CS 224S / Linguist 285
Spoken Language Processing

Abhinav Garg | Stanford University | Spring 2024

Lecture 9: State-of-the-art ASR with deep learning

Outline

- Listen, Attend & Spell
- Convolutional Transformer (Conformer)
- Semi Supervision in ASR (wav2vec2)
- Weak Supervision and Whisper
- LLM + Speech (SpeechGPT, AudioPalm)
- Streaming ASR

Listen, Attend, & Spell

Listen, Attend, and Spell

- Discriminative, character-based encoder-decoder
- Unlike CTC:
  - Outputs also condition on previous outputs so far
  - No blank/epsilon. LAS just outputs characters
- Attention-based decoder. Precursor to modern encoder-decoder and transformer approaches

$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x}, y_{<i})$$

$$\mathbf{h} = \text{Listen}(\mathbf{x})$$
$$P(\mathbf{y}|\mathbf{x}) = \text{AttendAndSpell}(\mathbf{h}, \mathbf{y})$$

From: Chan, Jaitly, Le, & Vinyals. 2015

Listen, Attend, and Spell

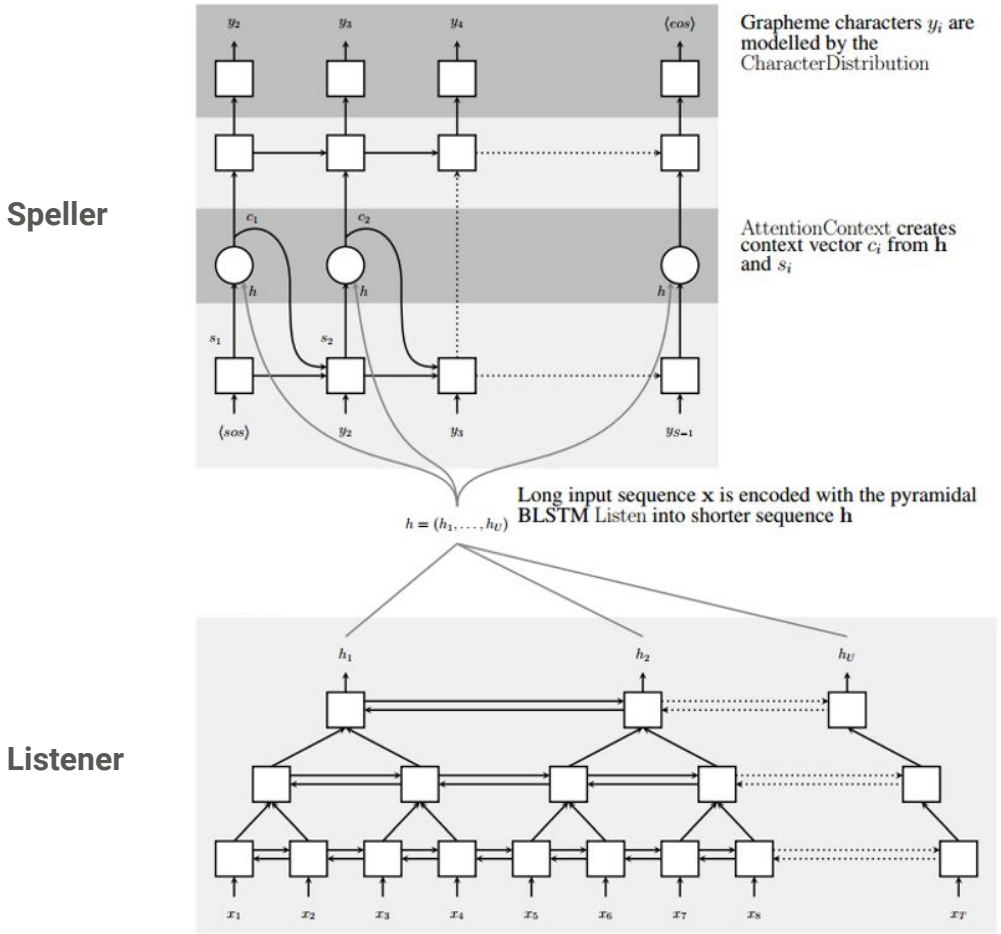


Figure: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence x into high level features h, the speller is an attention-based decoder generating the y characters from h.

(Chan, Jaitly, Le, & Vinyals. 2015)

