Google

Reducing Longform Errors in End2End Speech Recognition*

Liangliang Cao
http://llcao.net

*A similar version was presented at Rutgers University Efficient AI seminar, April 2022

Acknowledgement

The whole project would not succeed without my fantastic colleagues in Google Speech, Brain, and Cloud.

About myself (and why I did speech recognition)

0. M.Phil from CUHK mmlab, Ph.D. from UIUC, both in computer vision
1. Co-founded a small startup (Switi Inc) which was acquired by Google 2018
2. Was asked to work on Speech 2019-2021 after joining Google
    - Speech TL for Goolge-Verizon Contact Center AI (“biggest contract in the history of Goolge Cloud AI”)
    - Built a team to lead GCloud migrate to end2end speech recognition (error rate reduced 50% in 2 years, 10+ launches on Google Cloud)
3. Then back to computer vision since 2022
    - TL to launch Google Cloud CoCa embedding
    - Joined Apple in Nov. 2022, and then worked on visual-language/LLM/3D synthesis

About myself (and why I did speech recognition)

0. Master from CUHK mmlab in 2005, Ph.D. from UIUC 2011
1. Co-founded a small startup (Switi Inc) which was acquired by Google 2018
2. **Was asked to work on Speech 2019-2021 after joining Google** ⇦ **Today's talk**
   - Speech TL for Goolge-Verizon Contact Center AI ("biggest contract in the history of Goolge Cloud AI")
   - Built a team to lead GCloud migrate to end2end speech recognition (error rate reduced 50% in 2 years, 10+ launches on Google Cloud)
3. Then back to computer vision since 2022
   - TL to launch Google Cloud CoCa embedding
   - Joined Apple in Nov. 2022, and then worked on visual-language/LLM/3D synthesis

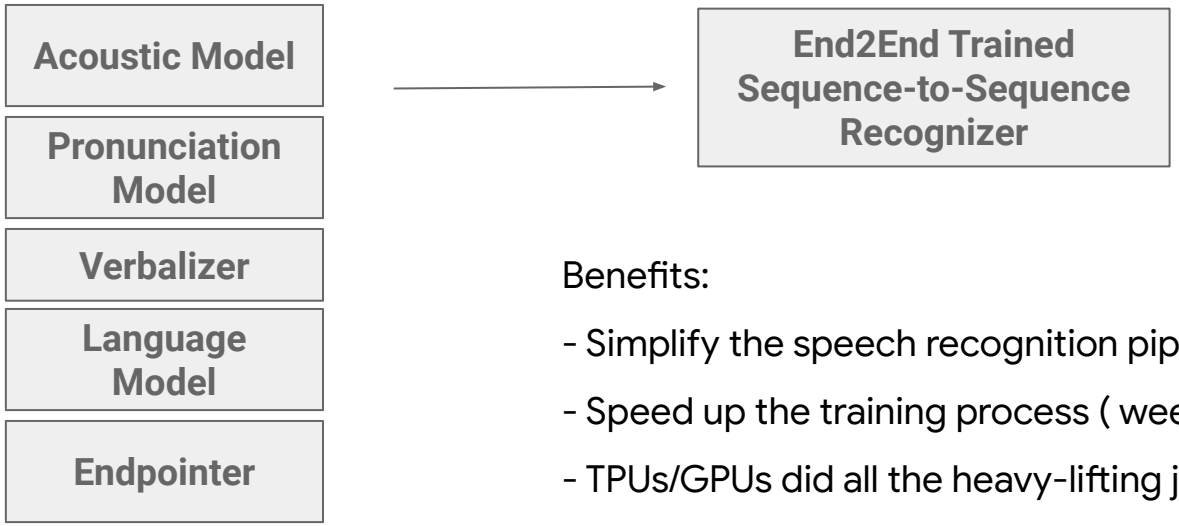
Outline

- End2End speech recognition and longform errors
- Basic solution: distillation on Youtube data
- Universal perturbation to simulate longform errors
- Enhanced solution: learning from stronger teachers
- Conclusion

End-to-End ASR and Longform Errors

From conventional to end2end speech recognition

Conventional Speech System



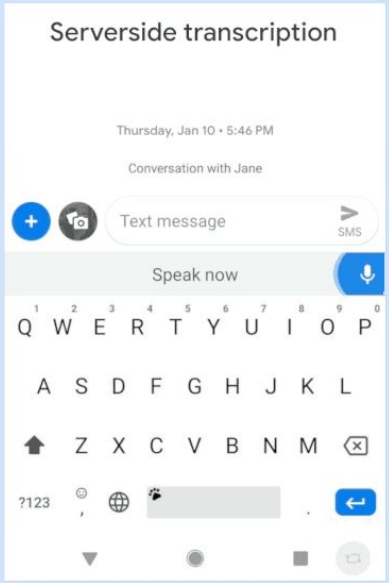
Benefits:

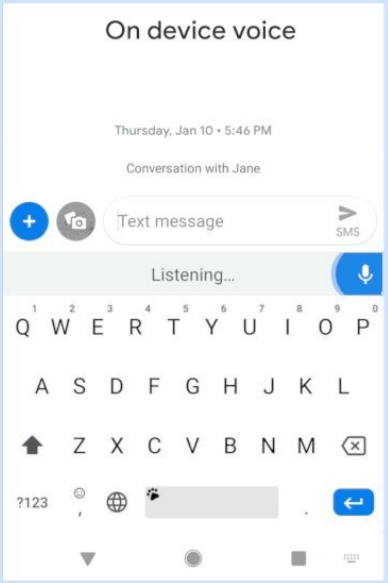
- Simplify the speech recognition pipeline
- Speed up the training process (weeks -> days)
- TPUs/GPUs did all the heavy-lifting jobs
- Effectively learn from large scale training data
- Some end2end models (such as RNN-T) is 10x smaller than the conventional models!

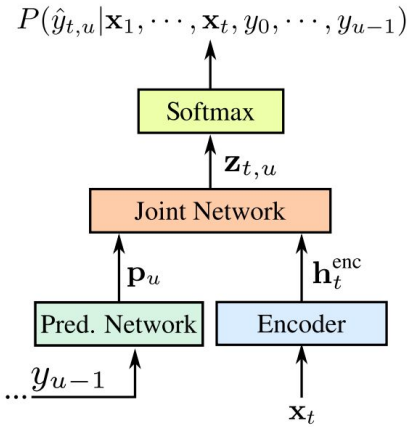
Other end2end learning examples:
- AlexNet (end2end image classification)
- DETR (end2end object detection)
- Fast R-CNN (not end2end)

RNN-T for streaming ASR

RNN-Transducer has been *de facto* model for Google end2end ASR (official blog), for streaming applications on both devices and servers.



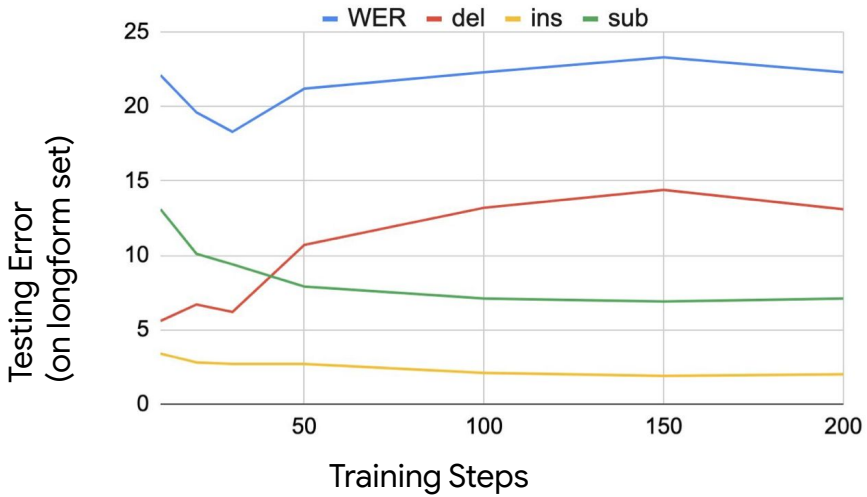




RNN-Transducer

But RNN-T may suffer from high deletion errors on longform audios (1)

The more learn steps, the higher deletion error becomes!



Why does it happen?

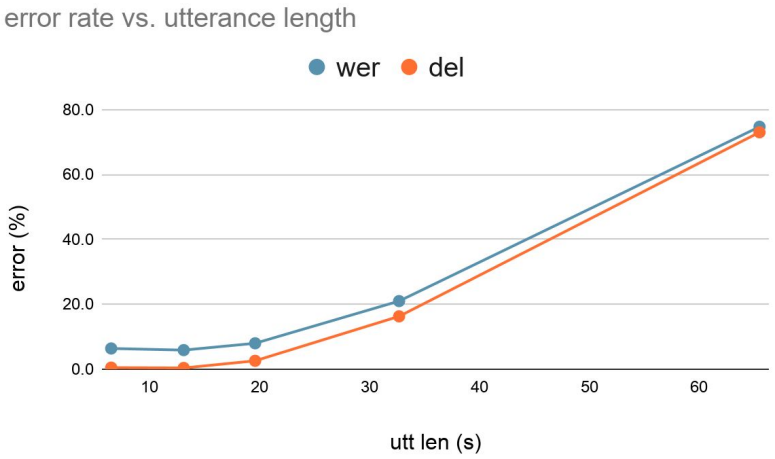
- The TPU/GPU memory is limited, so the training utterances are shorter than those in real testing set.
- The <empty> hypothesis dominates the beam search on long form audios. Both LSTM encoder and conformer encoder suffer from this.

