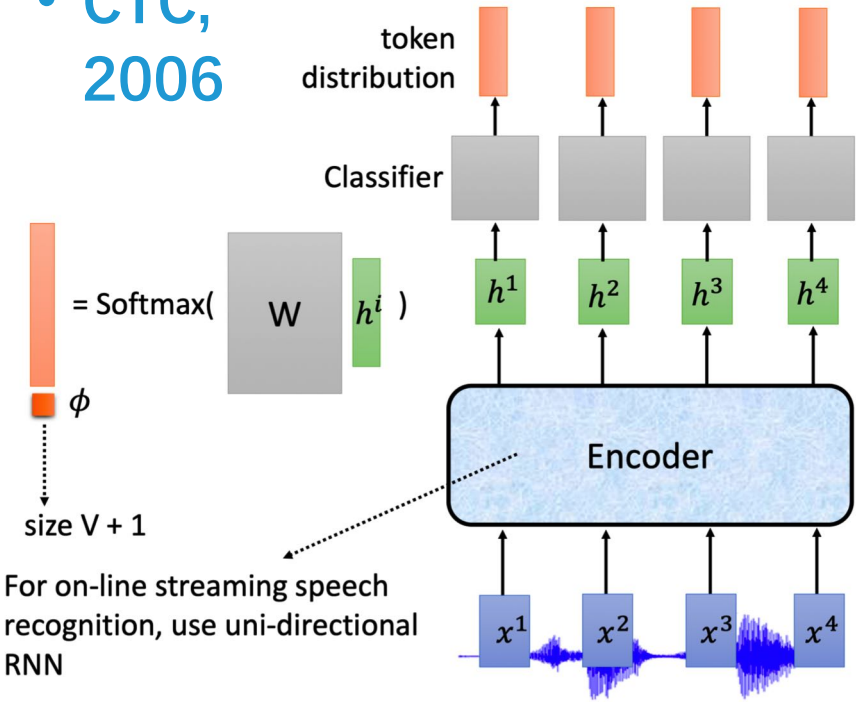
- Listen, Attend, and Spell (**LAS**)
  - Seq2seq + Attention
  - Beam Search



  - LAS outputs the first token after listening the whole input
  - not streaming

模型简介：Sequence Labeling

- CTC, 2006



For on-line streaming speech recognition, use uni-directional RNN

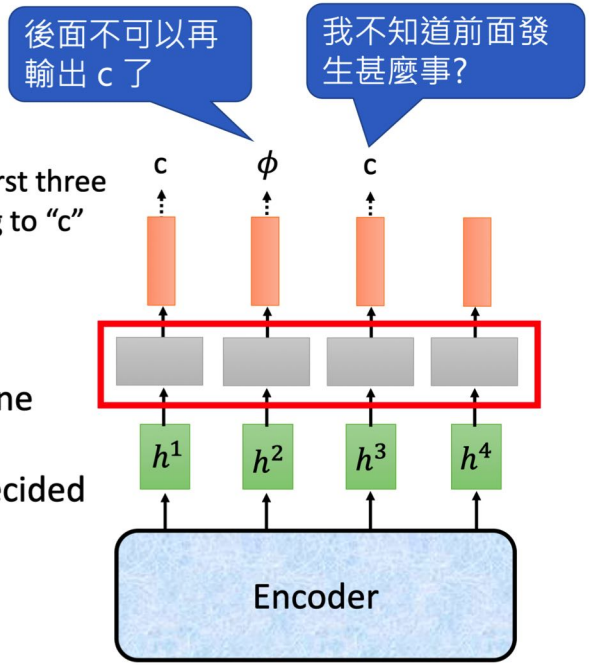
Issue

Assume the first three frames belong to “c”

“Decoder”:

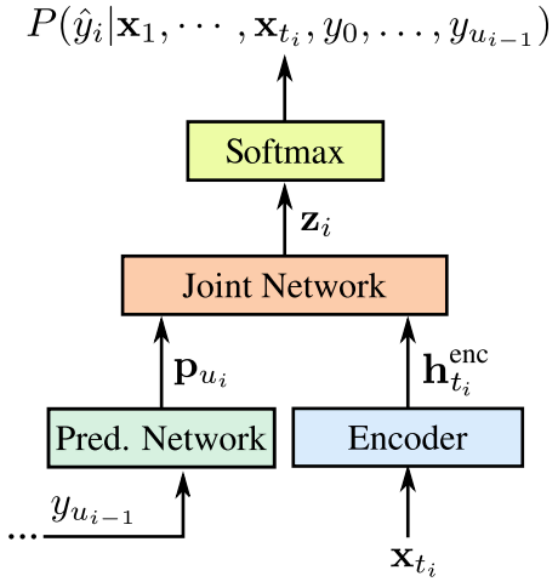
- Only attend on one vector
- Each output is decided independently

