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

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

\[
h = \text{Listen}(x)
\]

\[
P(y|x) = \text{AttendAndSpell}(h, y)
\]

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

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)
C TC + LAS Multi-Task Approach

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)

$$\mathcal{L}_{MTL} = \lambda \mathcal{L}_{CTC} + (1 - \lambda) \mathcal{L}_{Attention}$$
CTC + LAS
Multi-Task
Approach

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(λ = 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)
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

(a) Attention 1 epoch  (b) Attention 3 epoch  (c) Attention 5 epoch  (d) Attention 7 epoch  (e) Attention 9 epoch

(f) MTL 1 epoch  (g) MTL 3 epoch  (h) MTL 5 epoch  (i) MTL 7 epoch  (j) MTL 9 epoch

**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

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Unlabeled data</th>
<th>LM</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>clean</td>
<td>other</td>
</tr>
<tr>
<td>10 min labeled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete BERT [4]</td>
<td>LS-960</td>
<td>4-gram</td>
<td>15.7</td>
<td>16.3</td>
</tr>
<tr>
<td>BASE</td>
<td>LS-960</td>
<td>4-gram</td>
<td>8.9</td>
<td>9.1</td>
</tr>
<tr>
<td>LARGE</td>
<td>LS-960</td>
<td>4-gram</td>
<td>6.6</td>
<td>6.9</td>
</tr>
<tr>
<td></td>
<td>LV-60k</td>
<td>Transf.</td>
<td>6.6</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transf.</td>
<td>4.6</td>
<td>4.8</td>
</tr>
</tbody>
</table>

| 1h labeled | | | | |
| Discrete BERT [4] | LS-960 | 4-gram | 8.5 | 9.0 |
| BASE | LS-960 | 4-gram | 5.0 | 5.5 |
| LARGE | LS-960 | 4-gram | 3.8 | 4.0 |
|       | LV-60k | Transf. | 3.8 | 3.9 |
|       |         | Transf. | 2.9 | 2.9 |

| 10h labeled | | | | |
| Discrete BERT [4] | LS-960 | 4-gram | 5.3 | 5.9 |
| Iter. pseudo-labeling [58] | LS-960 | 4-gram | 23.51 | 24.37 |
|       | LV-60k | Transf. | 24.58 | 26.02 |
|       |         | Transf. | 17.00 | 18.03 |
|       |         | Transf. | 19.34 | 19.92 |

| 100h labeled | | | | |
| Hybrid DNN/HMM [34] | - | 4-gram | 5.0 | 5.8 |
| TTS data Augm. [30] | - | LSTM | 19.5 | 18.6 |
| Discrete BERT [4] | LS-960 | 4-gram | 4.0 | 4.3 |
| Iter. pseudo-labeling [58] | LS-860 | 4-gram | 4.0 | 4.3 |
|       | LV-60k | Transf. | 10.9 | 12.1 |
|       |         | Transf. | 7.97 | 9.59 |
|       |         | Transf. | 3.19 | 3.72 |
| Noisy student [42] | LS-860 | 4-gram | 3.9 | 4.2 |
|       | LV-60k | LSTM | 8.8 | 8.6 |
| BASE | LS-960 | 4-gram | 2.7 | 3.4 |
| LARGE | LS-960 | 4-gram | 2.2 | 2.6 |
|       | LV-60k | Transf. | 2.1 | 2.3 |
|       |         | Transf. | 4.0 | 5.0 |

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 울라 내려다보면 너무나 넓고 넓은 …”
- 언덕 위에 울라 내려다보면 너무나 넓고 넓은 …

No speech
- (background music playing)
- Ø

Sequence-to-sequence learning

Transformer Encoder Blocks
- MLP
- self attention

Transformer Decoder Blocks
- MLP
- cross attention
- self attention

Log-Mel Spectrogram

Tokens in Multitask Training Format

(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)

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

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)

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)
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)

*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.
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

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.

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