From SLT 2021: “RNN-T Models Fail to Generalize to Out-of-Domain Audio: Causes and Solutions” (with Chung-Cheng Chiu et al)

But RNN-T may suffer from high deletion errors on longform audios (2)

Conformer model on some concatenated librispeech test-other auidos.

# concatenation	# seconds	WER (del/ins/sub)
1 (original)	6.5	6.4 (0.5/0.8/5.1)
3	19.6	8.0 (**2.6**/0.7/4.7)
5	32.7	21.0 (**16.3**/0.6/4.1)
10	65.5	74.7 (**73.0**/0.2/1.4)



Interspeech 2021: “Exploring Targeted Universal Adversarial Perturbations to End-to-end ASR Models” (with Zhiyun Lu et al)

Basic Solution: teacher distillation on Youtube data

Which model suffers less from longform errors?

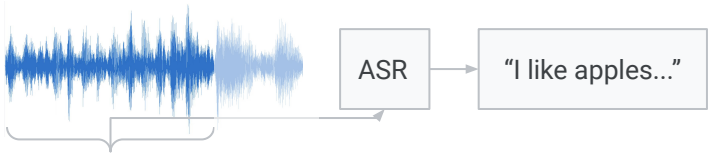
Streaming RNN-T may easily suffer from long form deletion errors

But we can train bigger non-streaming RNN-T** with less deletion

- Access signals from both past and the future
- Use bi-directional LSTM or non-causal transformer/conformer
- Can also use overlapping inference on long audios
- Limitation: Non-streaming, not supported by production
    - But we can use them as teachers!

** details at SLT 2021: “RNN-T Models Fail to Generalize to Out-of-Domain Audio: Causes and Solutions”, (with Chung-Cheng Chiu et al)

Non-streaming ASR

	Non-streaming models	Streaming models
Context		
Considerations	• Have access to full context before processing the audio. • Performs better than streaming models. • Less user-friendly.	• Must produce words on-the-fly. • Does not have access to future context.
Inference	• Overlapping inference • Do not care latency (so big models are OK)	• Beam search • Low latency (say <100ms)

Distill non-streaming teacher

Given a strong non-streaming teacher

1. Gather unlabeled utterances from YouTube.
2. Randomly segment utterances, between 5-15 seconds.
3. Label the resulting utterances using the teacher model.
4. Train a streaming RNN-T with pseudo labels using SpecAug

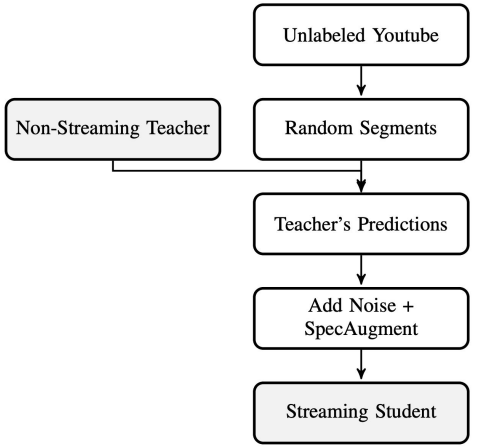


Fig. 1. Our method trains a streaming model, learning from the predictions of a powerful non-streaming teacher model on large-scale unlabeled data via a teacher-student training framework. See Sect. 2.2 for more details.

ICASSP 2021: "Improving streaming automatic speech recognition with non-streaming model distillation on unsupervised data" (with Thibault Doutre and Wei Han et al)

Experiment setup

Compare two sources of training data:

- **Confisland**: YouTube data aligned user-uploaded transcripts [13]
- **YT-segments**: Unsupervised segments from original audios corresponding to Confisland

	Confisland	*YT-segments*
Spanish	13,000	41,000
French	10,000	29,000
Portugese	2,500	5,000

Test data:

- **YT-long**: long utterances from had-transcribed YouTube videos

[13] Hank Liao, Erik McDermott, and Andrew Senior, “Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription,” in ASRU 2013, pp. 368–373.

Teacher model:

- Slow, non-streaming RNN-T with bi-directional LSTM encoder.

Student model:

- Exactly the same configuration as the streaming RNN-T in Production.

Experimental results

Teacher transcribed YT-segments are much better than confisland!

Significant improvement on YT-long, common voice and Cloud testing sets (not shown) across four languages.

Table 4. Comparing the WERs of streaming RNN-T models trained on *Confisland* with the model from our distillation approach trained on the corresponding random segments.

	Test set	Streaming model on *Confisland*	Streaming student on *YT-segments*
French	YT-long	34.5	25.0
	Common Voice	36.2	34.7
Spanish	YT-long	35.9	28.0
	Common Voice	22.0	16.5
Portuguese	YT-long	30.8	28.3
	Common Voice	30.9	28.9
Italian	YT-long	24.0	20.8
	Common Voice	30.0	23.6

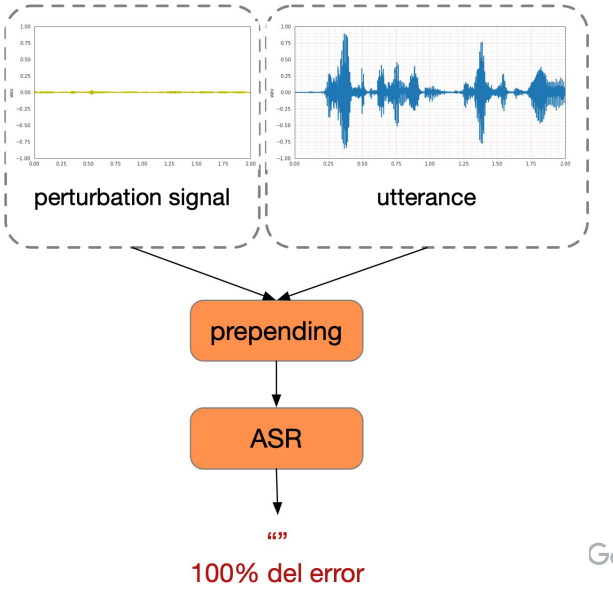
More details in ICASSP 2021: Improving streaming automatic speech recognition with non-streaming model distillation on unsupervised data (with Thibault Doutre and Wei Han et al)

Targeted universal perturbation to simulate longform errors

Can we intentionally create deletion error?

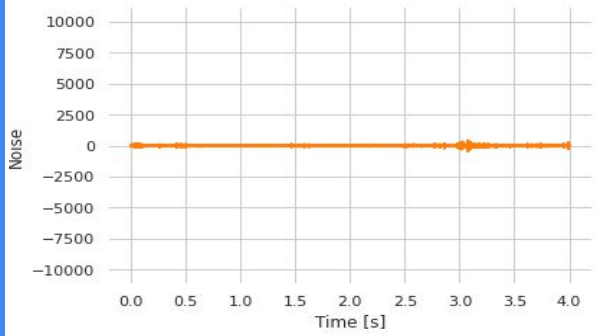
Next we will show that we can learn a magic 4 second audio from Librispeech train. When we append such audio to the beginning of unseen audios from Librispeech test and test-others, the model will generate empty prediction (100% deletion error)

Goal: **one perturbation per model for all utterances!**



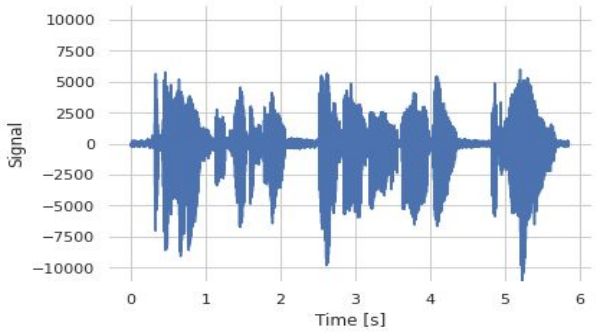
Interspeech 2021: Exploring Targeted Universal Adversarial Perturbations to End-to-end ASR Models, (with Zhiyun Lu, Wei Han, Yu Zhang)

Problem statement



δ

$\oplus$



$\boldsymbol{x}$

ASR

θ

“”

y'

x	an audio from some $\mathcal{D}$
$\mathcal{T}_\delta(\boldsymbol{x})$	append δ to x
y’	the mis-transcription that we want to generate
$\ell(\mathcal{T}_\delta(\boldsymbol{x}), y'; \theta)$	ASR loss with the targeted y’