条件独立假设



模型简介：Sequence Labeling

- **RNN Transducer (RNN-T), 2012**
  - CTC的改进版，依旧是序列标注任务

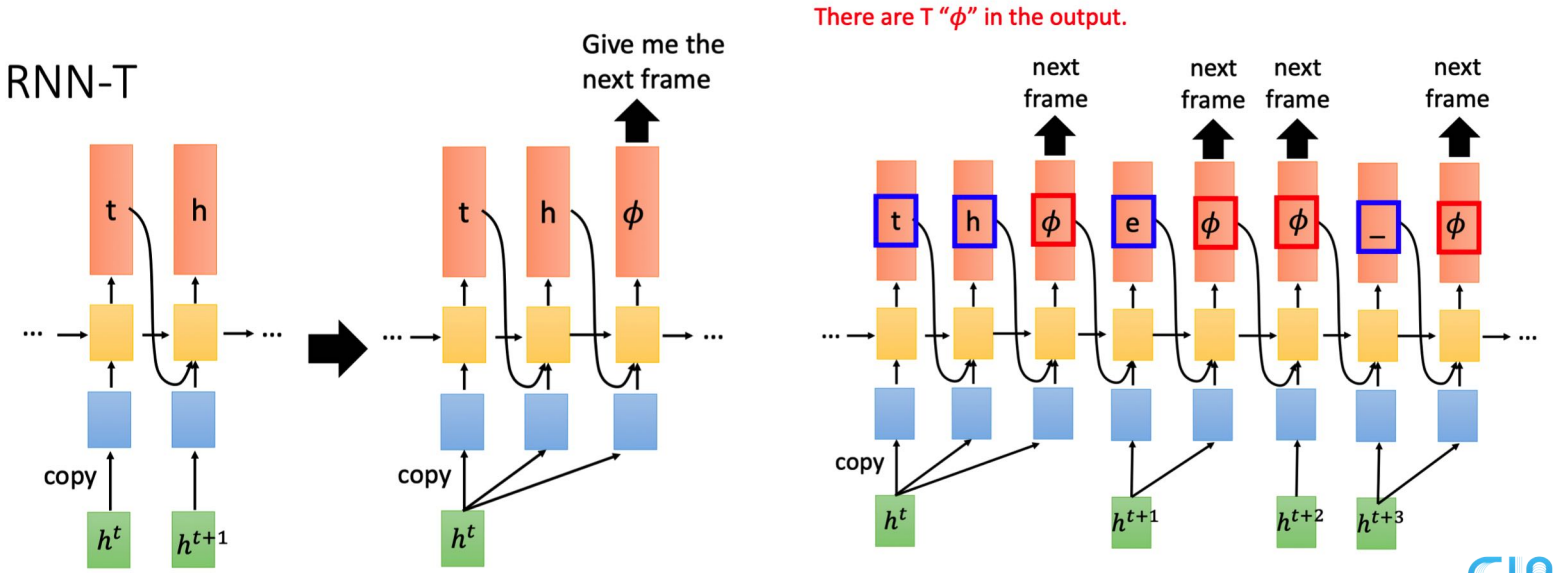


类似LM

类似CTC，声学模型

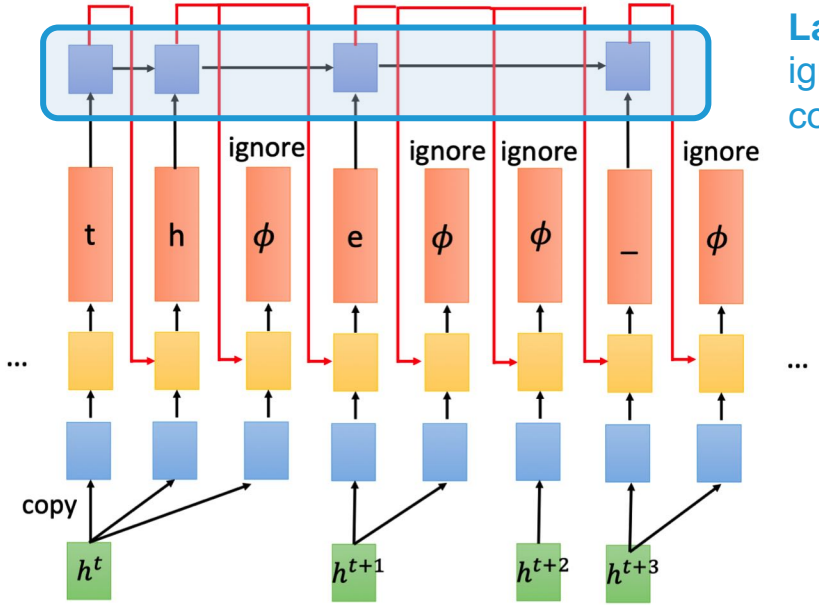
模型简介：Sequence Labeling

- **RNN Transducer (RNN-T)**
  - **Encoder/Transcription Network: 1-to-M**



模型简介：Sequence Labeling

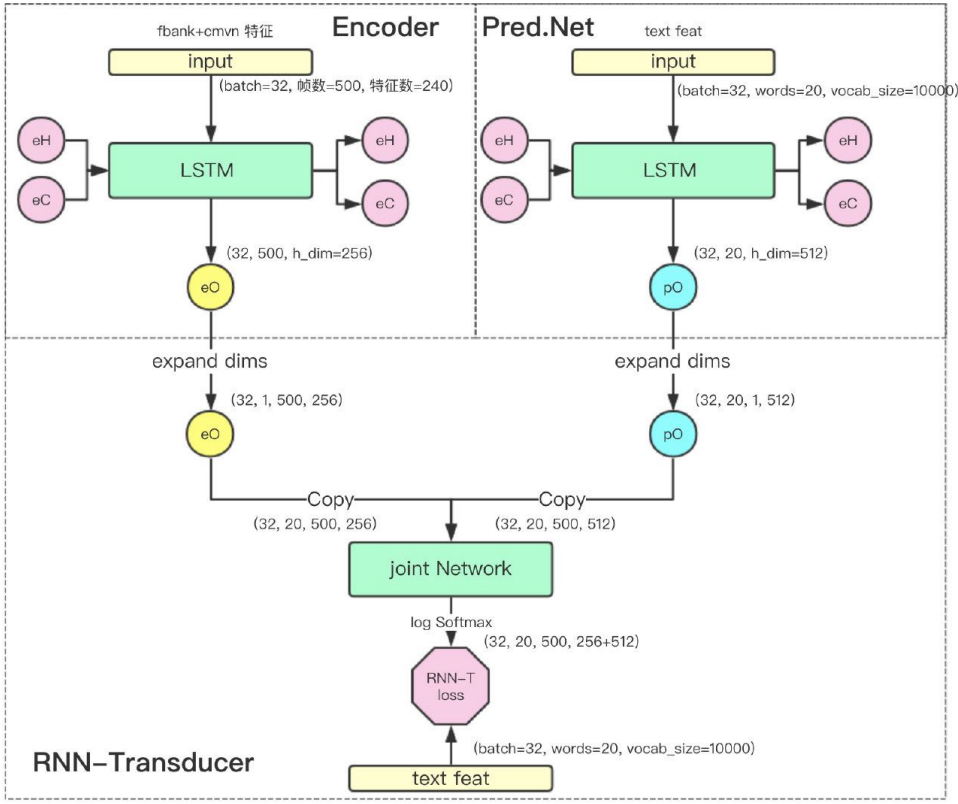
- **RNN Transducer (RNN-T)**
  - **Prediction Network**
    - 可单独训练



