CSAIL

Massachusetts Institute of Technology

Generative Pre-Training for Speech with Autoregressive Predictive Coding

Yu-An Chung   James Glass

Computer Science and Artificial Intelligence Laboratory

Massachusetts Institute of Technology

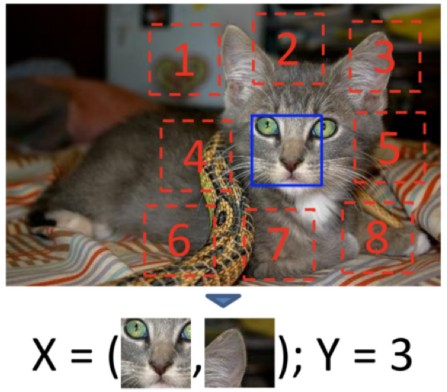
Cambridge, MA 02139, USA

ICASSP 2020

Self-supervised learning background

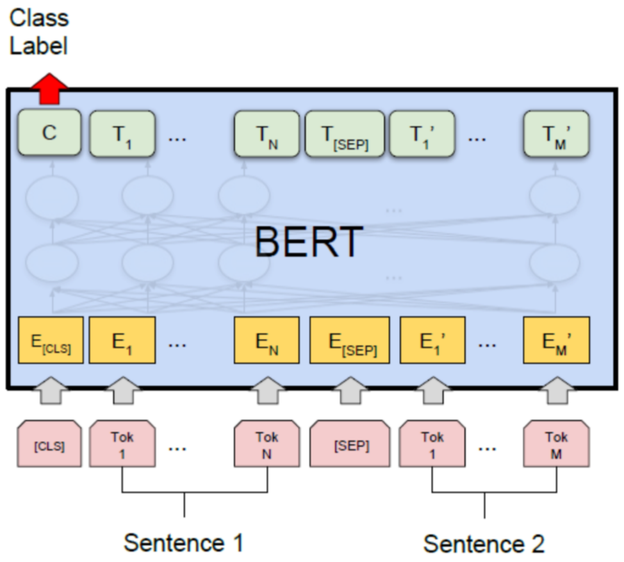
- What is self-supervised learning?
  - A form of unsupervised learning where the data itself provides supervision
  - In general, the goal is to predict some part of the data from any other part of it
  - Can leverage large quantities of unlabeled data → cheaper data and richer representations

- Very successful in Vision and NLP
  - Vision (pretext tasks)
    - Colorization
    - Image patches relationship prediction
  - NLP (pre-training)
    - Masked LM (BERT)
    - Autoregressive LM (GPT)
    - Permutation LM (XLNet)



Relative location prediction

[Doersch et al., 2015]



BERT

[Devlin et al., 2019]

Self-supervised approaches for speech (incomprehensive)

- Future prediction
  - To predict future audio features from the historical ones
    - Contrastive predictive coding (CPC) [Oord et al., 2018]
    - Autoregressive predictive coding (APC) [Chung et al., 2019]
    - wav2vec [Schneider et al., 2019]
- Mask prediction
  - To predict masked part of the input audio signals
    - Mockingjay [Liu et al., 2020]
    - Masked reconstruction [Wang et al., 2020]
- Multiple self-supervised tasks at the same time
  - Ideally, solving each task contributes prior knowledge into the representation
    - Problem-agnostic speech encoder (PASE) [Pascual et al., 2019]