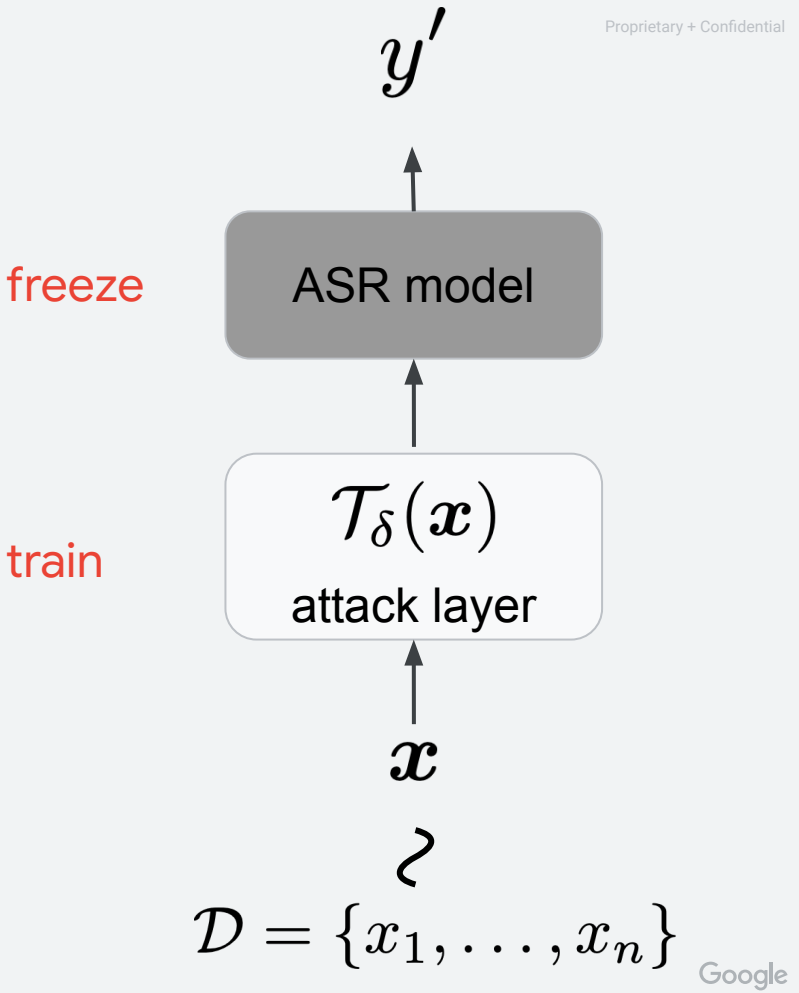
Learning the universal perturbation

$$\min_{\delta} \sum_{\boldsymbol{x} \in \mathcal{D}} \ell(\mathcal{T}_\delta(\boldsymbol{x}), y'; \theta)$$

cf. normal model training

$$\min_{\theta} \sum_{\boldsymbol{x} \in \mathcal{D}} \ell(\theta; \boldsymbol{x}, y)$$

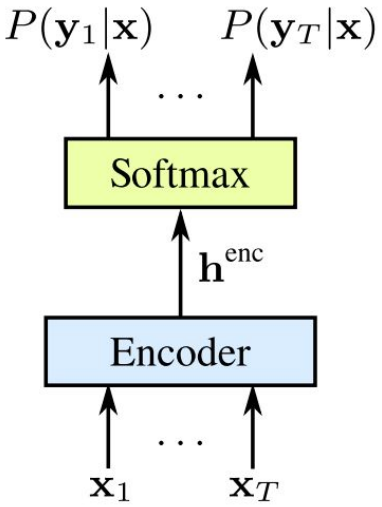
Note: Our experiment only applies for Librispeech models, but NOT Google's production models (for latter we cannot compute gradient with a non-differentiable frontend)



Attack different end2end ASR models

- CTC (Connectionist Temporal Classification)
- Listen Attend and Spell (LAS)
- RNN-Transducer (RNN-T) with conformer encoder

Connectionist Temporal Classification (CTC)

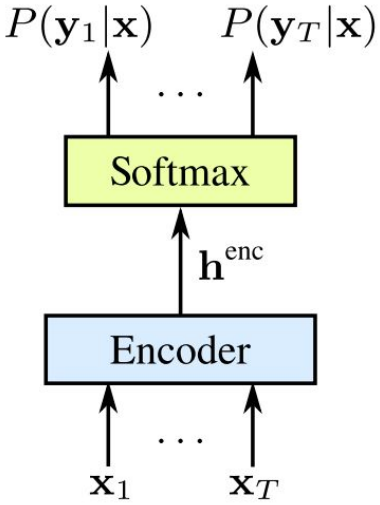


References:
- Alex Graves, Navdeep Jaitly, Towards End-To-End Speech Recognition with Recurrent Neural Networks, 2014
- Dario Amodei et al, DeepSpeech2, 2015

Key Takeaway

CTC allows for training an acoustic model without the need for frame-level alignments between the acoustics and the transcripts.

Connectionist Temporal Classification (CTC)



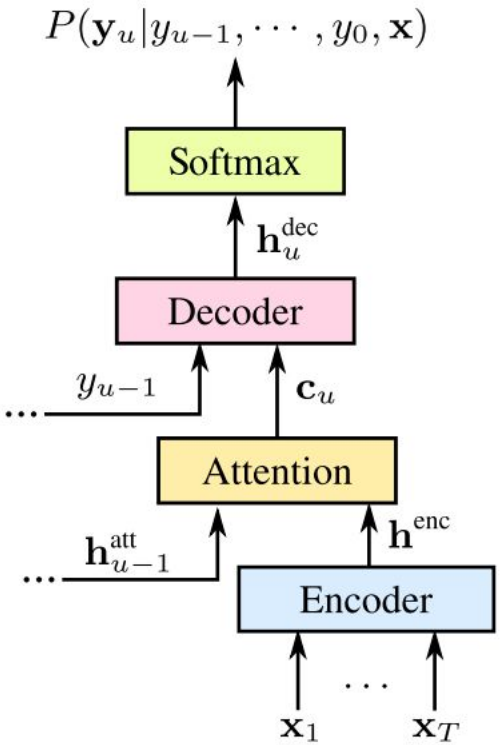
B	B	**c**	B	B	**a**	**a**	B	B	**t**
B	**c**	**c**	B	**a**	B	B	B	B	**t**
				...					
B	**c**	B	B	**a**	B	B	**t**	**t**	B

$$P(\mathbf{y}|\mathbf{x}) = \sum_{\hat{\mathbf{y}} \in \mathcal{B}(\mathbf{y},\mathbf{x})} \prod_{t=1}^{T} P(\hat{y}_t|\mathbf{x})$$

Key Takeaway

CTC introduces a special symbol - blank (denoted by B) - and maximizes the total probability of the label sequence by marginalizing over all possible alignments

Listen, Attend and Spell (LAS)

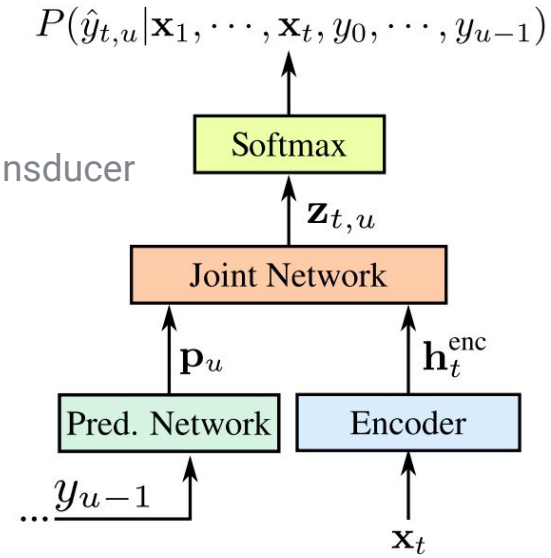


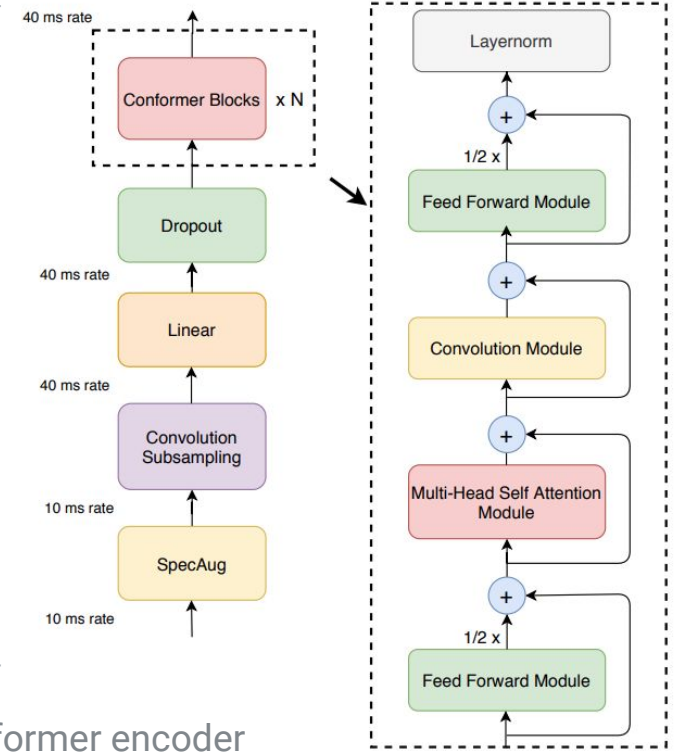
- Encoder (analogous to AM):
  - Transforms input speech into higher-level representation
- Attention (alignment model):
  - Computes a similarity score between the decoder and each frame of the encoder
  - Identifies encoded frames that are relevant to producing current output
- Decoder (analogous to PM, LM):
  - Operates autoregressively by predicting each output token as a function of the previous predictions