Language Model: ignore speech, only consider tokens

模型简介：Sequence Labeling

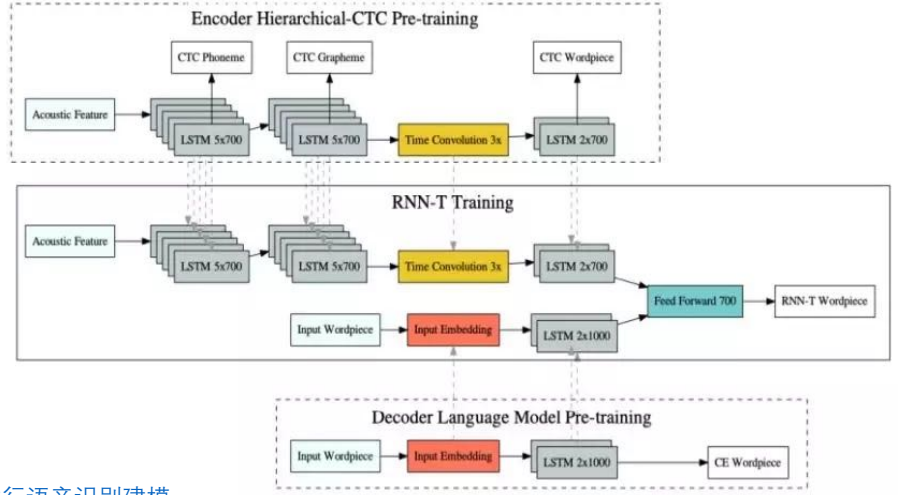
- **RNN Transducer (RNN-T)**



CTC,RNN-Transducer, LAS

模型简介：RNN-T

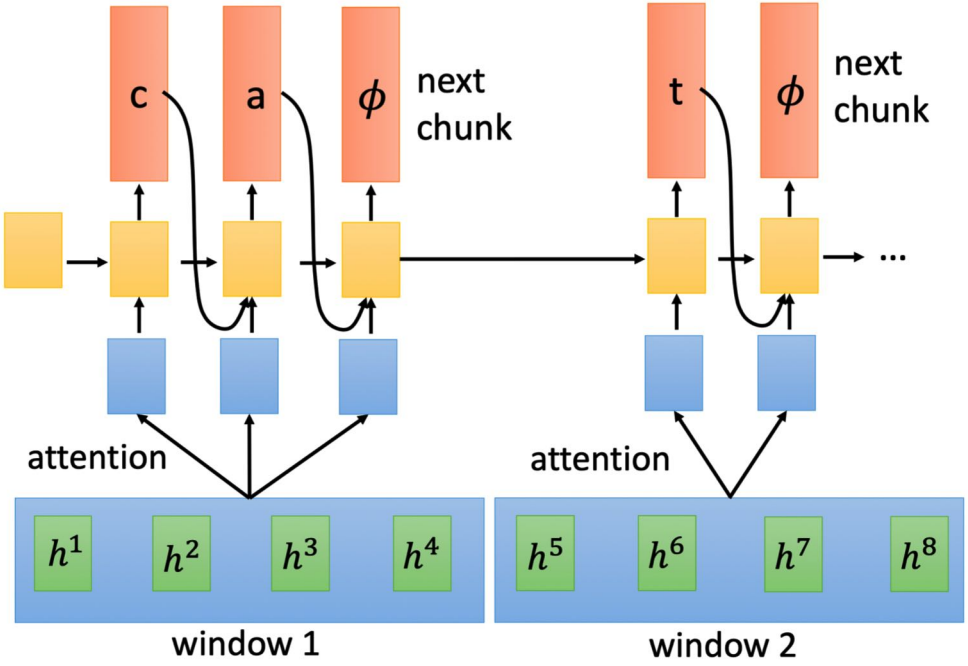
- 优势
  - 可以同时建模输入和输出的条件依赖（CTC 不依赖于输出）
  - 对输入输出的长度没有限制（CTC 的输入长度不能小于输出）
  - 支持流式
- 劣势
  - 训练困难，可先预训练两个子模型