Listen, Attend, and Spell

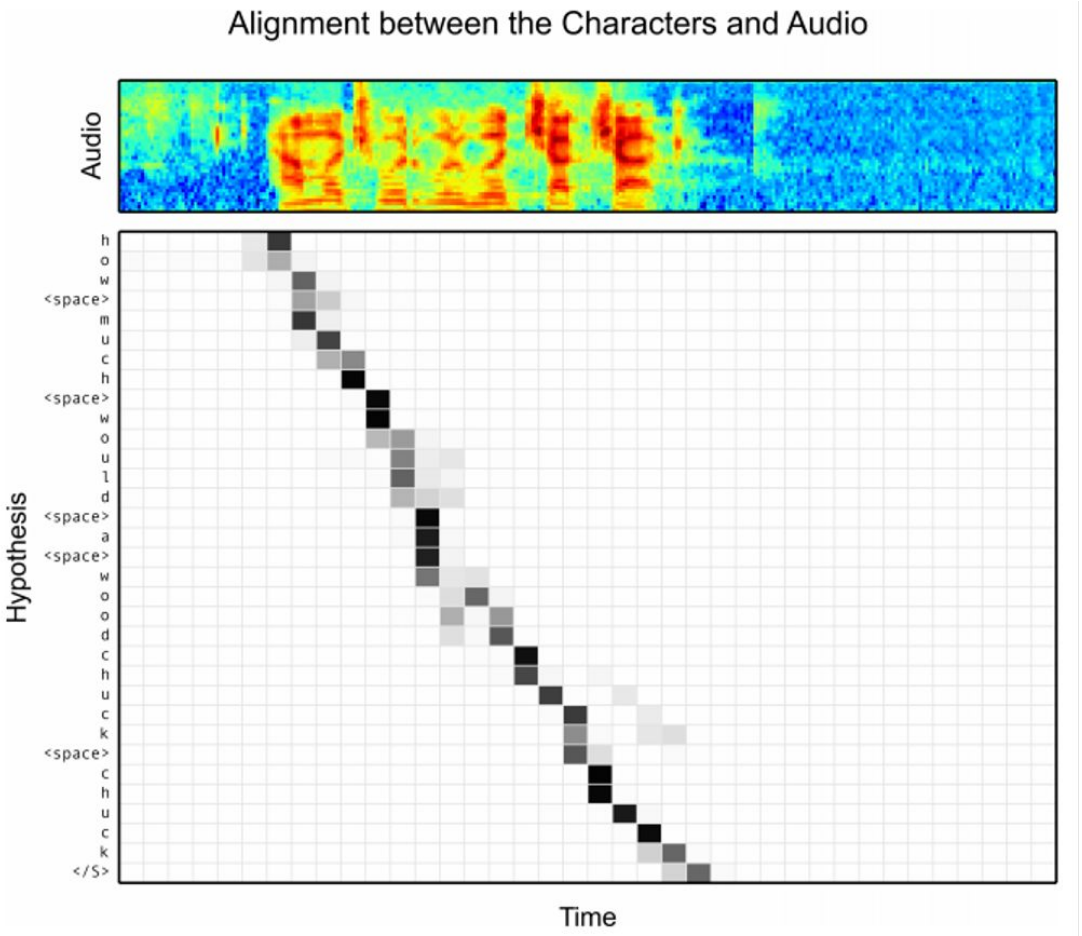


Figure: Chan, Jaitly, Le, & Vinyals. 2015

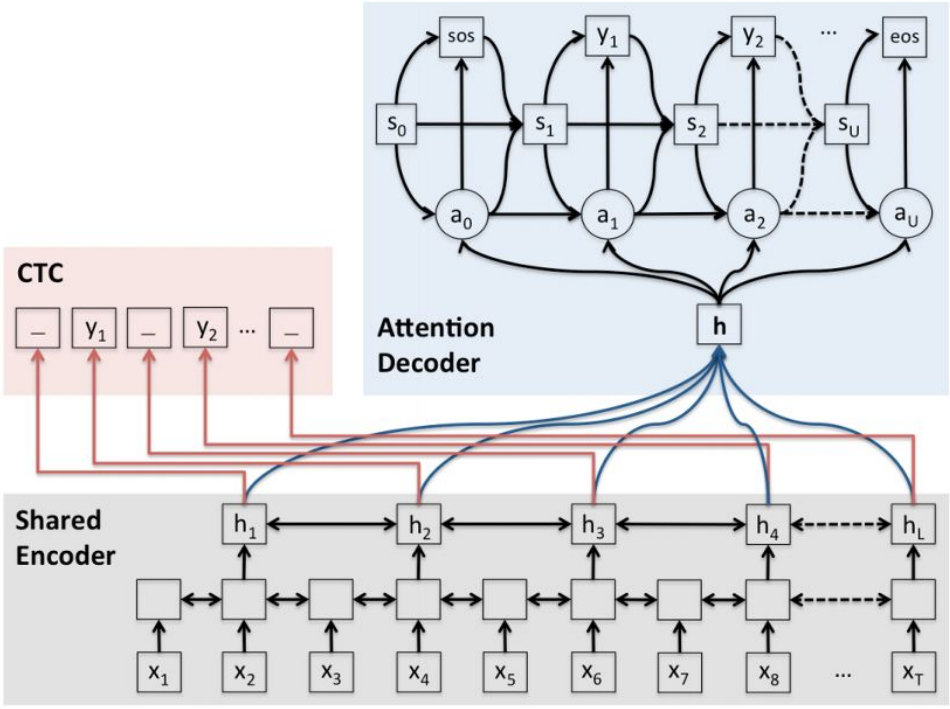
Listen, Attend, and Spell

Model	Clean WER	Noisy WER
CLDNN-HMM [20]	8.0	8.9
LAS	16.2	19.0
LAS + LM Rescoring	12.6	14.7
LAS + Sampling	14.1	16.5
LAS + Sampling + LM Rescoring	10.3	12.0

Table: WER comparison on the clean and noisy Google voice search task. The CLDNN-HMM system is the state-of-the-art system, the Listen, Attend and Spell (LAS) models are decoded with a beam size of 32. Language Model (LM) rescoring was applied to our beams, and a sampling trick was applied to bridge the gap between training and inference.

(Chan, Jaitly, Le, & Vinyals. 2015)

CTC + LAS Multi-Task Approach



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

Figure: Our proposed Joint CTC-attention based end-to-end framework: the shared encoder is trained by both CTC and attention model objectives simultaneously. The shared encoder transforms our input sequence x into high level features h, the location-based attention decoder generates the character sequence y

(Kim, Hori, & Watanabe. 2017)

CTC + LAS Multi-Task Approach

Model(train)	CER(valid)	CER(eval)
WSJ-train_si284 (80hrs)	dev93	eval92
CTC	11.48	8.97
Attention(content-based)	13.68	11.08
Attention(location-based)	11.98	8.17
MTL($\lambda = 0.2$)	**11.27**	**7.36**
MTL($\lambda = 0.5$)	12.00	8.31
MTL($\lambda = 0.8$)	11.71	8.45
WSJ-train_si84 (15hrs)	dev93	eval92
CTC	27.41	20.34
Attention(content-based)	28.02	20.06
Attention(location-based)	24.98	17.01
MTL($\lambda = 0.2$)	**23.03**	**14.53**
MTL($\lambda = 0.5$)	26.28	16.24
MTL($\lambda = 0.8$)	32.21	21.30
CHiME-4-tr05_multi (18hrs)	dt05_real	et05_real
CTC	37.56	48.79
Attention(content-based)	43.45	54.25
Attention(location-based)	35.01	47.58
MTL($\lambda = 0.2$)	**32.08**	**44.99**
MTL($\lambda = 0.5$)	34.56	46.49
MTL($\lambda = 0.8$)	35.41	48.34

Table: Character Error Rate (CER) on clean corpora WSJ1 (80 hours) and WSJO (15 hours), and a noisy corpus CHiME-4 (18 hours). None of our experiments used any language model or lexicon information. (Word Error Rate (WER) of our model MTL(1 = 0.2) was 18.2% and WER of [7] was 18.6% on WSJ1.Note that this is not an exact comparison because the hyper parameters were not completely same as [7].)

(Kim, Hori, & Watanabe. 2017)

$$e_{u,l} = \begin{cases} \text{content-based:} \\ \quad w^T \tanh(W s_{u-1} + V h_l + b) \\ \text{location-based:} \\ \quad f_u = F * a_{u-1} \\ \quad w^T \tanh(W s_{u-1} + V h_l + U f_{u,l} + b) \end{cases}$$

$$a_{u,l} = \frac{\exp(\gamma e_{u,l})}{\sum_l \exp(\gamma e_{u,l})}$$

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

CTC + LAS Multi-Task Approach

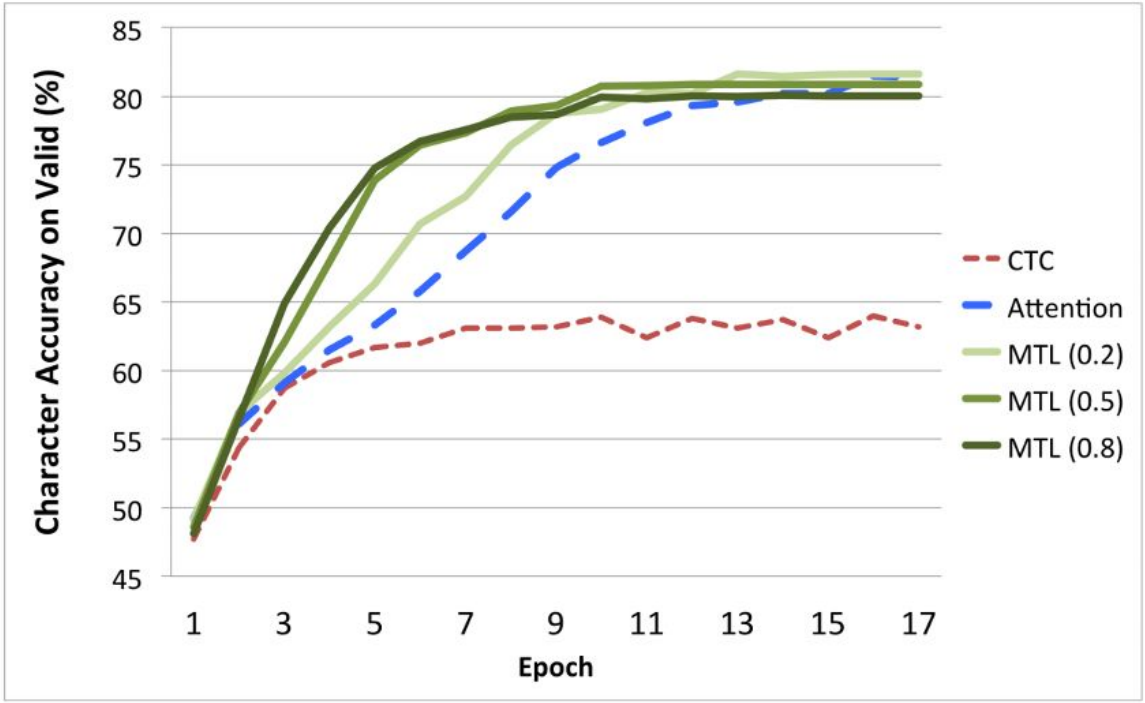


Figure: Comparison of learning curves: CTC, location-based attention model, and MTL with (1 = 0.2, 0.5, 0.8). The character accuracy on the validation set of CHiME-4 is calculated by edit distance between hypothesis and reference. Note that the reference history were used in the attention and our MTL models.

(Kim, Hori, & Watanabe. 2017)