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, “Listen, Attend, and Spell”, ICASSP 2016

RNN-T with conformer encoder

RNN-Transducer





Conformer encoder

References:

Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton, Speech Recognition with Deep Recurrent Neural Networks, 2013

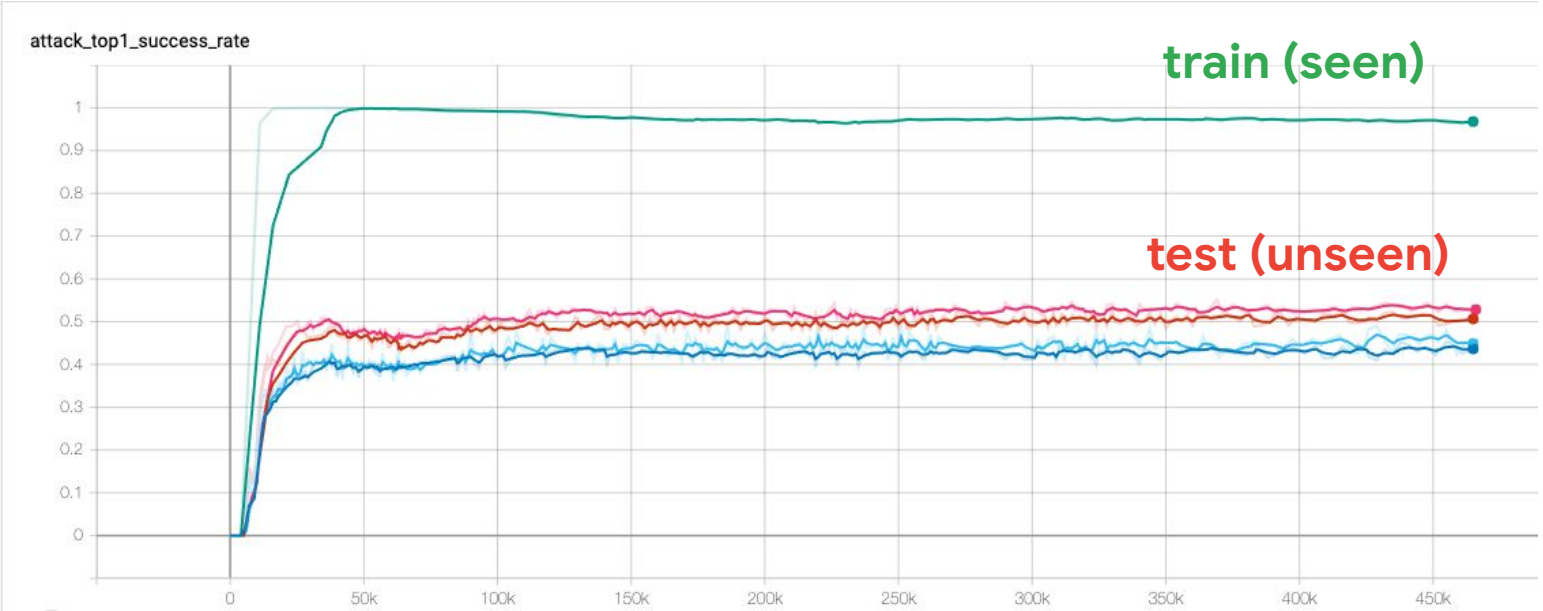
Gulati et al, “Conformer: Convolution-augmented Transformer for Speech Recognition”, Interspeech 2020

Experiment setup

- dataset: Librispeech
  - train on 960h
  - report on test-clean (2620 audio), test-others (2939 audio)
- evaluation metrics
  - success rate: $\frac{\#(\text{utt outputs} = y')}{\#(\text{utt})}$
  - dB: measure distortion (loudness)

$$D(\delta, \boldsymbol{x}) = \mathrm{dB}(\delta) - \mathrm{dB}(\boldsymbol{x}), \quad \mathrm{dB}(\boldsymbol{x}) = 20 \log_{10}(\max_i(\boldsymbol{x}_i))$$

Success rates during the learning process



y’ = ""
prepend noise

Listen to the adversarial perturbation (Conformer-Transducer)

Using models trained from public Librispeech and an unseen data

Fool another model to predict “ ” on the unseen testing set.

universal perturbation
(4 seconds)

prediction “”

transcript_truth
a cold lucid indifference reigned in his soul

prediction “”

transcript_truth
he hoped there would be stew for dinner turnips and carrots and bruised potatoes and fat mutton pieces to be ladled out in thick peppered flour fattened sauce

Listen to the adversarial perturbation (Conformer-LAS)

Using models trained from public Librispeech and an unseen data

Fool the model to predict “ ” on all utterances in Librispeech test sets.

universal perturbation (4 seconds)

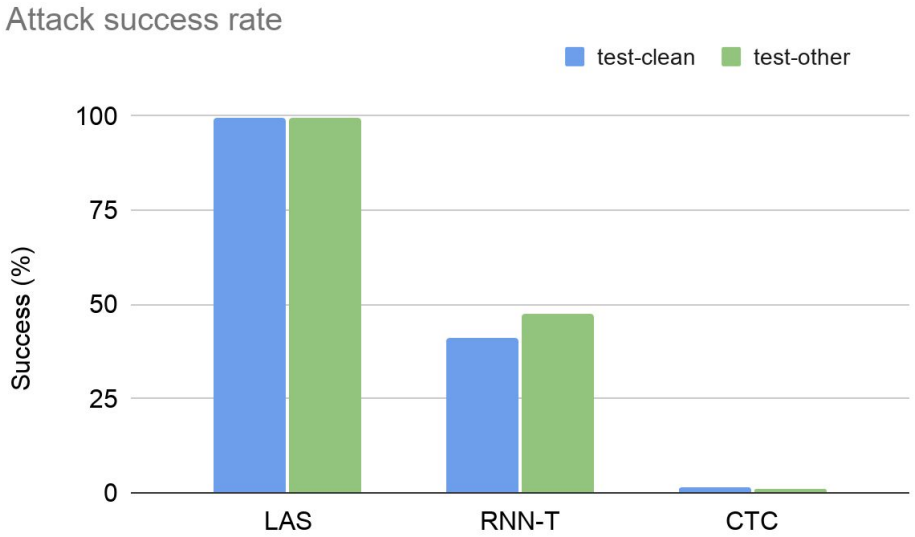
prediction “”

transcript_truth
a cold lucid indifference reigned in his soul

prediction “”

transcript_truth
he hoped there would be stew for dinner turnips and carrots and bruised potatoes and fat mutton pieces to be ladled out in thick peppered flour fattened sauce

Attack easiness: LAS > RNN-T > CTC



y’ = ""
prepend noise

What does this experiment tell us?

- It does **not** mean we can attack Google ASR systems (which uses a non-differentiable frontend).
- It does suggest CTC model is more robust against long form deletion errors.
- We do not want to use CTC in production because RNN-T has better supports with lower WERs on short forms.
- But we can combine CTC and RNN-T for a stronger teacher!

Enhanced solution: learning from stronger teachers

Internspeech 2021: “Bridging the gap between streaming and non-streaming ASR systems by distilling ensembles of CTC and RNN-T models” (with Thibault Doutre et al)

Expand to multiple teachers

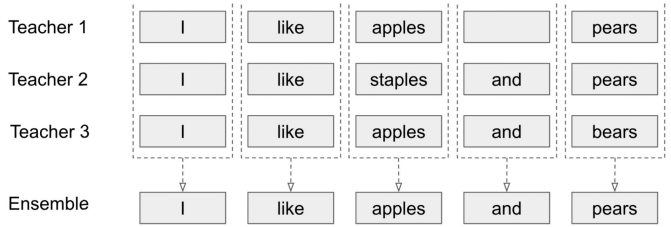
Non-streaming teacher models

We use 3 different teacher models, trained on various types of data.

	Encoder	Decoder	Data
MD-RNNT	17 blocks	1 LSTM	Multi-domain
YT-RNNT	16 blocks	1 LSTM	YouTube
YT-CTC	16 blocks	1 layer	YouTube

Teacher ensemble

Predictions of multiple teacher models are ensemble using **Recognizer Output Voting Error Reduction (ROVER)**.