What this work is about

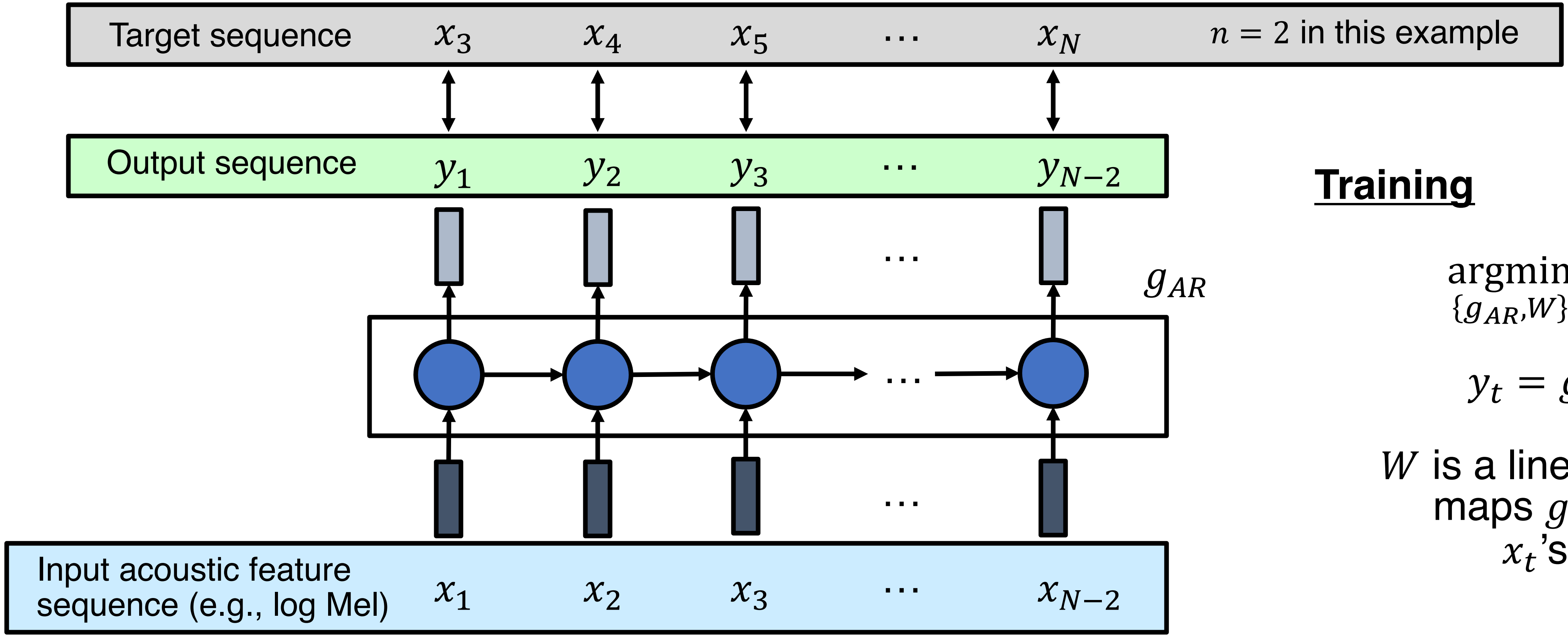
- In our previous work (Chung et al., 2019), we:
  - Proposed autoregressive predictive coding (APC)
  - Used RNNs as the backbone architecture
  - Experimented on toy tasks such as phonetic classification

- In this work, we further explore APC by:
  - Replacing RNNs with Transformers as the backbone architecture
  - Experimenting on real-world applications such as ASR, speech translation, and speaker identification, comparing with CPC and PASE features
  - Investigating the usefulness of the representations in low-resource regime, where only small amounts of labeled speech data are available

APC is a ***simple*** yet ***effective*** generative pre-training method for speech applications

Autoregressive Predictive Coding (APC)

- Given a previous context $(\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_t})$, APC tries to predict a future audio feature $\boldsymbol{x_{t+n}}$ that is $\boldsymbol{n}$ steps ahead of $\boldsymbol{x_t}$
  - Uses an autoregressive model g_{AR} to summarize history and produce output
  - $n \geq 1$ encourages g_{AR} to infer more global underlying structures of the data rather than simply exploiting local smoothness of speech signals



Training

$$\underset{\{g_{AR}, W\}}{\operatorname{argmin}} \sum_{t=1}^{N-n} |x_{t+n} - y_t|,$$

$$y_t = g_{AR}(x_1, \dots, x_t) \cdot W$$

W is a linear transformation that maps g_{AR}'s output back to x_t's dimensionality