CTC + LAS Multi-Task Approach

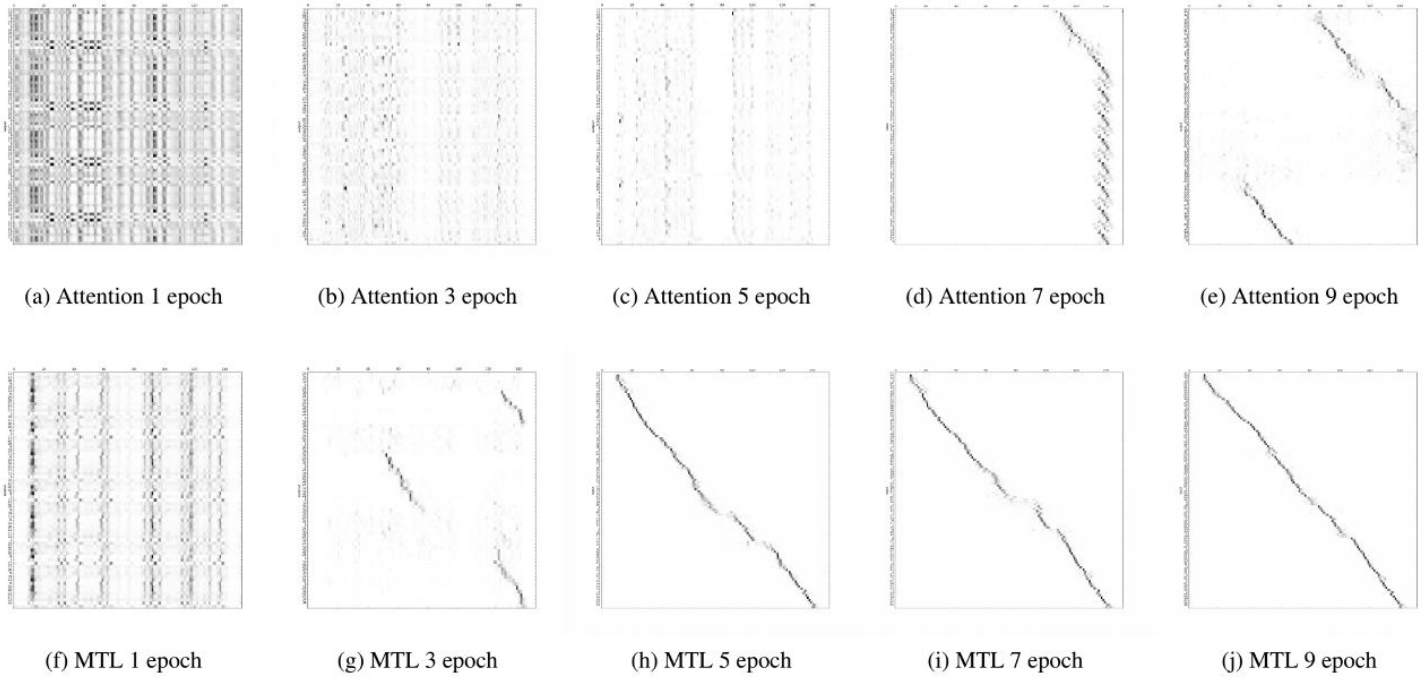


Figure: Comparison of speed in learning alignments between characters (y-axis) and acoustic frames (x-axis) between the location-based attention model (1st row) and our model MTL (2nd row) over training epoch (1,3,5,7, and 9). All alignments are for one manually chosen utterance F05_442C020U_CAF_REAL - "THE ONE HUNDRED SHARE INDEX CLOSED SIX POINT EIGHT POINTS LOWER AT ONE THOUSAND SEVEN HUNDRED FIFTY NINE POINT NINE") in the noisy CHiME-4 evaluation set. (Kim, Hori, & Watanabe. 2017)

CTC + LAS Multi-Task Approach

Model	Hour	Task1	Task2	Task3
Attention	581	11.4	7.9	9.0
MTL	581	10.5	7.6	8.3
MTL + joint decoding (rescoring)	581	10.1	7.1	7.8
MTL + joint decoding (one pass)	581	10.0	7.1	7.6
MTL-large + joint decoding (rescoring)	581	**8.4**	6.2	**6.9**
MTL-large + joint decoding (one pass)	581	**8.4**	**6.1**	**6.9**
GMM-discr. (Moriya et al., 2015)	236 for AM, 581 for LM	11.2	9.2	12.1
DNN/HMM (Moriya et al., 2015)	236 for AM, 581 for LM	9.0	7.2	9.6
CTC-syllable (Kanda et al., 2016)	581	9.4	7.3	7.5

Table: Character error rate (CER) for conventional attention and hybrid CTC/attention end-to-end ASR. Corpus of Spontaneous Japanese speech recognition (CSJ) task.

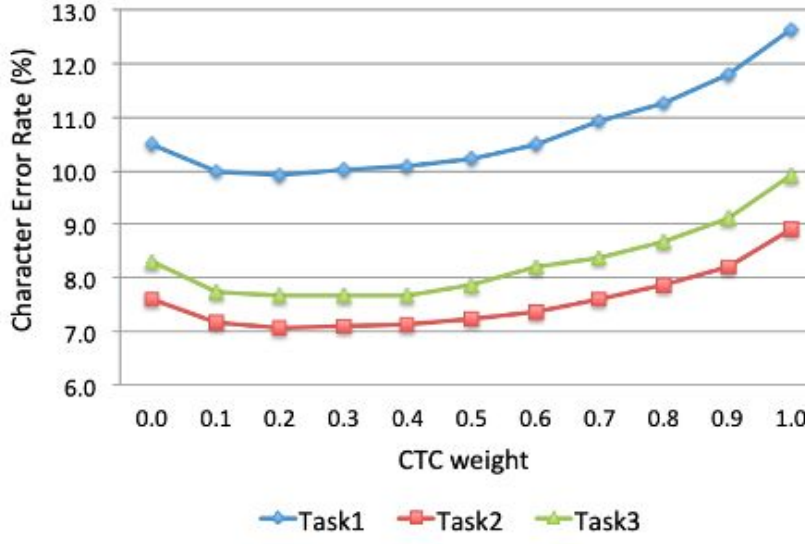


Figure: The effect of weight parameter \ in Eq. (14) on the CSJ evaluation tasks (The CERs were obtained by one-pass decoding).

(Hori, Watanabe, Zhang, & Chan. 2017)

Convolutional Transformer

Conformer: Convolution-augmented Transformer for Speech Recognition

- Sequence-to-sequence transformer with multi-headed self attention.
- Combines attention (global context) with convolution (local invariance)
- RNN-T loss architecture

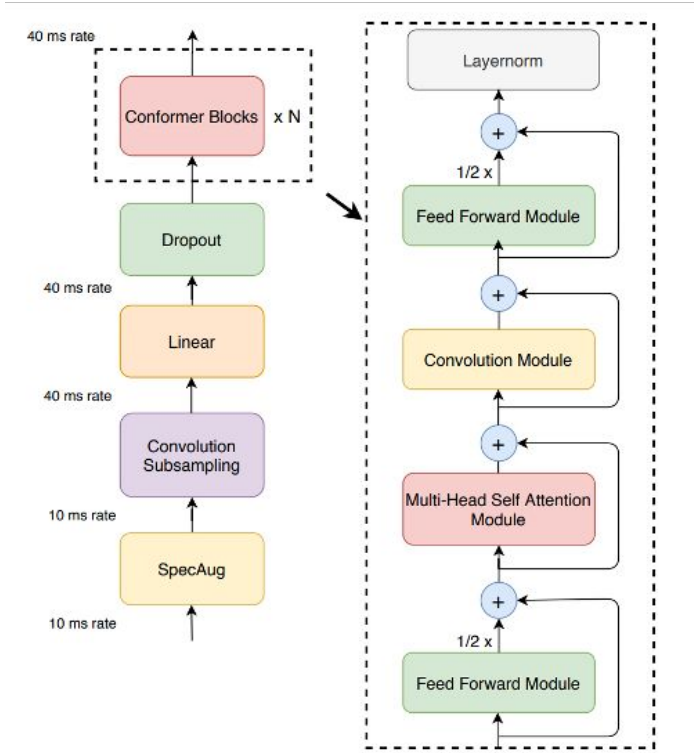


Figure: Conformer encoder model architecture. Conformer comprises of two macaron-like feed-forward layers with half-step residual connections sandwiching the multi-headed self-attention and convolution modules. This is followed by a post layernorm.

(Gulati *et al.* 2020)

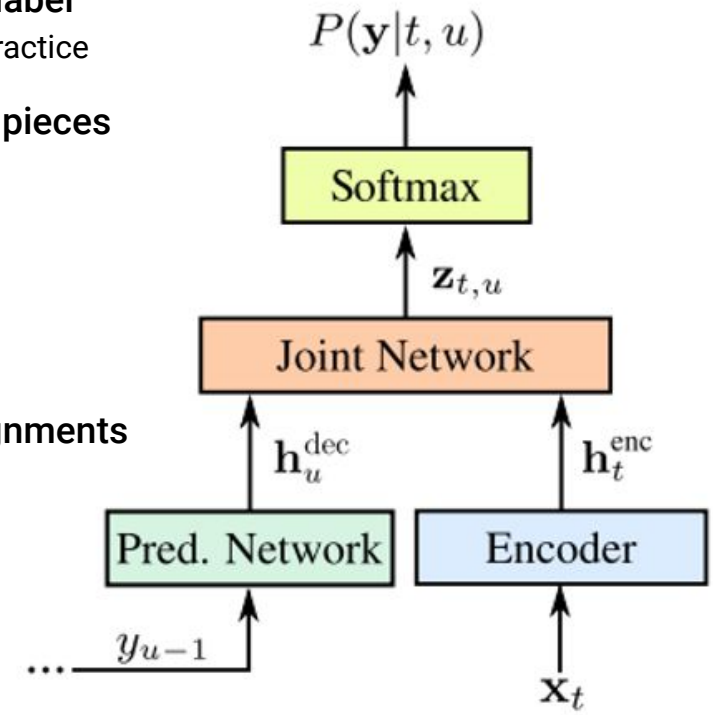
Conformer: RNN-Transducer Loss

- Directly optimizes target word sequence as correct label
  - Graphemes (letters) or word parts (10k-50k) used in practice
- Learned combination of acoustic + language model pieces
- Conditions on sequence output so far (y_{t-1})
- Single alignment:

$$P(\mathbf{z}|\mathbf{x}) = \prod_i P(z_i|\mathbf{x}, t_i, \text{Labels}(z_{1:(i-1)}))$$