使用RNN-Transducer进行语音识别建模

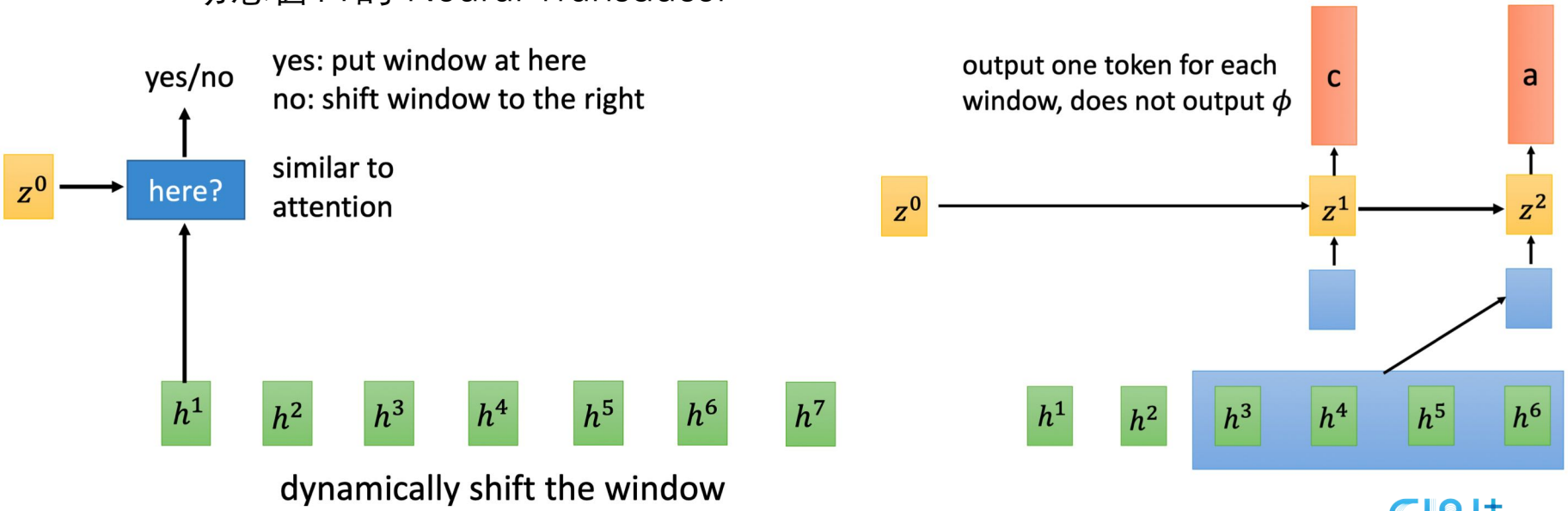
模型简介：Sequence Labeling

- Neural Transducer, 2016



模型简介：Sequence Labeling

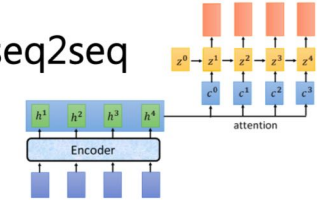
- **Monotonic Chunkwise Attention (MoChA), 2018**
  - 动态窗口的 Neural Transducer



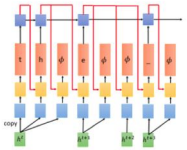
模型：总结

Summary

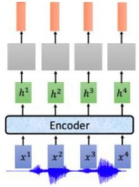
LAS: 就是 seq2seq



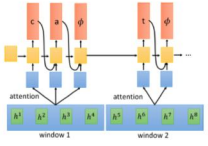
RNN-T: 輸入一個東西可以輸出多個東西的 seq2seq



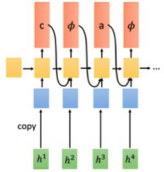
CTC: decoder 是 linear classifier 的 seq2seq



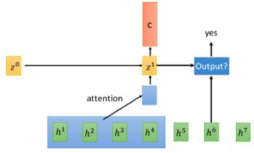
Neural Transducer: 每次輸入一個 window 的 RNN-T