Types of autoregressive model g_{AR}

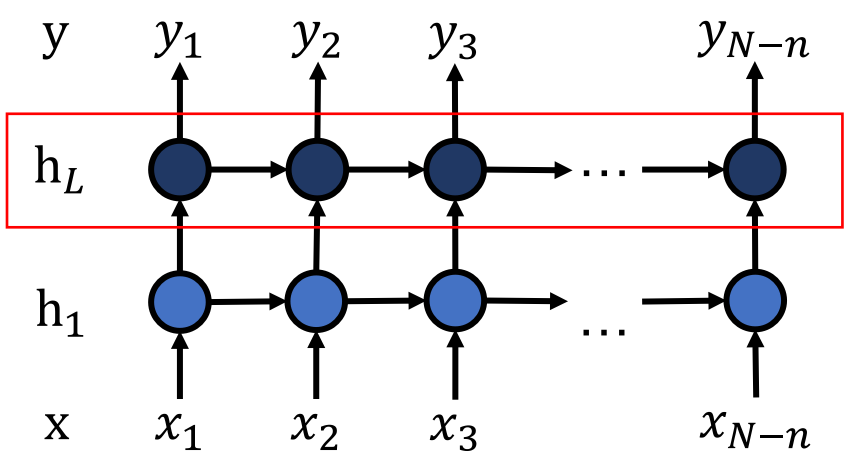
- g_{AR}
  - Input: $\mathbf{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N})$
  - Output: $\mathbf{y} = (\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_N})$
- L-layer Unidirectional RNN:

$$\mathrm{h}_0 = \mathrm{x}$$
$$\mathrm{h}_l = \mathrm{RNN}^{(l)}(h_{l-1}), \forall l \in [1, L]$$
$$\mathrm{y} = \mathrm{h}_L \cdot W$$

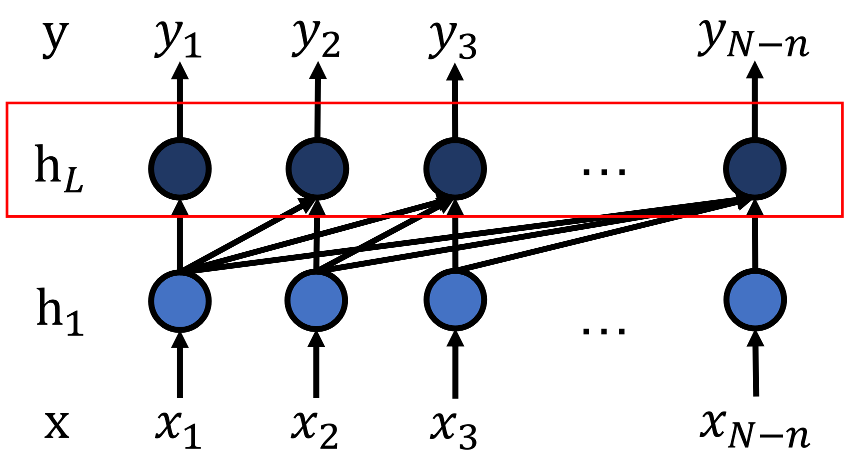
- L-layer Transformer *decoder* blocks

$$\mathrm{h}_0 = \mathrm{x} \cdot W_{in} + P(\mathrm{x})$$
$$\mathrm{h}_l = \mathrm{TRF}^{(l)}(h_{l-1}), \forall l \in [1, L]$$
$$\mathrm{y} = \mathrm{h}_L \cdot W_{out}$$

- Feature extraction: $\mathbf{h}_L$



RNN



Transformer (decoder)

- Positional encodings, W_{in} and W_{out} are not shown here
- We keep $W_{out} = W_{in}^T$ as regularization in practice