- Maximize P(y|x) by summing over all consistent alignments

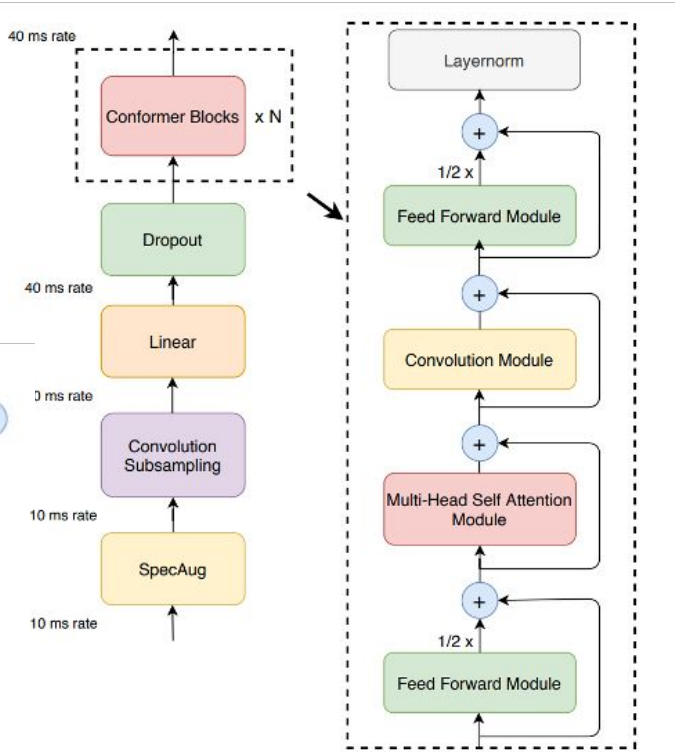
$$P(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{z} \in \mathcal{Z}(\mathbf{y}, T)} P(\mathbf{z}|\mathbf{x})$$

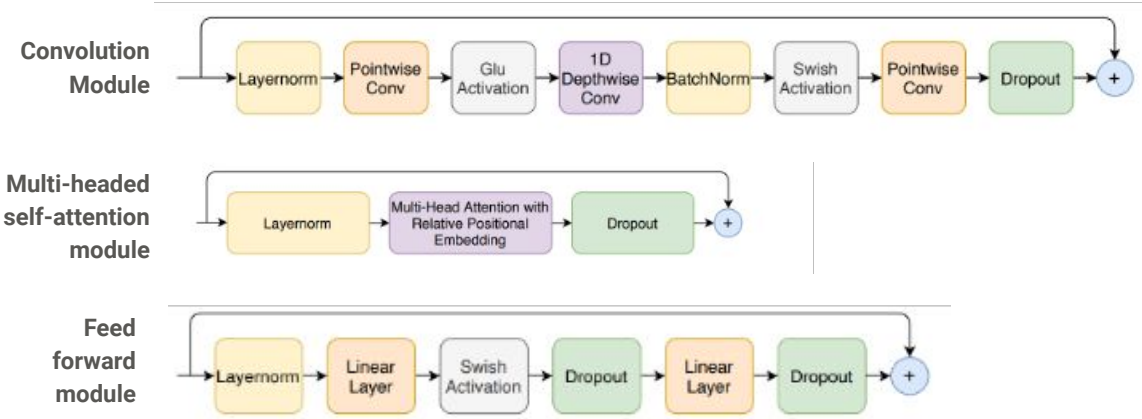


RNN-T loss: (Graves, 2012)
Figure: Rao, Sak, & Prabhavalkar. 2017)

Conformer: Convolution-augmented Transformer for Speech Recognition

- Sequence-to-sequence transformer with multi-headed self attention. Directly optimizes target word sequence
- Combines attention (global context) with convolution (local invariance)

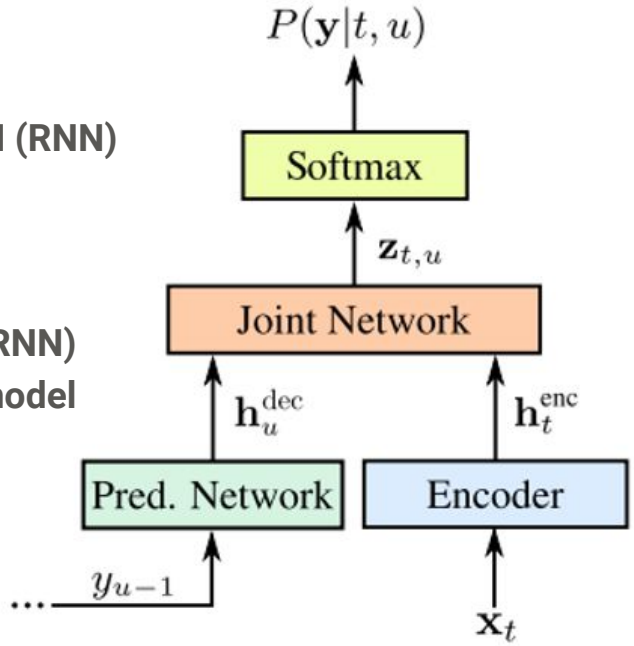


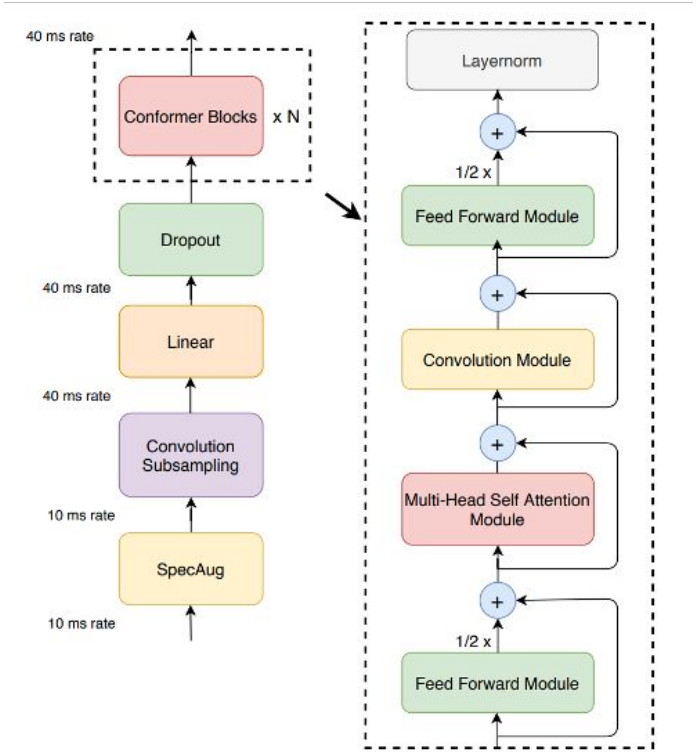


Conformer: Putting it All Together

1 hidden layer LSTM (RNN)

3 hidden layer LSTM (RNN)
Pre-trained as language model





Conformer: Convolution-augmented Transformer for Speech Recognition

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: Comparison of Conformer with recent published models. Our model shows improvements consistently over various model parameter size constraints. At 10.3M parameters, our model is 0.7% better on test other when compared to contemporary work, ContextNet(S) [10]. At 30.7M model parameters our model already significantly outperforms the previous published state of the art results of Transformer Transducer [7] with 139M parameters.