RNA: 輸入一個東西就要輸出一個東西的 seq2seq

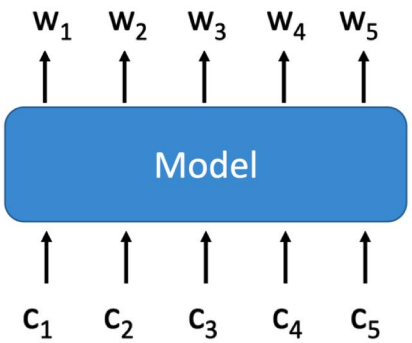


MoCha: window 移動伸縮自如的 Neural Transducer

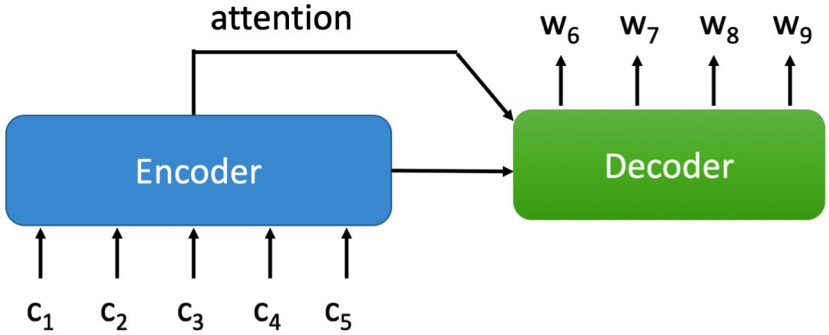


ASR

- 两种基本框架



- 易训练、高效
- 条件独立假设导致预测结果结巴重复

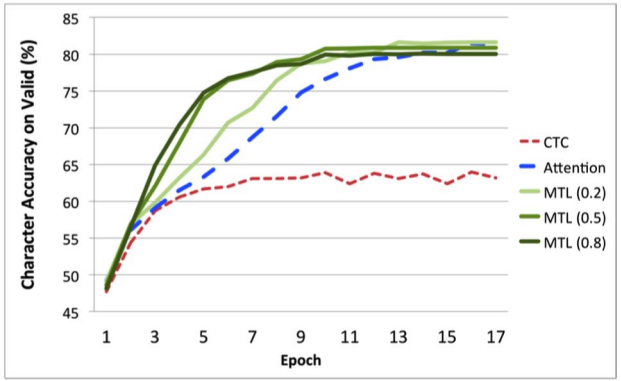


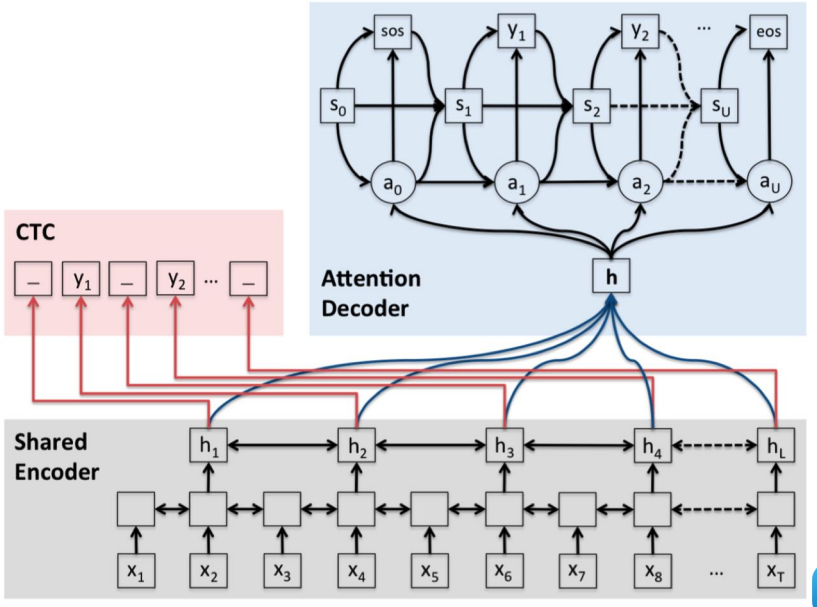
- CER更低，预测结果更灵活
- 对齐过于灵活导致训练困难，尤其对于长序列
- 容易被噪音干扰，实际预测可能效果更差

Joint of CTC and Attention

- 基于 Multi-Task Learning 的联合训练
  - 利用CTC限制对齐的灵活性，加速训练收敛速度

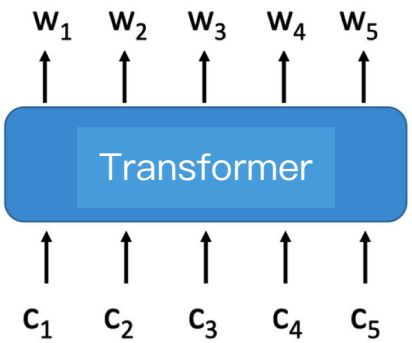
$$\mathcal{L}_{\text{MTL}} = \lambda \mathcal{L}_{\text{CTC}} + (1-\lambda)\mathcal{L}_{\text{Attention}}$$

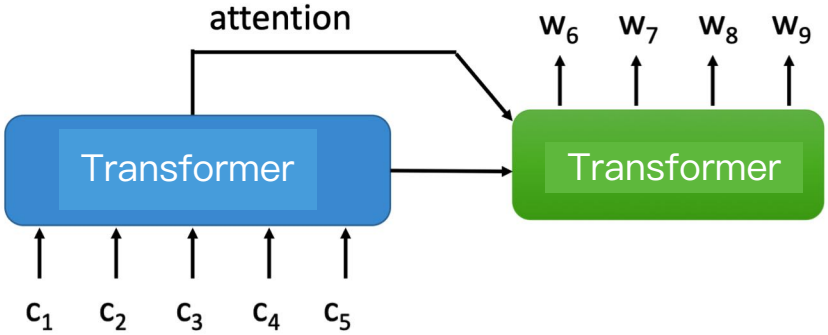




Transformer

- RNN → Transformer





Conformer Transducer

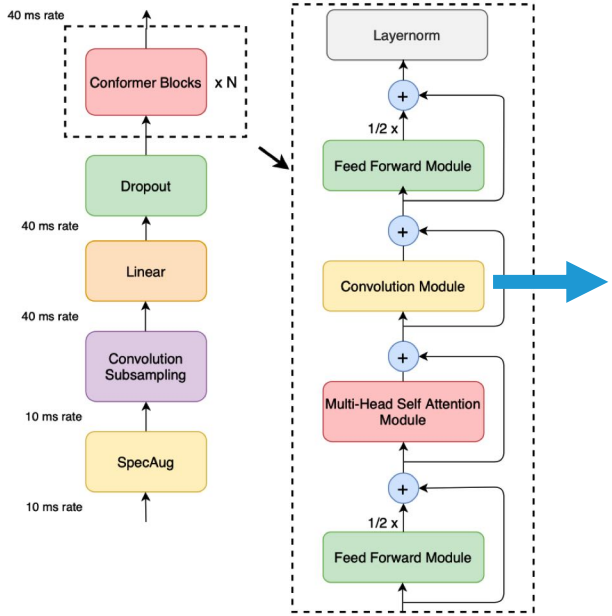
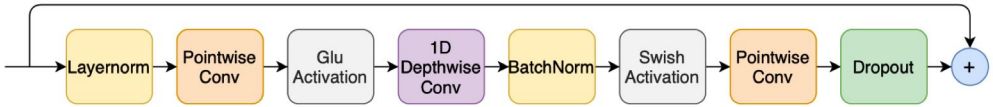


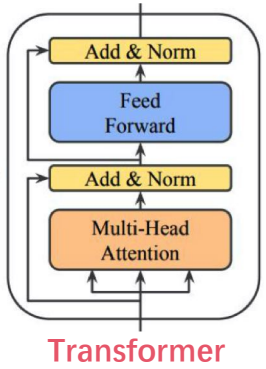
Figure 1: ***Conformer encoder model architecture.*** *Conformer comprises of two macaron-like feed-forward layers with half-step residual connections sandwiching the multi-headed self-attention and convolution modules. This is followed by a post layernorm.*

- CNN: local features
- Transformer: global features
- **Conformer**: CNN + Transformer