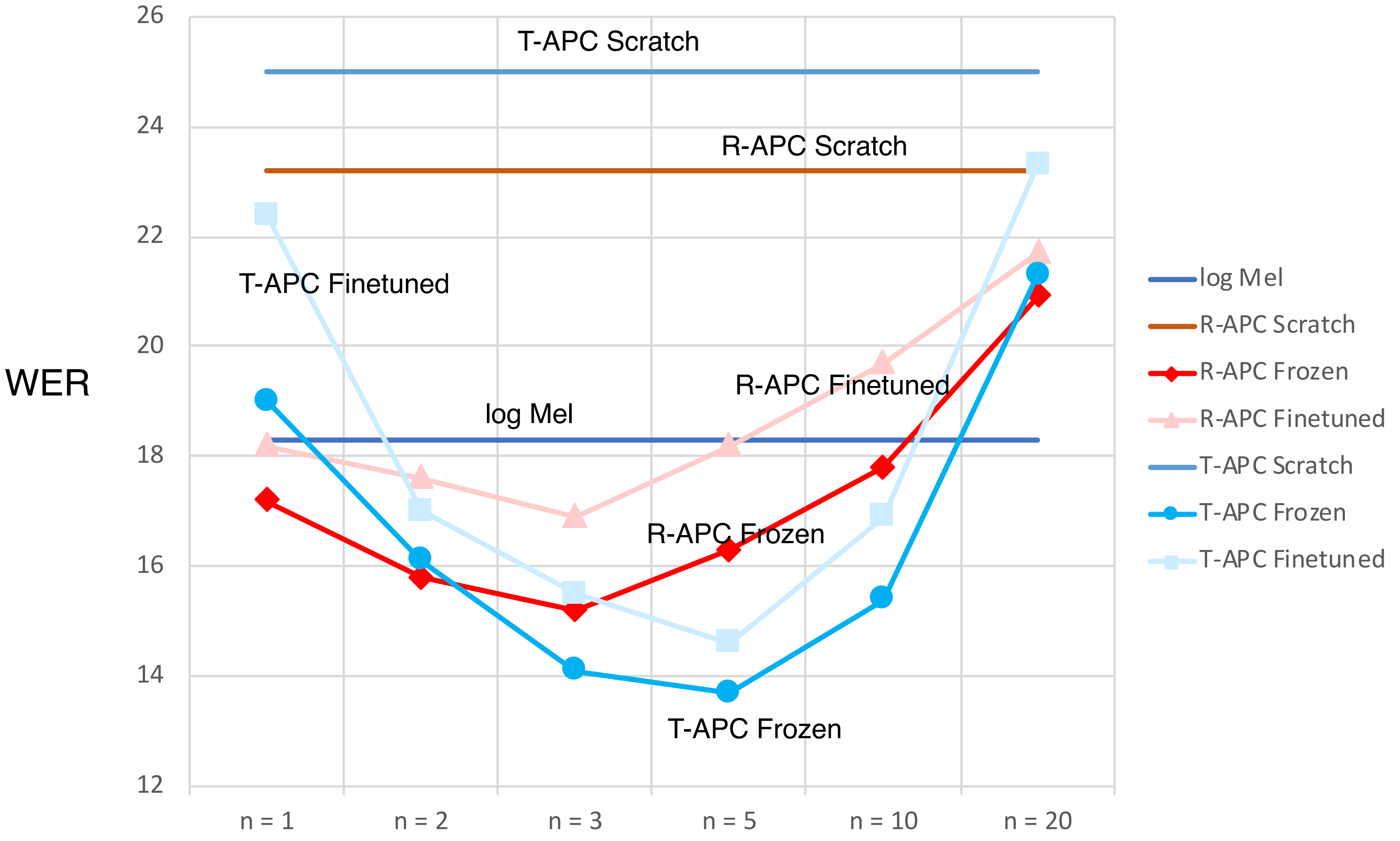
Transfer learning experiments

- Setup: pre-training + fine-tuning
- Pre-training data
  - Speech portion of the LibriSpeech 360 hours subset
  - 921 speakers
  - 80-dimensional log Mel spectrograms as input acoustic features (i.e., $x_t \in \mathbb{R}^{80}$)
  - Use extracted features to replace log Mel as new inputs to downstream models
- Considered downstream tasks
  - Speech recognition
  - Speech translation
  - Speaker identification (skipped in this talk, see paper!)
- Comparing methods
  - Contrastive predictive coding (CPC)
  - Problem-agnostic speech encoder (PASE)

Speech Recognition

- Considered dataset: Wall Street Journal
  - Training: 90% of si284 (~ 72 hours of audio)
  - Validation: 10% of si284
  - Test: dev93
- APC g_{AR}
  - RNNs: 4-layer, 512-dim GRUs
  - Transformers: 4-layer, 512-dim Transformer decoder blocks
- Downstream ASR model
  - Seq2seq with attention [Chorowski et al., 2015]
  - Beam search with beam size = 5
  - No language model rescoring