(Gulati *et al.* 2020)

Semi Supervision in ASR

Wav2Vec2 (Pretraining)

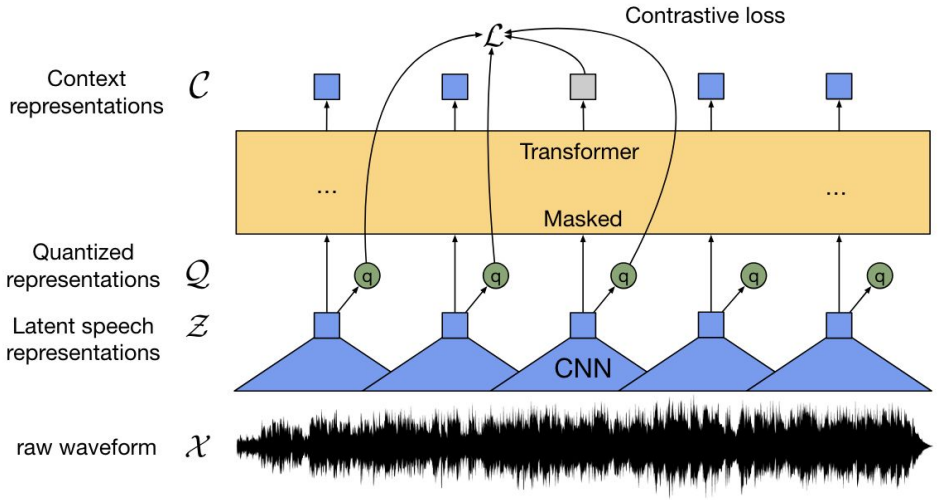


Table: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units.

(Baevski et al. 2020)

Wav2Vec2 (Fine Tuning)

- Pre trained model are general purpose and can be used for any downstream task such as emotion detection, speaker identification, etc.
- Pre-trained models are fine-tuned for speech recognition by adding a randomly initialized linear projection on top of the context network into classes representing the vocabulary of the task.
- The fine-tuning process involves optimizing models by minimizing a CTC loss
- 4-gram and Transformer LMs are used for beam search decoding

Wav2Vec2

Model	Unlabeled data	LM	dev		test	
			clean	other	clean	other
10 min labeled						
Discrete BERT [4]	LS-960	4-gram	15.7	24.1	16.3	25.2
BASE	LS-960	4-gram	8.9	15.7	9.1	15.6
		Transf.	6.6	13.2	6.9	12.9
LARGE	LS-960	Transf.	6.6	10.6	6.8	10.8
	LV-60k	Transf.	4.6	7.9	4.8	8.2
1h labeled						
Discrete BERT [4]	LS-960	4-gram	8.5	16.4	9.0	17.6
BASE	LS-960	4-gram	5.0	10.8	5.5	11.3
		Transf.	3.8	9.0	4.0	9.3
LARGE	LS-960	Transf.	3.8	7.1	3.9	7.6
	LV-60k	Transf.	2.9	5.4	2.9	5.8
10h labeled						
Discrete BERT [4]	LS-960	4-gram	5.3	13.2	5.9	14.1
Iter. pseudo-labeling [58]	LS-960	4-gram+Transf.	23.51	25.48	24.37	26.02
	LV-60k	4-gram+Transf.	17.00	19.34	18.03	19.92
BASE	LS-960	4-gram	3.8	9.1	4.3	9.5
		Transf.	2.9	7.4	3.2	7.8
LARGE	LS-960	Transf.	2.9	5.7	3.2	6.1
	LV-60k	Transf.	2.4	4.8	2.6	4.9
100h labeled						
Hybrid DNN/HMM [34]	-	4-gram	5.0	19.5	5.8	18.6
TTS data augm. [30]	-	LSTM			4.3	13.5
Discrete BERT [4]	LS-960	4-gram	4.0	10.9	4.5	12.1
Iter. pseudo-labeling [58]	LS-860	4-gram+Transf.	4.98	7.97	5.59	8.95
	LV-60k	4-gram+Transf.	3.19	6.14	3.72	7.11
Noisy student [42]	LS-860	LSTM	3.9	8.8	4.2	8.6
BASE	LS-960	4-gram	2.7	7.9	3.4	8.0
		Transf.	2.2	6.3	2.6	6.3
LARGE	LS-960	Transf.	2.1	4.8	2.3	5.0
	LV-60k	Transf.	1.9	4.0	2.0	4.0

Table: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units.

(Baevski et al. 2020)

Weak Supervision and Whisper

Weak supervision

- Labelled data are generally scarce (Librispeech is 1000 hrs)
- Models like wav2vec2 can use unlabelled data (much more abundant ~1,000,000, Zhang et al.) to train encoder, but , the lack of an equivalently high-quality pre-trained decoder, limits their usefulness and robustness
- Whisper uses weak supervision to create : 680,000 hours of multilingual and multitask labeled audio data
- 680,000 hours of audio : 117,000 hours cover 96 other languages, 125,000 hours of X→en translation data; 438 hours of English language
- Used multiple heuristics to get high quality data and remove machine generated transcripts
- Whisper Demo

Whisper (Model Architecture)

Multitask training data (680k hours)

English transcription

"Ask not what your country can do for ···"

Ask not what your country can do for ···

Any-to-English speech translation

"El rápido zorro marrón salta sobre ···"

The quick brown fox jumps over ···

Non-English transcription

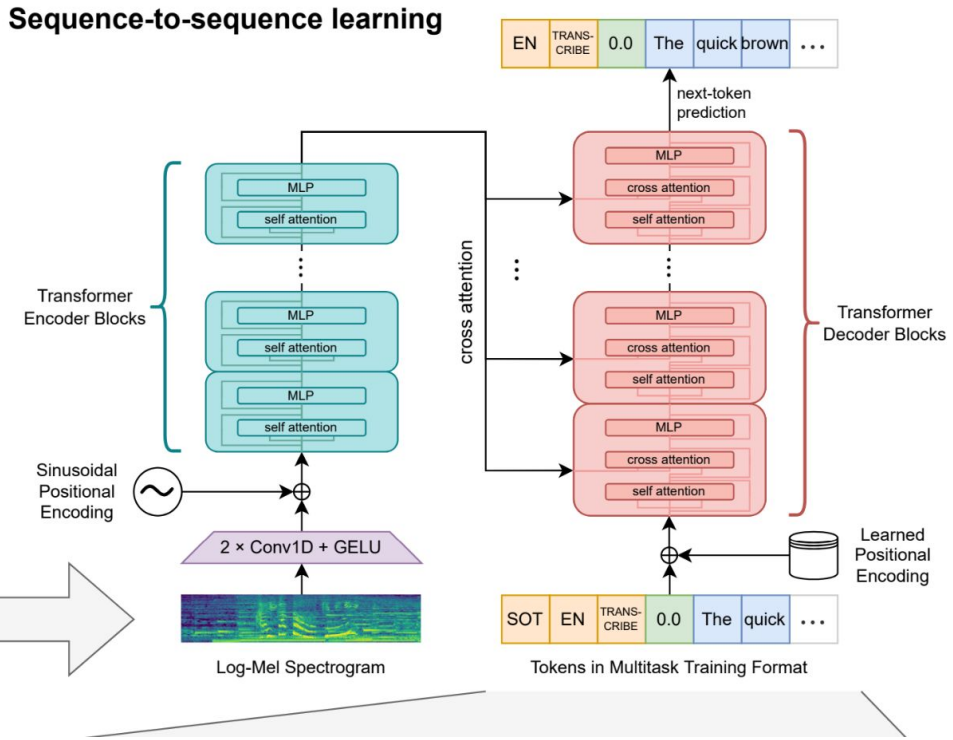
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···"

언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···

No speech

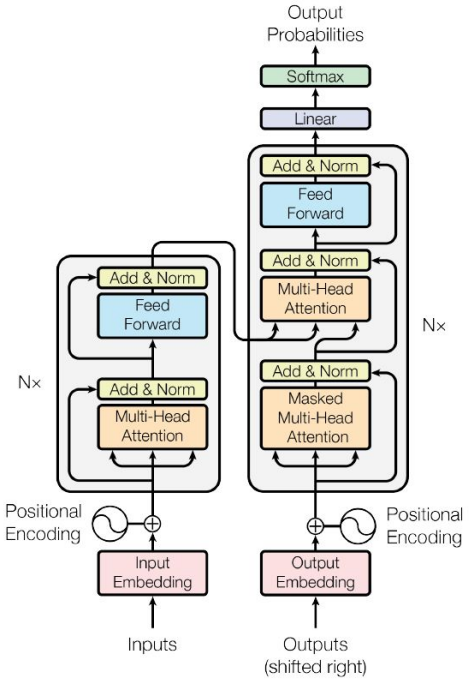
(background music playing)