Conformer 变化：

- FFN → ½ FFN + ½ FFN
- + CNN



Transformer

Conformer Transducer

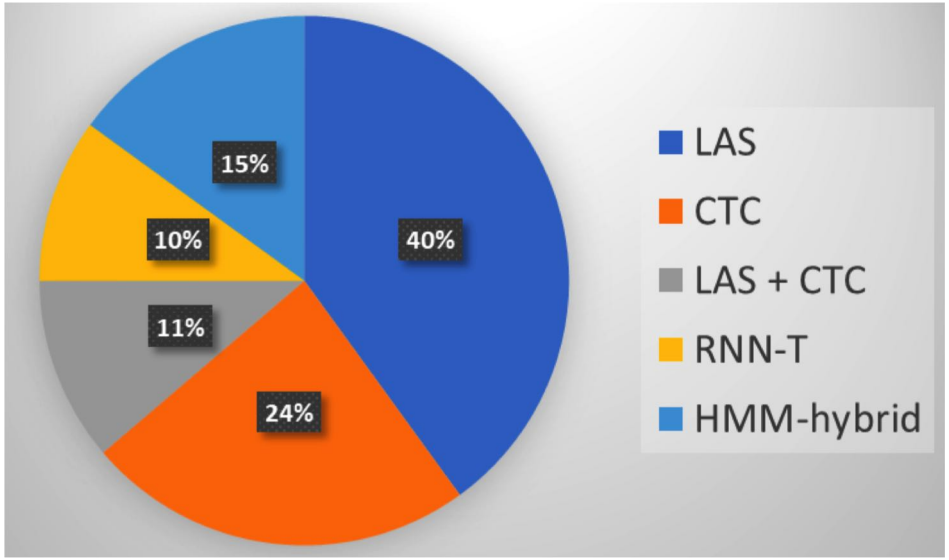
Method	#Params (M)	WER Without LM		WER With LM	
		testclean	testother	testclean	testother
Hybrid					
Transformer [33]	-	-	-	2.26	4.85
CTC					
QuartzNet [9]	19	3.90	11.28	2.69	7.25
LAS					
Transformer [34]	270	2.89	6.98	2.33	5.17
Transformer [19]	-	2.2	5.6	2.6	5.7
LSTM	360	2.6	6.0	2.2	5.2
Transducer					
Transformer [7]	139	2.4	5.6	2.0	4.6
ContextNet(S) [10]	10.8	2.9	7.0	2.3	5.5
ContextNet(M) [10]	31.4	2.4	5.4	**2.0**	4.5
ContextNet(L) [10]	112.7	**2.1**	4.6	**1.9**	4.1
Conformer (Ours)					
Conformer(S)	10.3	**2.7**	**6.3**	**2.1**	**5.0**
Conformer(M)	30.7	**2.3**	**5.0**	**2.0**	**4.3**
Conformer(L)	118.8	**2.1**	**4.3**	**1.9**	**3.9**

Table 3: ***Disentangling Conformer.*** *Starting from a Conformer block, we remove its features and move towards a vanilla Transformer block: (1) replacing SWISH with ReLU; (2) removing the convolution sub-block; (3) replacing the Macaron-style FFN pairs with a single FFN; (4) replacing self-attention with relative positional embedding [20] with a vanilla self-attention layer [6]. All ablation study results are evaluated without the external LM.*

Model Architecture	dev clean	dev other	test clean	test other
Conformer Model	1.9	4.4	2.1	4.3
– SWISH + ReLU	1.9	4.4	2.0	4.5
– **Convolution Block**	2.1	4.8	2.1	4.9
– Macaron FFN	2.1	5.1	2.1	5.0
– Relative Pos. Emb.	2.3	5.8	2.4	5.6

ASR：总结

- 模型框架流行度



- Go through more than 100 papers in INTERSPEECH'19, ICASSP'19, ASRU'19

ASR：总结

- Comparisons
  - **Performance**:

    Transformer > Joint C/A ~ RNN-T > ATT > CTC
  - **Training and inference speed**:

    CTC > ATT > Joint C/A ~ RNN-T > Transformer
    (transformer's training is fast)
  - **Online:**

    CTC, RNN-T > Joint C/A, ATT, Transformer
  - **Application:**

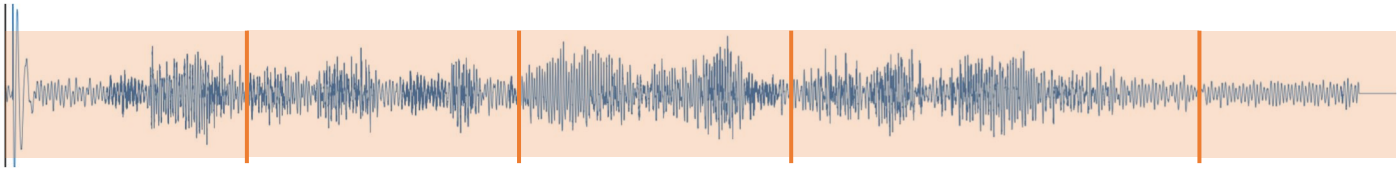
    ATT~Transformer~RNN-T > CTC, Joint C/A
  - **Easiness of implementation:**