Results

- The teacher ensemble outperforms all teachers separately
- Student models trained from the teacher ensemble are better

Table 3: *WERs of a streaming Conformer student model trained on YT-segments, distilled from non-streaming teacher models.*

	Teacher model	Teacher WER on *YT-long*	Student WER on *YT-long*
Spanish	MD-RNNT	16.4	33.4
	YT-RNNT	18.6	23.4
	YT-CTC	20.2	16.9
	Teacher ensemble	18.1	16.4
Portuguese	MD-RNNT	29.1	31.9
	YT-RNNT	22.8	26.7
	YT-CTC	24.8	23.0
	Teacher ensemble	21.9	20.5
French	MD-RNNT	31.9	42.8
	YT-RNNT	18.8	23.6
	YT-CTC	21.0	16.6
	Teacher ensemble	20.2	16.7

CTC vs RNN-T teachers

The paradox of CTC teachers

- CTC models have a higher WER than RNN-T teachers
- CTC transcripts suffer from linguistic issues
- On long-form test sets, **RNN-T students trained on CTC models outperform their counterparts** trained on RNN-T teachers.

Key findings from ablation studies

- Using **at least 1 CTC teacher** leads to lower student WER
- **Combining** CTC and RNN-T teachers give best results
- RNN-T student models outperform their CTC teachers

Improvement over previous study

- CTC teacher may not outperform RNN-T teacher
- But the resulted student from CTC is stronger!

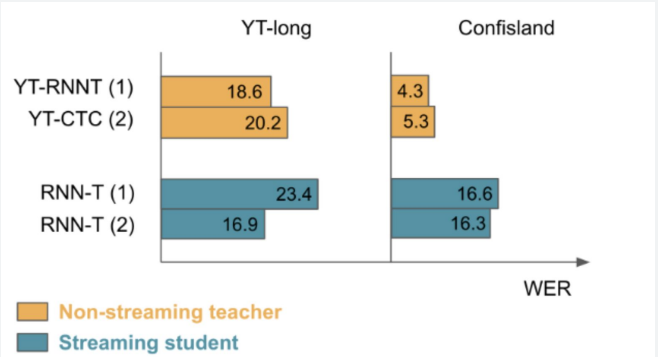


Figure 4: *Comparing a RNN-T streaming student model trained from CTC and RNN-T non-streaming teachers. The CTC model is a better teacher, despite having a higher WER.*

Summary of the improvement (built a team from scratch, over 1.5 years)

WERs on YT-long dataset

	Baseline (before our team start)	ICASSP 2021	Interspeech 2021
French	34.5	25.0	**16.7**
Spanish	35.9	28.0	**16.4**
Portuguese	30.8	28.3	**20.5**

And significant improvement on Cloud benchmarks with 10+ launches.

Related papers

1. “Improving Streaming Automatic Speech Recognition With Non-Streaming Model Distillation On Unsupervised Data”, ICASSP 2021. (with Thibault Doutre, Wei Han et al).
2. “Exploring Targeted Universal Adversarial Perturbations to End-to-end ASR Models” Interspeech 2021. (with Zhiyun Lu, Wei Han, Yu Zhang).
3. “Bridging the gap between streaming and non-streaming ASR systems by distilling ensembles of CTC and RNN-T models”, Interspeech 2021. (with Thibal Doutre et al)

Other related paper

"BigSSL: Exploring the Frontier of Large-Scale Semi-Supervised Learning for Automatic Speech Recognition" (with Yu Zhang, Daniel S. Park, Wei Han, et al)

- 8B transformer pre-trained + self-trained on 1M hour audio

“Universal Speech Model: Scaling Automatic Speech Recognition Beyond 100 Languages” (Yu Zhang, Wei Han, James Qin, Yongqiang Wang et al)

- Best-RQ pretraining + 2B transformer with chunk-wise attention for 100+ langs
- Google’s Whiper but with much lower error rate

Conclusion

Foundation models as a teacher to improve ML products:

- Improving student models' robustness and generalizability
- Alleviating annotation errors
- No need to change the existing models in production

A few lessons I learned

- Data + infra + algorithm
- It is quite fun and helpful to study error patterns (e.g. universal adversarial attacks)

Thank you for your attention!

Backup slides

Deletion errors in beam search

Beam search step 200	Beam search step 280	Beam search step 299
hey guys i don't know a	*<empty>*	*<empty>*
<empty>	*hey guys i don't know a lot of work yes oh you're*	*subscribe*
oh hey guys i don't know a	*hey guys i don't know a lot of work yes oh*	*yeah*
hey hey guys i don't know a	*hey guys i don't know a lot of work yes*	*h*
hey guys i don't know	*subscribe*	*this*
uh hey guys i don't know a	*hey guys i don't know a lot of work yes please*	*uh*
oh hey guys i don't know a	*hey guys i don't know a lot of work yes subscribe*	*hmm*
hey guys i don't know a a	*hey guys i don't know a lot of work yes for*	*is*

Fig. 4: An example of beam search steps for one of the utterance in *YT-long* that exhibit high deletion errors. The model is able to recognize the utterance at the beginning, but as the beam search proceeds the <*empty*> hypothesis starts to dominate and eventually fill the beam with hypotheses branched from the <*empty*> hypothesis. Once the beam is filled with these hypotheses, the beam search process can no longer recover and result a final hypothesis with high deletion errors.

Overlapping inference

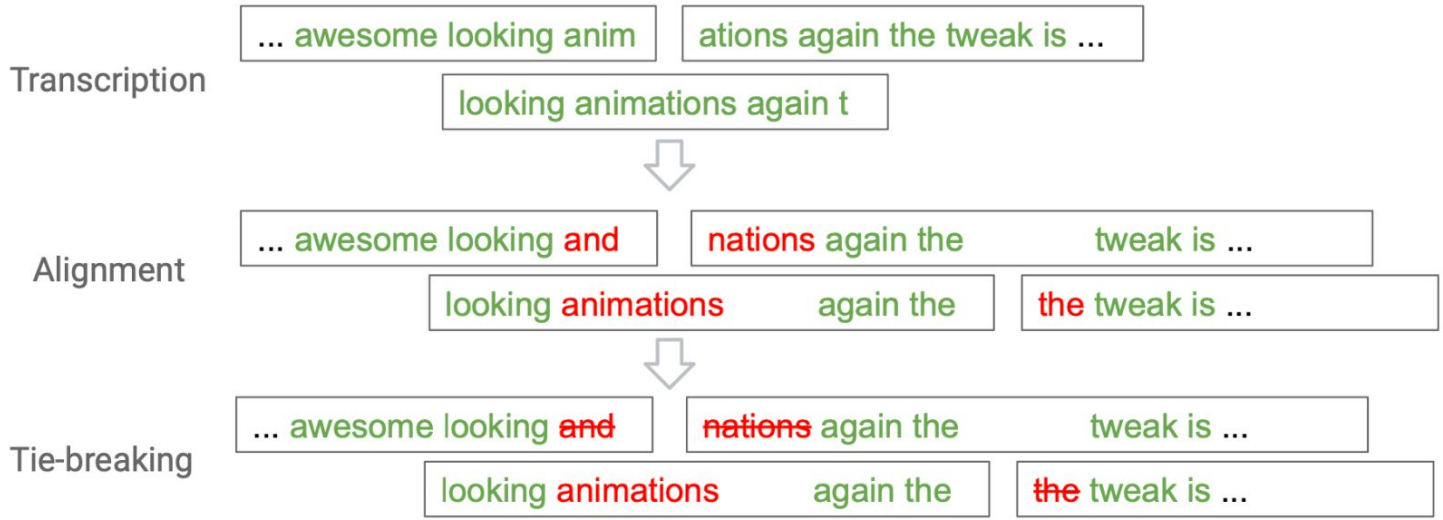


Fig. 2: Overlapping inference. The algorithm first breaks a long utterance into overlapped segments, each of which is then transcribed independently. It then merges the transcripts from overlapped segments into a consensus transcript for the original long utterance. In case there are conflicts in predictions, it prefers the predictions further from the utterance boundary.