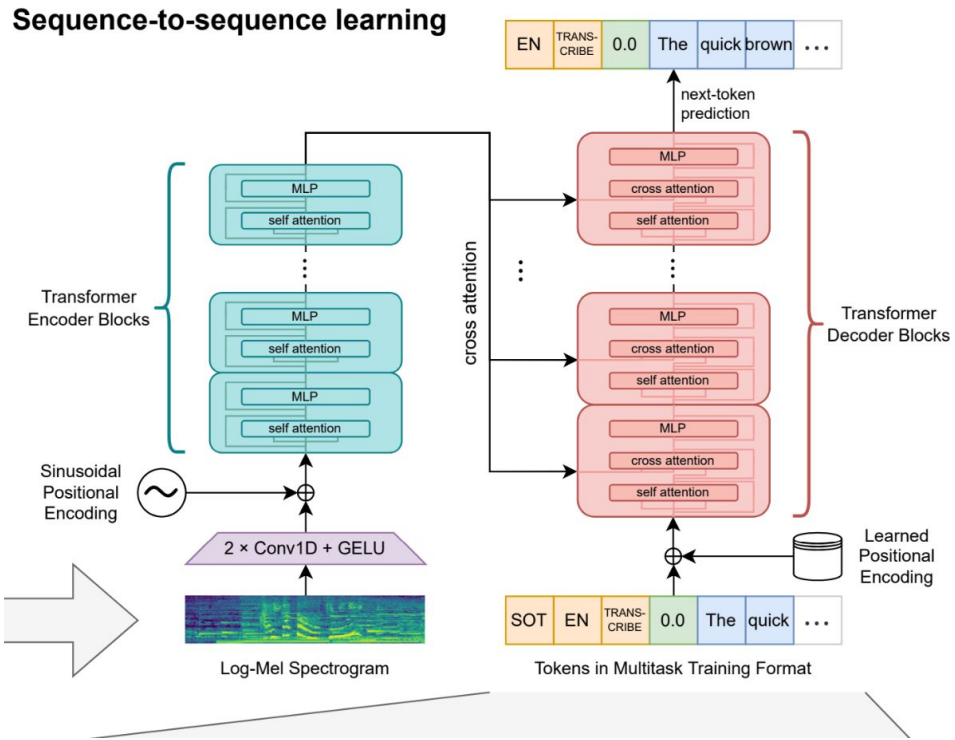
Ø



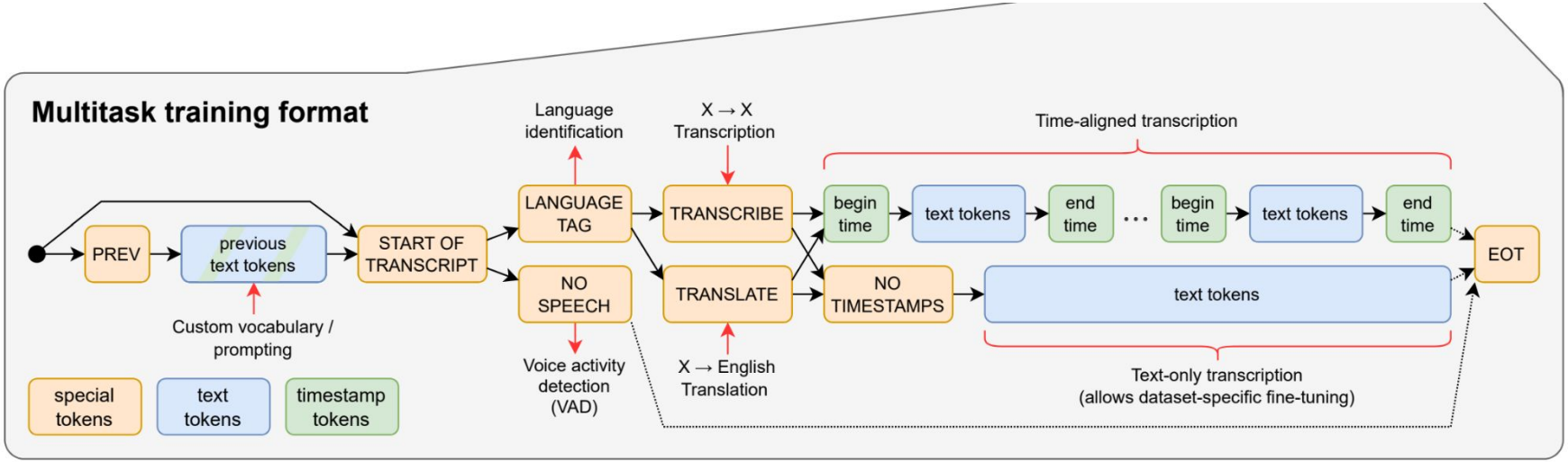
(Radford et al. 2020)

Whisper is a transformer encoder-decoder approach





Whisper (Multitask Training)



Whisper (Demo)

- Working in noisy env
- Working in other language
- Multi-lingual
- Give examples

Whisper (Results)

Dataset	wav2vec 2.0 Large (no LM)	Whisper Large V2	RER (%)
LibriSpeech Clean	**2.7**	**2.7**	0.0
Artie	24.5	**6.2**	74.7
Common Voice	29.9	**9.0**	69.9
Fleurs En	14.6	**4.4**	69.9
Tedlium	10.5	**4.0**	61.9
CHiME6	65.8	**25.5**	61.2
VoxPopuli En	17.9	**7.3**	59.2
CORAAL	35.6	**16.2**	54.5
AMI IHM	37.0	**16.9**	54.3
Switchboard	28.3	**13.8**	51.2
CallHome	34.8	**17.6**	49.4
WSJ	7.7	**3.9**	49.4
AMI SDM1	67.6	**36.4**	46.2
LibriSpeech Other	6.2	**5.2**	16.1
Average	29.3	**12.8**	55.2

Table: Although both models perform within 0.1% of each other on LibriSpeech, a zero-shot Whisper model performs much better on other datasets than expected for its LibriSpeech performance and makes 55.2% less errors on average.

(Radford et al. 2020)

Whisper (Results)

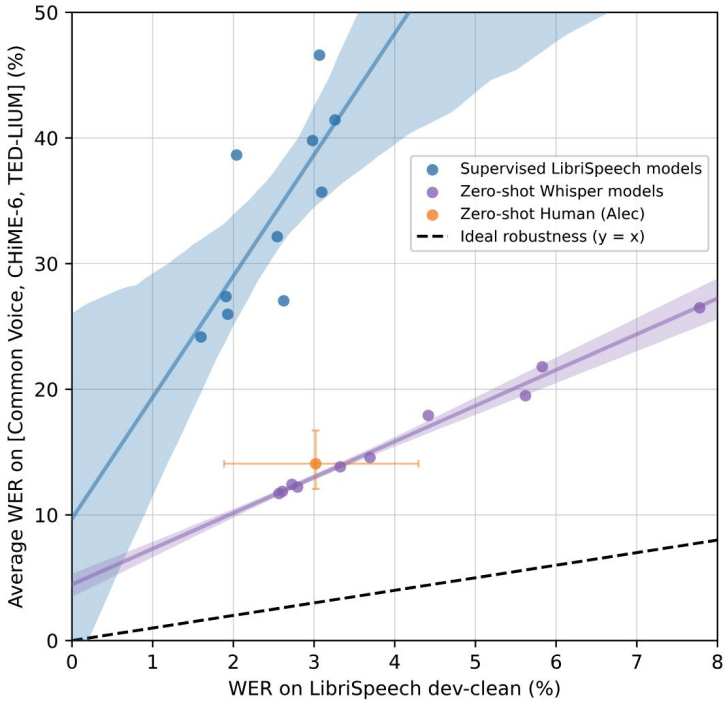


Figure: Zero-shot Whisper models close the gap to human robustness. Despite matching or outperforming a human on LibriSpeech dev-clean, supervised LibriSpeech models make roughly twice as many errors as a human on other datasets demonstrating their brittleness and lack of robustness. The estimated robustness frontier of zero-shot Whisper models, however, includes the 95% confidence interval for this particular human.

(Radford et al. 2020)

Whisper (Results)

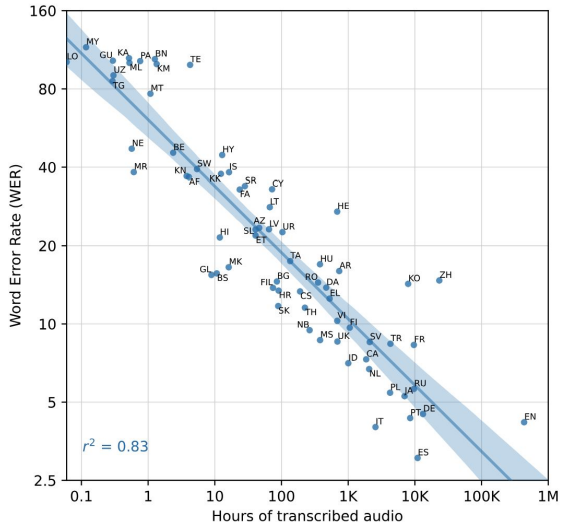


Figure 3. **Correlation of pre-training supervision amount with downstream speech recognition performance.** The amount of pre-training speech recognition data for a given language is very predictive of zero-shot performance on that language in Fleurs.