    CTC > ATT > Transformer > Joint C/A > RNN-T

Streaming ASR

- 流式 ASR：尽快出结果，降低延时
- 实现流式的两种方式
  - 真流式

  - 伪流式



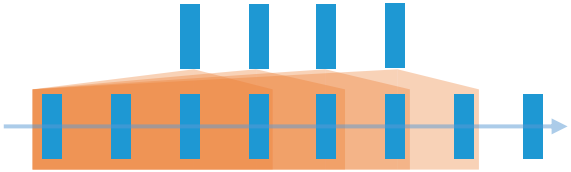
Streaming ASR

- 流式 ASR：尽快出结果，降低延时

- 模型
  - 天然可用：CTC、RNN-T
  - 需调整结构：LAS、Transformer

Streaming ASR

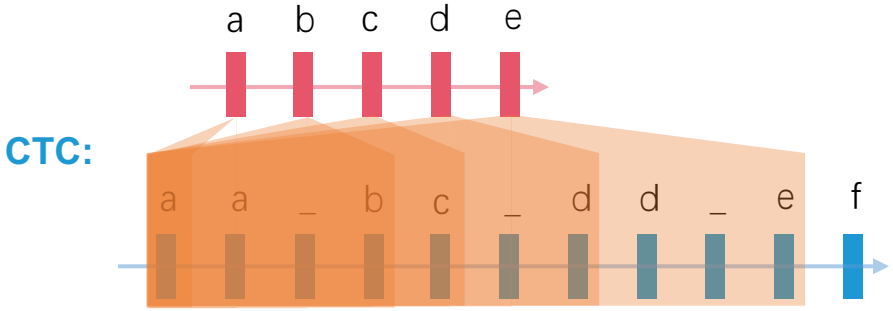
- Streaming ASR with Transformers
  - **time-restricted self-attention**



$$\boldsymbol{x}^{E}_{1:n} = \text{EncSA}^{\text{tr}}(\boldsymbol{x}^{0}_{1:n+\varepsilon^{\text{enc}}})$$

$$\boldsymbol{x}^{0}_{1:n+\varepsilon^{\text{enc}}} = X_0[1 : n + \varepsilon^{\text{enc}}] = (\boldsymbol{x}^{0}_{1}, \ldots, \boldsymbol{x}^{0}_{n+\varepsilon^{\text{enc}}})$$

  - **triggered attention** for the encoder-decoder attention mechanism



$$p_{\text{ta}}(Y|X_E) = \prod_{l=1}^{L} p(y_l|\boldsymbol{y}_{1:l-1}, \boldsymbol{x}^{E}_{1:\nu_l})$$

$$\nu_l = n'_l + \varepsilon^{\text{dec}}$$

- n'_l denotes the position of the first occurrence of label y_l in the CTC forced alignment sequence

Joint of ASR and EP/EOQ

- EP: EndPointer
- VAD: Voice Activity Detector
- EOQ: End Of Query

Joint of ASR and EP/EOQ

- 训练

- loss中加入端点偏移的惩罚，降低时延

$$\log \mathbf{P}_{\text{RNN-T}}(y_U|\mathbf{x}_t) -= \big(\max(0,\ \alpha_{\text{early}} * (t_{</s>} - t)) + \max(0,\ \alpha_{\text{late}} * (t - t_{</s>} - t_{\text{buffer}}))\big)$$