ASRU 2019: “A comparison of end-to-end models for long-form speech recognition” (with Chung-Cheng Chiu et al)

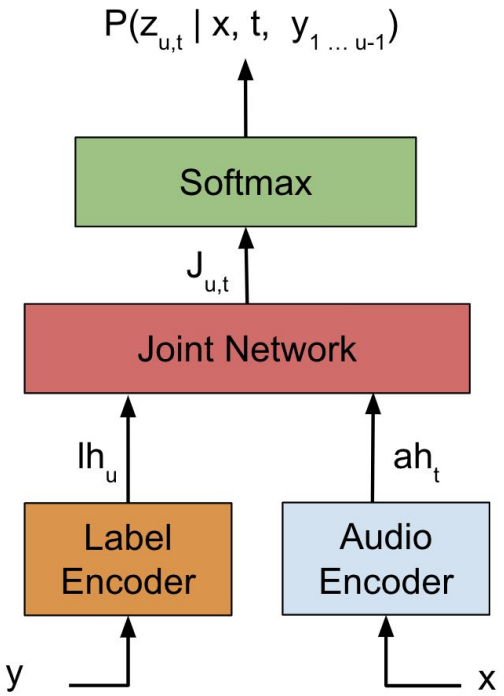
Non-streaming model has significantly lower long form errors than streaming models

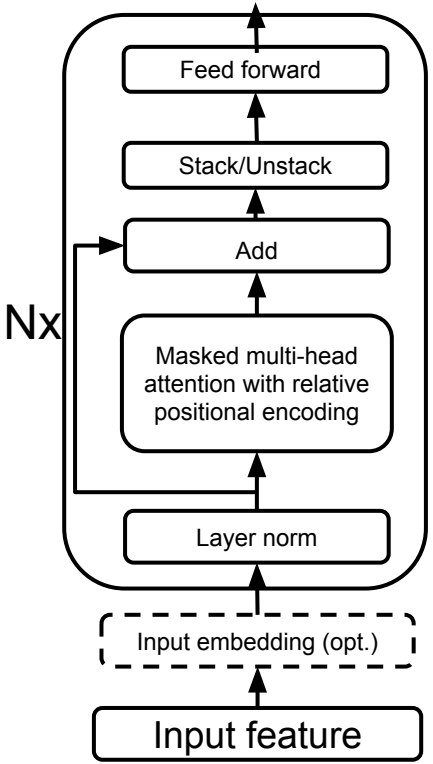
Table 2. WERs of ASR models trained on *Confisland*.

	Test set	Streaming model on *Confisland*	Non-streaming teacher model on *Confisland*
French	YT-long	34.5	18.6
	Common Voice	36.2	33.2
Spanish	YT-long	35.9	18.6
	Common Voice	22.0	11.2
Portuguese	YT-long	30.8	22.8
	Common Voice	30.9	25.8
Italian	YT-long	24.0	16.2
	Common Voice	30.0	27.3

More details from ICASSP 2021: Improving streaming automatic speech recognition with non-streaming model distillation on unsupervised data (with Thibault Doutre and Wei Han et al)

Transformer-Transducer

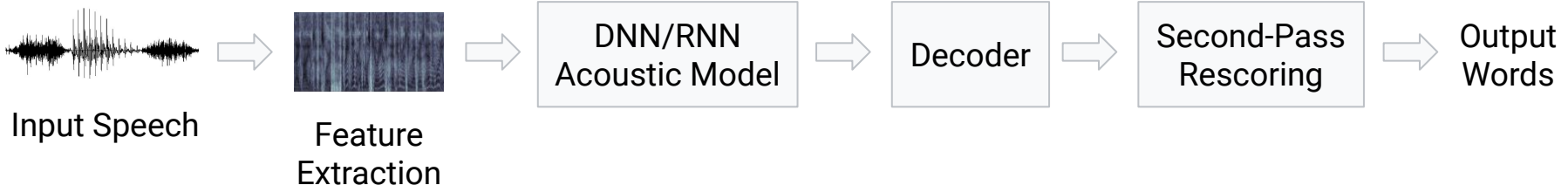




Zhang et al, Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss, ICASSP 2020

Conventional speech recognition

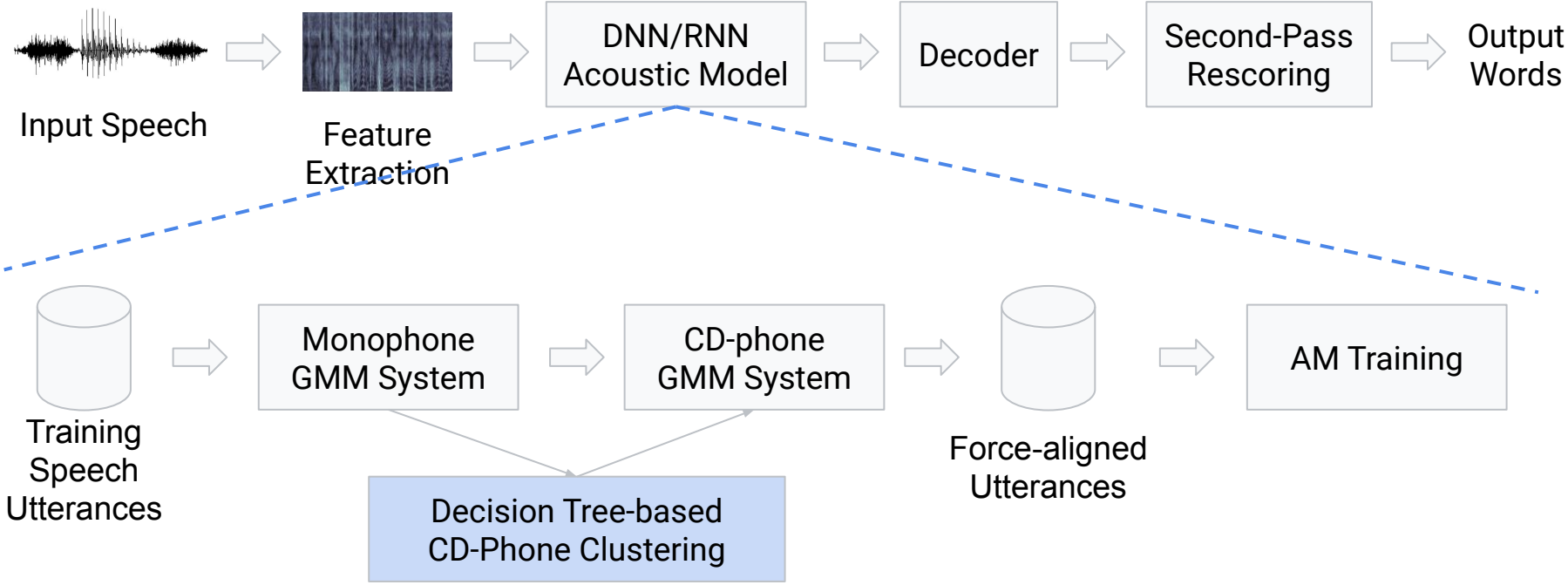
Pipeline



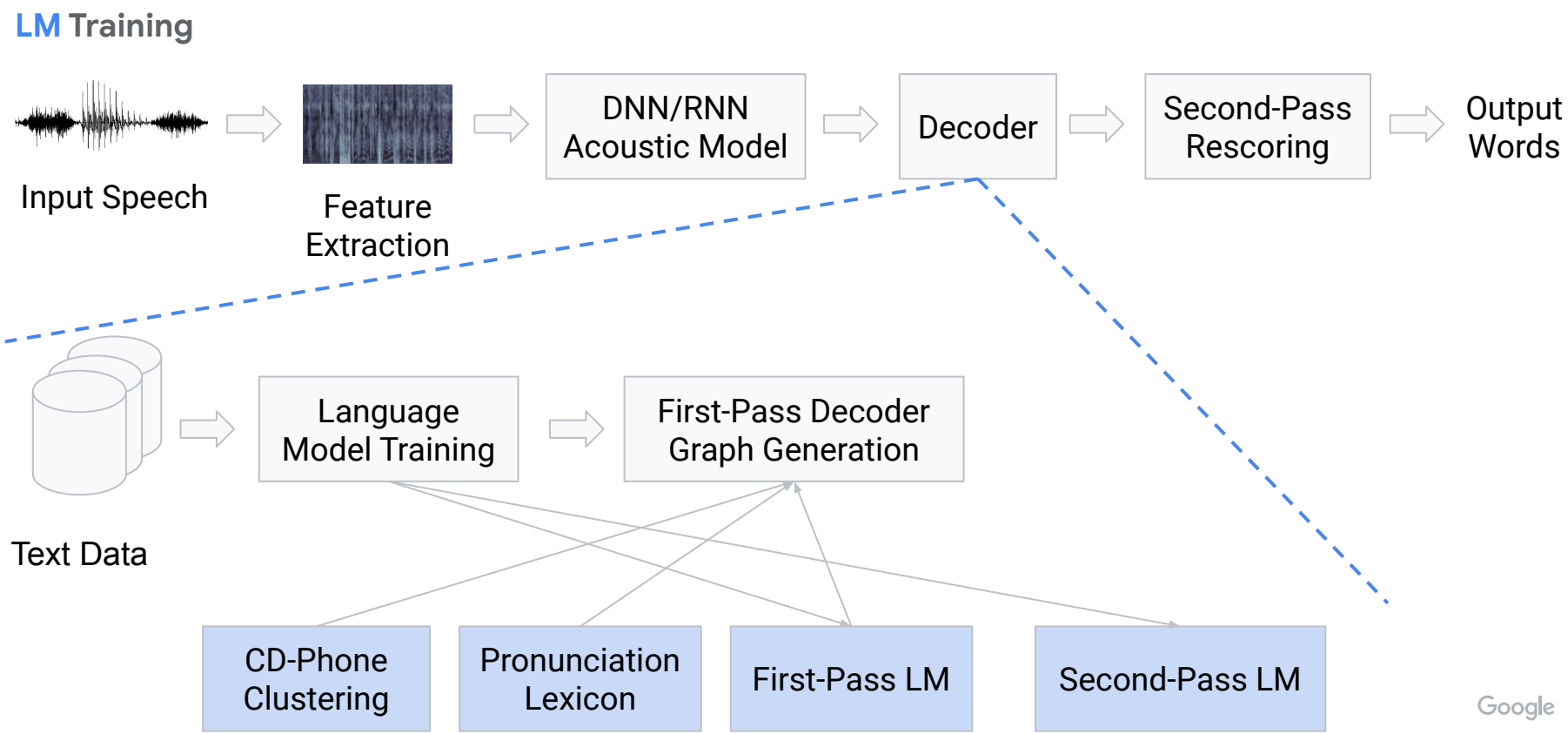
Many of the following slides are borrowed from Bo Li et al's ISCSLP'18 Tutorial