Observation: Many of the largest outliers are languages that have unique scripts and are more distantly related to the Indo-European languages.

Outliers: Hebrew (HE), Telugu (TE), Chinese (ZH), and Korean (KO)

(Radford et al. 2020)

Whisper (Results)

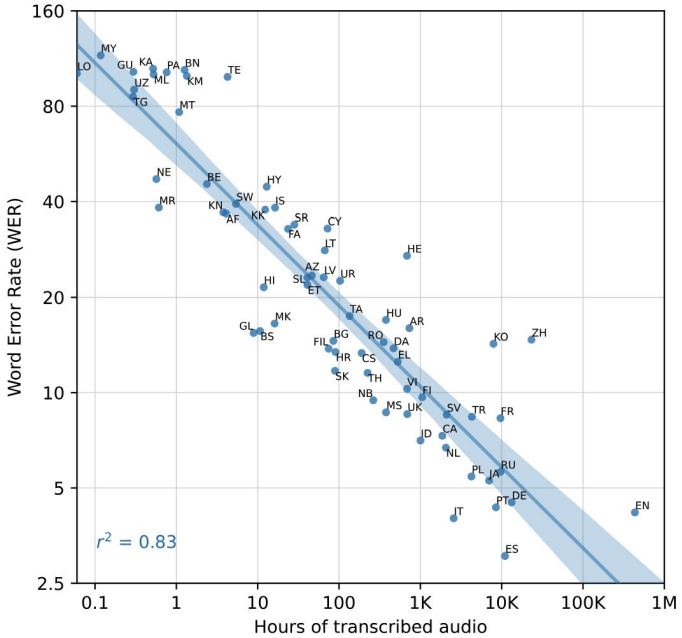


Figure 3. **Correlation of pre-training supervision amount with downstream speech recognition performance.** The amount of pre-training speech recognition data for a given language is very predictive of zero-shot performance on that language in Fleurs.

Observation: Many of the largest outliers are languages that have unique scripts and are more distantly related to the Indo-European languages.

Outliers: Hebrew (HE), Telugu (TE), Chinese (ZH), and Korean (KO)

(Radford et al. 2020)

Whisper (Results)

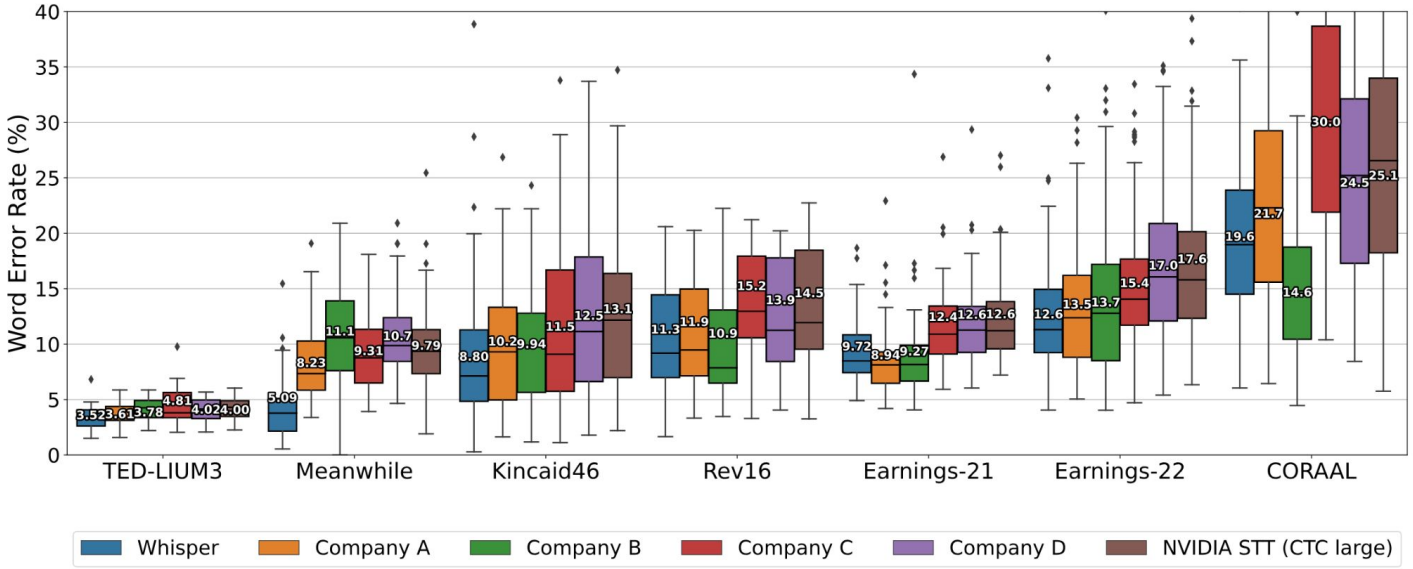
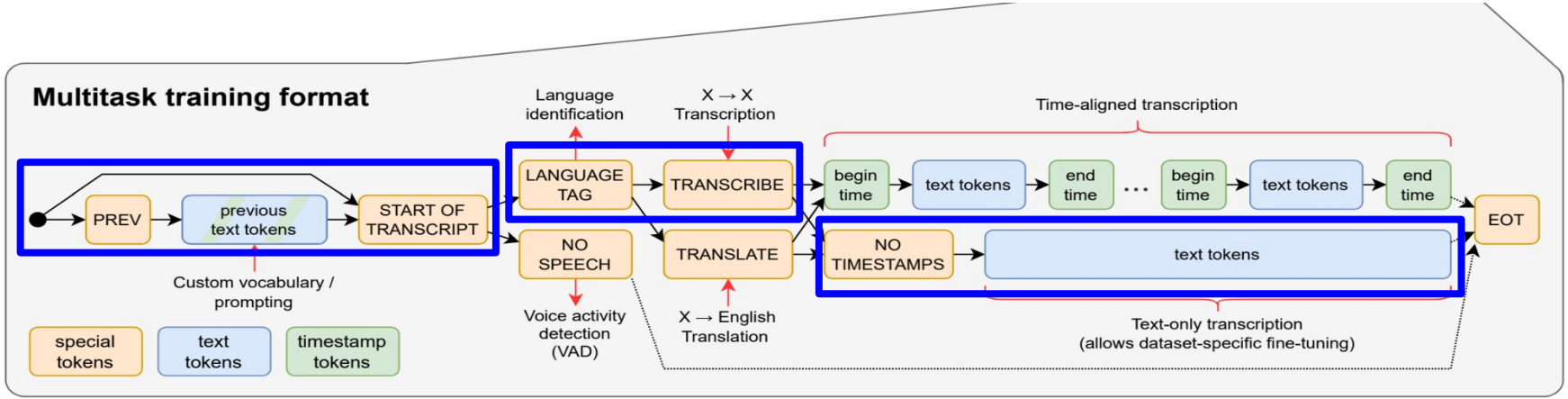


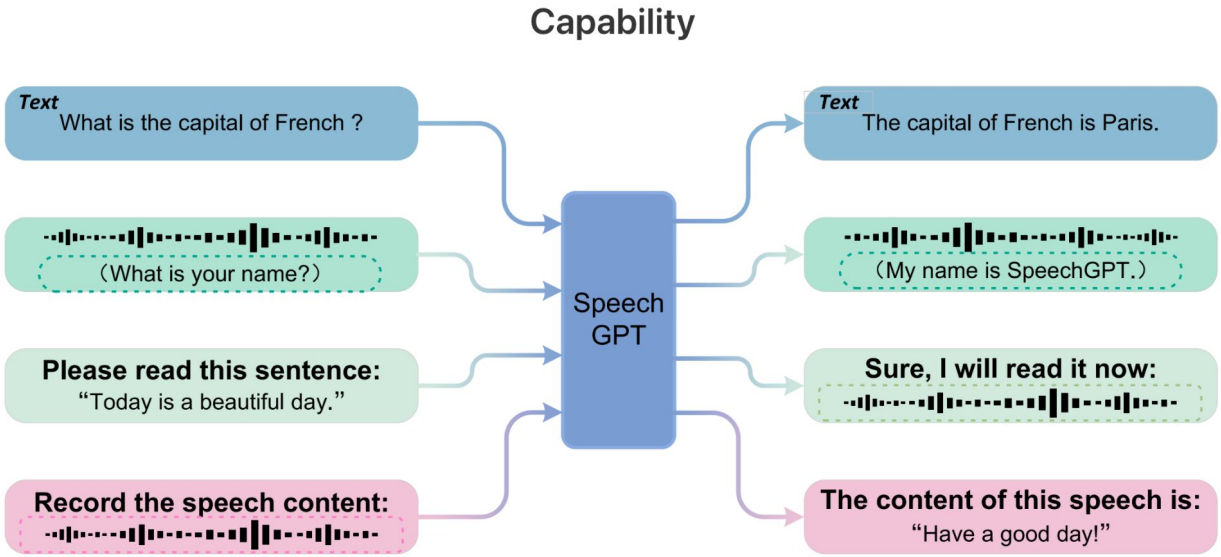
Figure 6. **Whisper is competitive with state-of-the-art commercial and open-source ASR systems in long-form transcription.** The distribution of word error rates from six ASR systems on seven long-form datasets are compared, where the input lengths range from a few minutes to a few hours. The boxes show the quartiles of per-example WERs, and the per-dataset aggregate WERs are annotated on each box. Our model outperforms the best open source model (NVIDIA STT) on all datasets, and in most cases, commercial ASR systems as well.