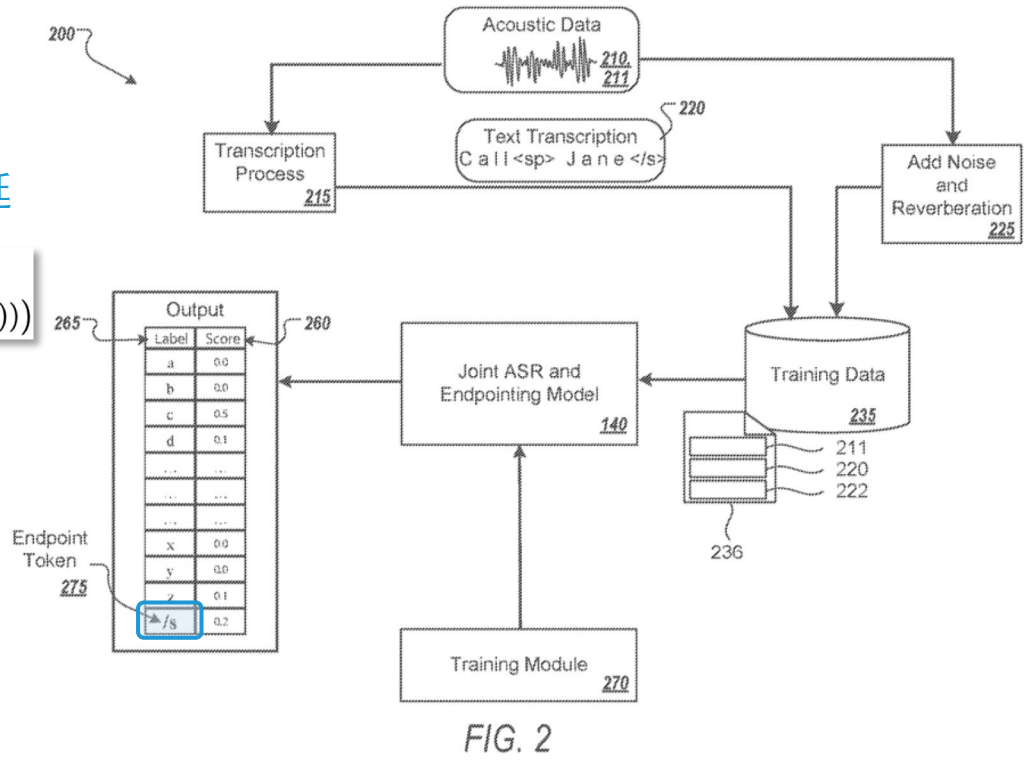


FIG. 2

Joint of ASR and SD

- SD: Speaker Diarization

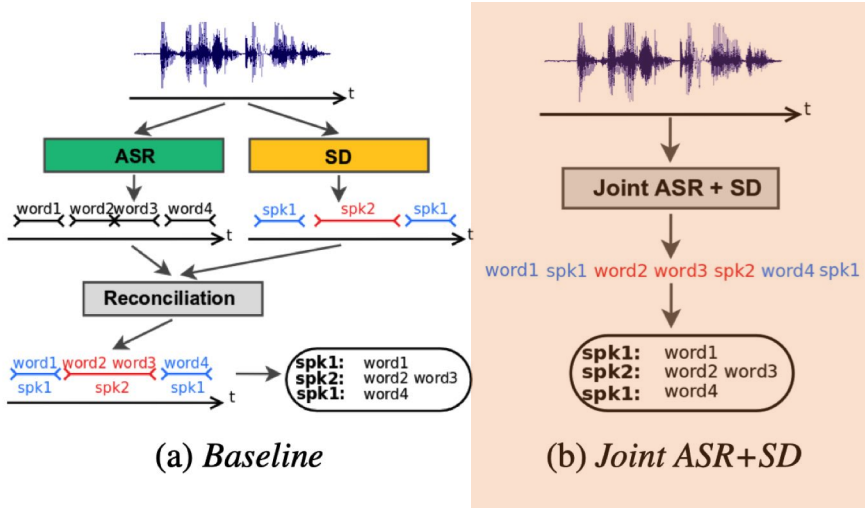


Figure 1: *Comparison of the conventional speech recognition and speaker diarization system (Figure 1a) with the proposed approach (Figure 1b), where the task consists of generating a speaker-decorated transcript from raw audio.*

hello dr jekyll <spk:pt> hello mr hyde what brings you here today <spk:dr> I am struggling again with my bipolar disorder <spk:pt>

Figure 2: *Example of an output sequence for our joint ASR and SD RNN-T system. The corresponding input would be the raw audio signal. Speaker turns are displayed in different colors.*

- SD标记作为额外的label加入到词典中
- 缺点：预先规定SD标记范围

常用工具

- Kaldi: C++
- Wav2letter++ : C++
- ESPnet: Python
- Jasper : Python
- MASR : Python

ESPnet

- **ESP**: **E**nd-to-end **S**peech **P**rocessing, **seq2seq** framework
- Python, PyTorch, Chainer; Bash

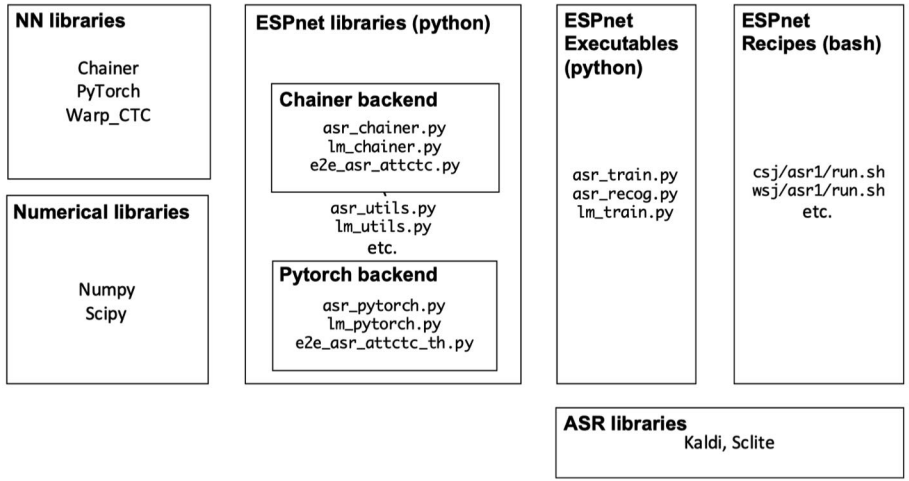
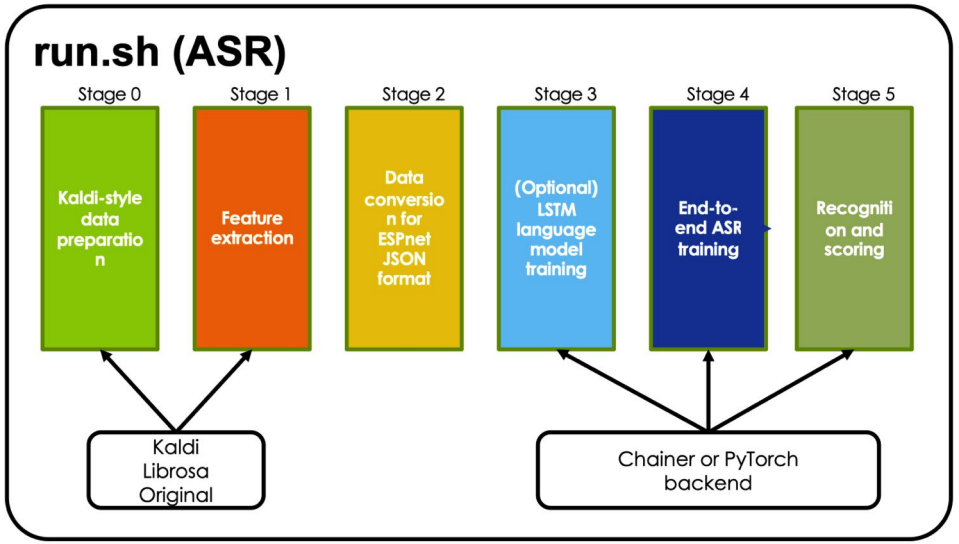


Figure 1: *Software architecture of ESPnet.*

ESPnet

- 处理流程类似Kaldi（bash ...）
- 应用场景 (**seq2seq**)：
  - ASR
  - TTS
  - ST: Speech Translation
  - VC: Voice Conversion



ESPnet

- Feature extraction
  - Kaldi → torchaudio
- Model
  - Conformer + CTC/Attention

References

- Deep Learning for Human Language Processing (2020, Spring)
- Conformer: Convolution-augmented Transformer for Speech Recognition
- Streaming ASR
  - Streaming automatic speech recognition with the transformer model
  - Towards Fast and Accurate Streaming End-to-End ASR
- Joint Speech Recognition and Speaker Diarization via Sequence Transduction
- ESPnet
  - ESPnet: End-to-End Speech Processing Toolkit
  - Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning
  - Hybrid CTC/Attention Architecture for End-to-End Speech Recognition

NLP/Speech

Thanks !

Your business is in good hands .