Conventional speech recognition

AM Training

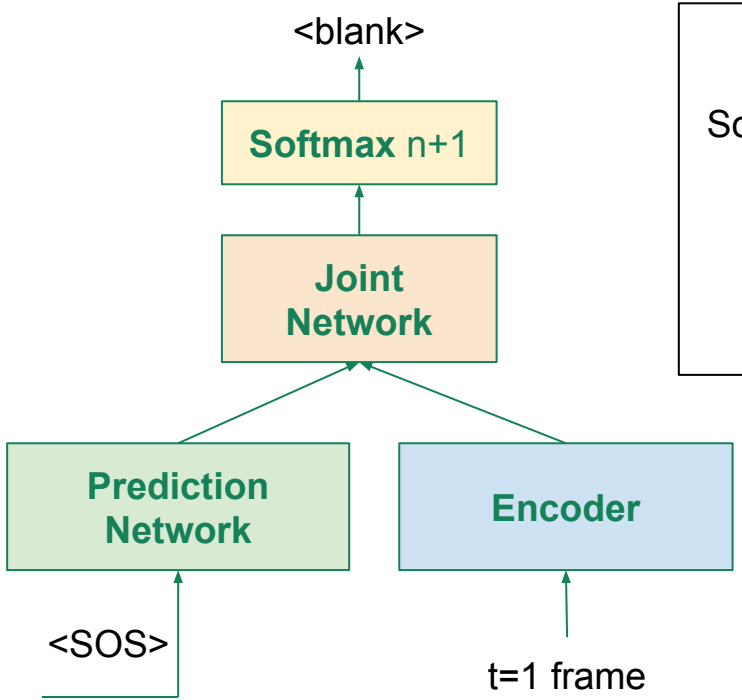


Conventional speech recognition



Recurrent Neural Network Transducer (RNN-T)

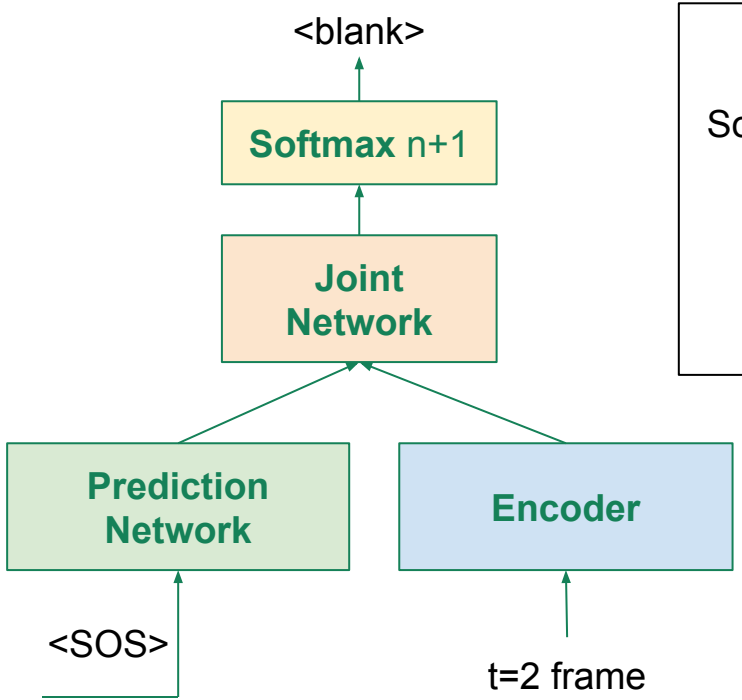
Output:



Softmax over n + 1 labels, includes a blank like CTC.
<blank> → advance in Encoder, retain prediction network state

Recurrent Neural Network Transducer (RNN-T)

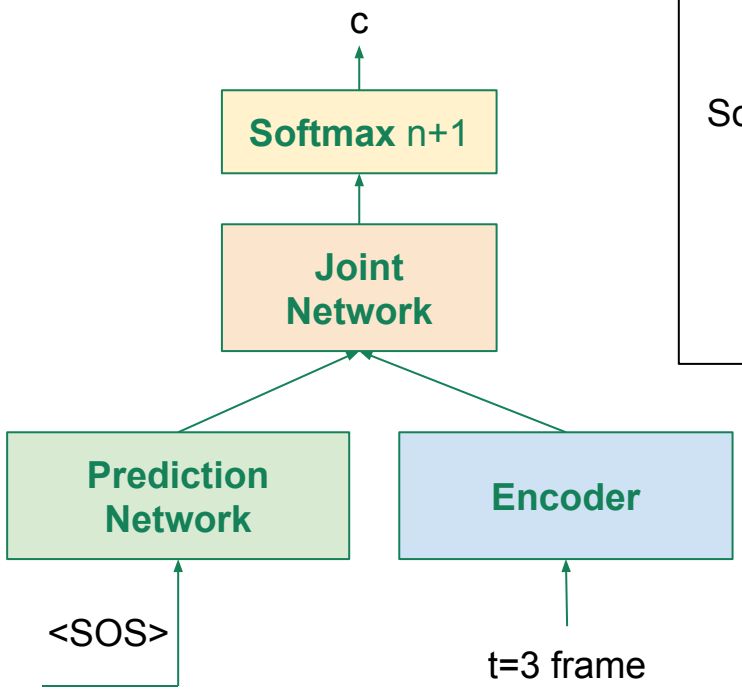
Output:



Softmax over n + 1 labels, includes a blank like CTC.
<blank> → advance in Encoder, retain prediction network state

Recurrent Neural Network Transducer (RNN-T)

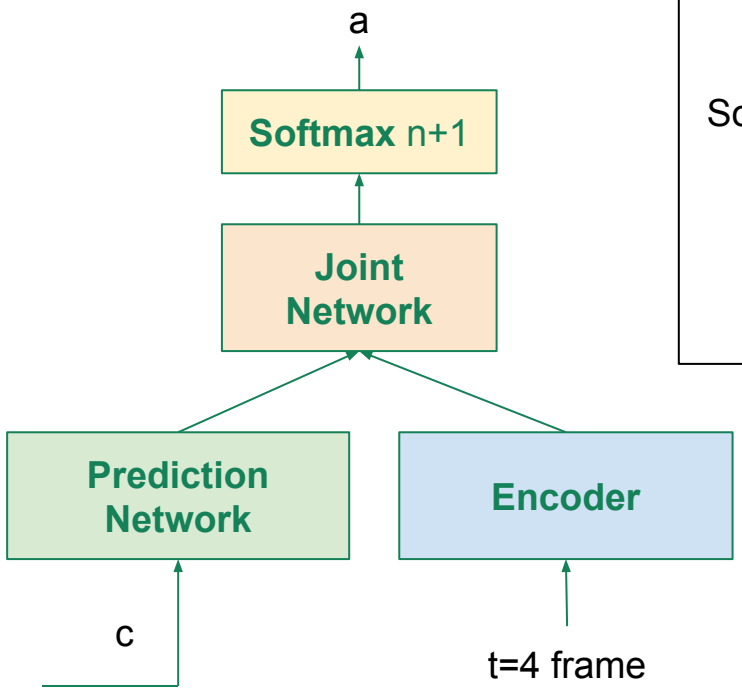
Output: c



Softmax over n + 1 labels, includes a blank like CTC.
<blank> → advance in Encoder, retain prediction network state

Recurrent Neural Network Transducer (RNN-T)