Whisper (Fine Tuning Assignment 2)



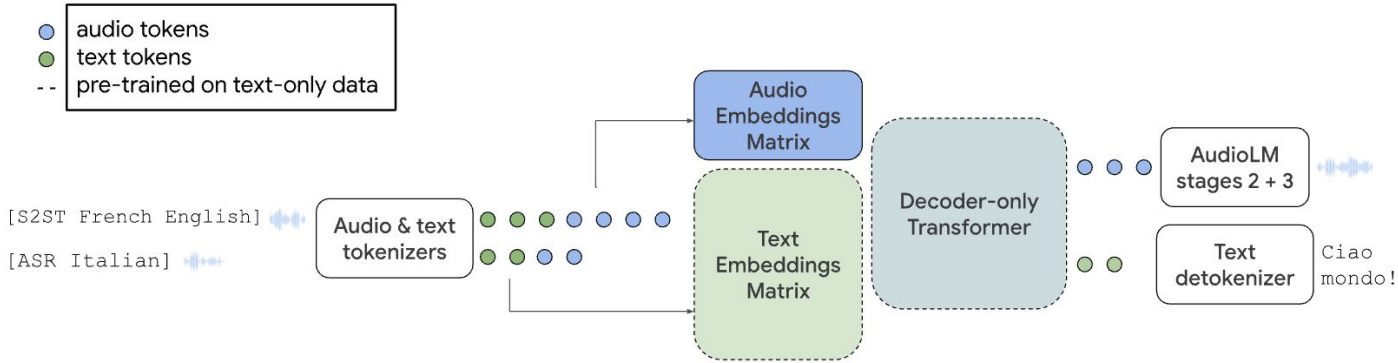
LLM + Speech (SpeechGPT, AudioPalm)

SpeechGPT



Website and Demo: https://0nutation.github.io/SpeechGPT.github.io/

AudioPalm



Website and Demo: https://google-research.github.io/seanet/audiopalm/examples/

Streaming ASRs

Dual Mode ASR: Joint Encoder + Training for Streaming & Full Context Models

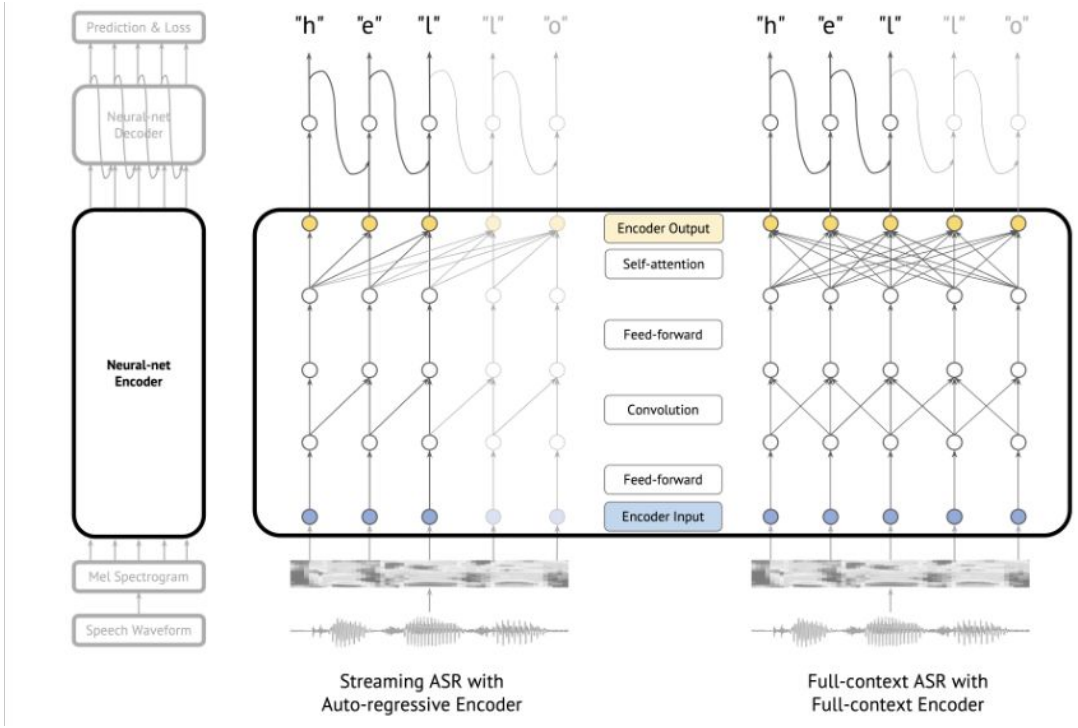


Figure: A simplified illustration of the similarity and difference between Streaming ASR and Full-context ASR networks. Modern end-to-end streaming and full-context ASR models share most of the neural architectures and training recipes in common, with the most significant difference in ASR encoder (highlighted). Streaming ASR encoders are auto-regressive models, with each prediction of the current timestep conditioned on previous ones (no future context). We show examples of feed-forward layer, convolution layer and self-attention layer in the encoder of streaming and full-context ASR respectively. With Dual-mode ASR, we unify them without parameters overhead. (Yu *et al.* 2021)

Dual Mode ASR: Joint Encoder + Training for Streaming & Full Context Models

Method	Mode	# Params (M)	Test Clean/Other WER(%)	Latency@50 (ms)	Latency@90 (ms)
LSTM-LAS	Full-context	360	2.6 / 6.0	——	——
QuartzNet-CTC	Full-context	19	3.9 / 11.3	——	——
Transformer	Full-context	29	3.1 / 7.3	——	——
Transformer	Full-context	139	2.4 / 5.6	——	——
ContextNet	Full-context	31.4	2.4 / 5.4	——	——
Conformer	Full-context	30.7	2.3 / 5.0	——	——
Transformer	Streaming	18.9	5.0 / 11.6	80	190
ContextNet	Streaming	31.4	4.5 / 10.0	70	270
Conformer	Streaming	30.7	4.6 / 9.9	140	280
ContextNet Look-ahead	Streaming	31.4	4.1 / 9.0	150	420
Dual-mode Transformer	Full-context	29	3.1 / 7.9	——	——
	Streaming		4.4 (-0.6) / 11.5 (-0.1)	-50 (-130)	30 (-160)
Dual-mode ContextNet	Full-context	31.8	2.3 / 5.3	——	——
	Streaming		3.9 (-0.6) / 8.5 (-1.5)	40 (-30)	160 (-110)
Dual-mode Conformer	Full-context	30.7	2.5 / 5.9	——	——
	Streaming		3.7 (-0.9) / 9.2 (-0.7)	10 (-130)	90 (-190)

Table 1: Summary of our results on Librispeech dataset (Panayotov et al., 2015). We report WER on TestClean and TestOther (noisy) set. Compared with standalone ContextNet and Conformer models, Dual-mode ASR models have both higher accuracy in average and better streaming latency.

Weight Sharing	Joint Training	Inplace Distillation	TestOther WER(%)	Latency@50 (ms)	Latency@90 (ms)
✔	✔	✔	8.5	40	160
✔	✔	✘	10.2 (+1.7)	120 (+80)	310 (+150)
✔	✘	✘	10.6 (+2.1)	90 (+50)	290 (+130)
✘	✔	✔	9.9 (+1.4)	50 (+10)	210 (+50)

Table 2: Ablation studies of weight sharing, joint training and inplace distillation. We report WER on TestOther (noisy) set (Panayotov et al., 2015) using ContextNet with same training settings.(Yu *et al.* 2021)

Thank You