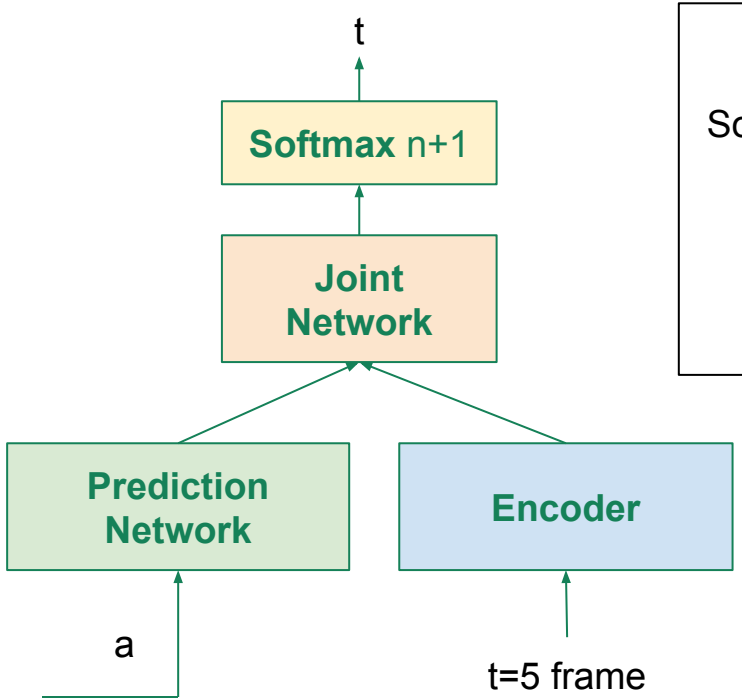
Output: ca



Softmax over n + 1 labels, includes a blank like CTC.
<blank> → advance in Encoder, retain prediction network state

Recurrent Neural Network Transducer (RNN-T)

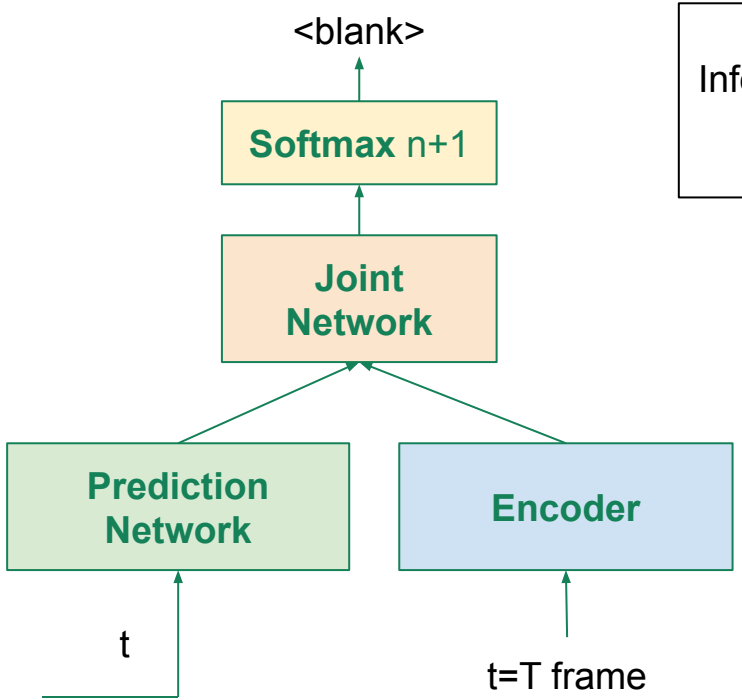
Output: cat



Softmax over n + 1 labels, includes a blank like CTC.
<blank> → advance in Encoder, retain prediction network state

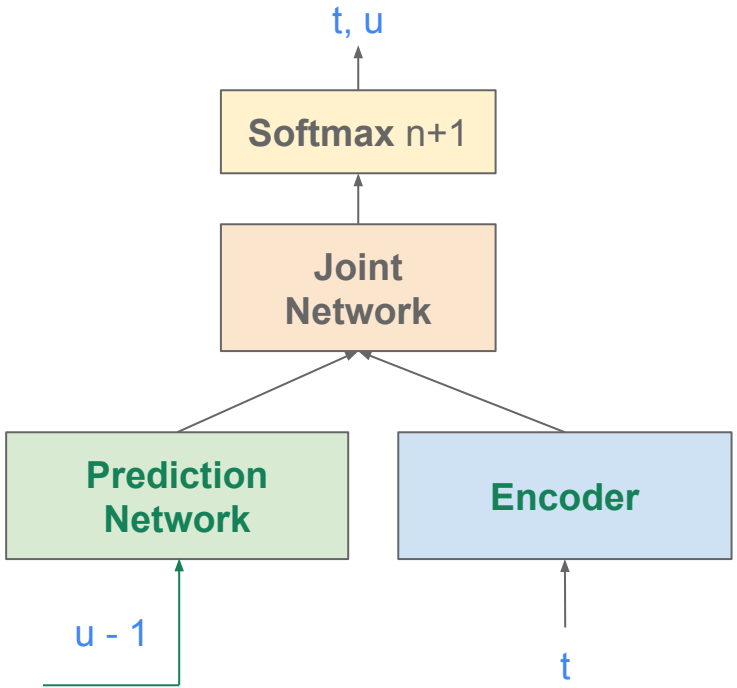
Recurrent Neural Network Transducer (RNN-T)

Output: cat

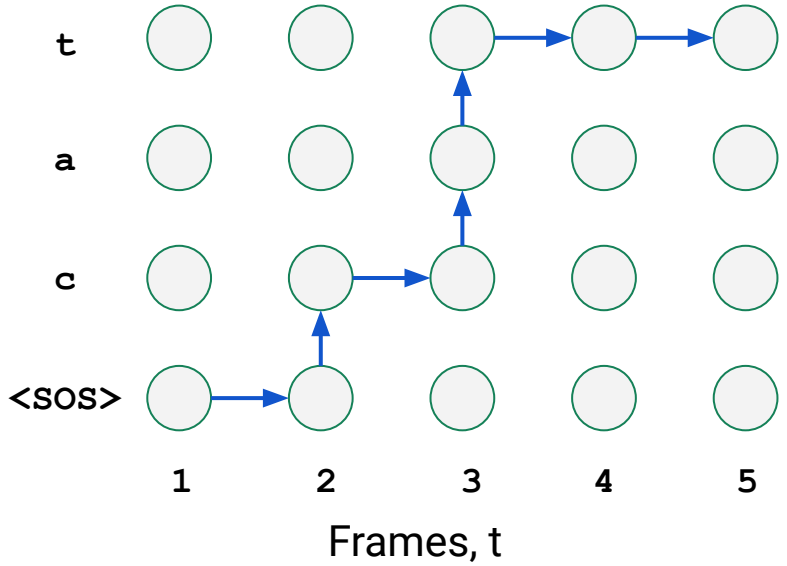


Inference terminates when all input frames have been consumed

Recurrent Neural Network Transducer (RNN-T)



During training feed the true label sequence to the LM.
Given a target sequence of length U and T acoustic frames we generate UxT softmax



ASR features for downstream tasks

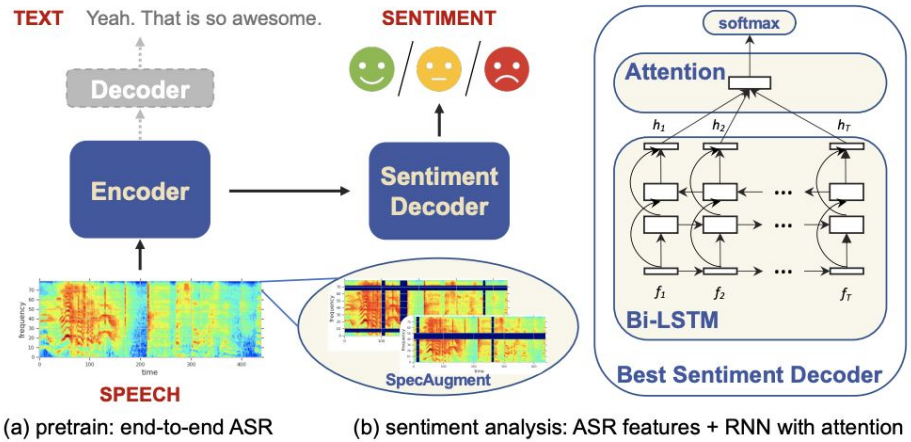


Fig. 1. We propose to use pre-trained features from e2e ASR model to solve sentiment analysis. The best performed sentiment decoder is RNN with self-attention. We apply SpecAugment to reduce overfitting in the training.

ICASSP 2019: "Speech Sentiment Analysis via Pre-trained Features from End-to-end ASR Models" (with Zhiyun Lu, Yu Zhang, Chung-Cheng Chiu, James Fan)

ASR features for downstream tasks

Table 2. Speech sentiment analysis performances of different methods.

	IEMOCAP dataset			SWBD-sentiment dataset		
Input features	Architecture	WA (%)	UA (%)	Architecture	WA (%)	UA (%)
acoustic	DRN + Transformer [1]	-	67.4	CNN	54.23	39.63
acoustic + text	DNN [7]	66.6	68.7	CNN and LSTM	65.65	54.59
e2e ASR	RNN w/ attention	71.7	72.6	RNN w/ attention	70.10	62.39
-	human	91.0	91.2	human	85.76	84.61