Choice of $\boldsymbol{n}$, and whether to fine-tune $\boldsymbol{g_{AR}}$



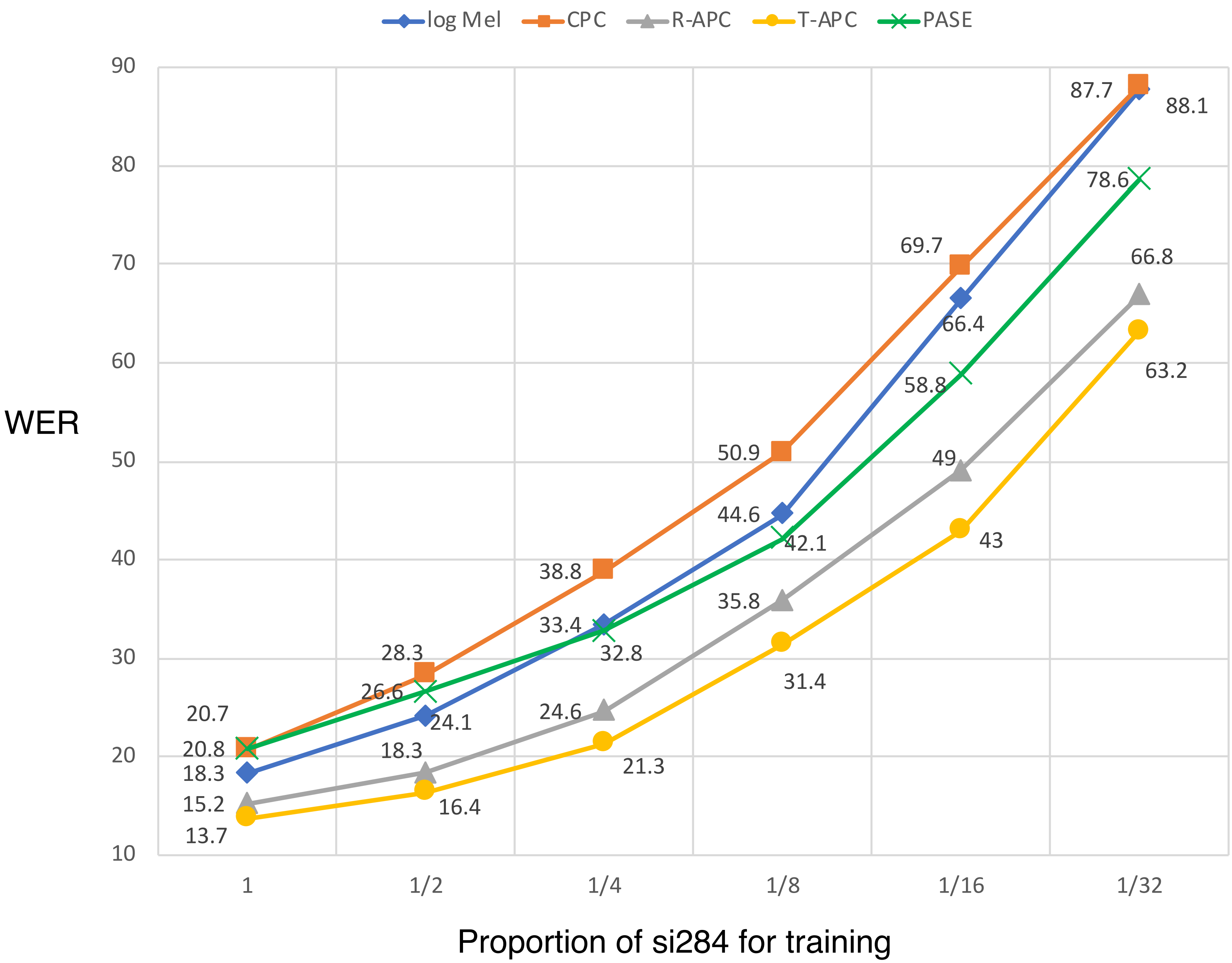
Notations

- R stands for RNN
- T stands for Transformer
- **Scratch**: g_{AR} randomly initialized and concatenate with ASR model
- **Frozen**: keep g_{AR} frozen when training ASR model
- **Finetuned**: fine-tune g_{AR} along with ASR model

Findings

- Sweet spot exists for both Frozen and Finetuned when varying n
- Scratch performance is poor, even worse than log Mel baseline
- APC outperforms log Mel most of the time
- For both R and T, Frozen outperforms Finetuned
- Will use R-APC Frozen with $n = 3$ and T-APC Frozen with $n = 5$ for the rest

APC for reducing the amount of labeled training data

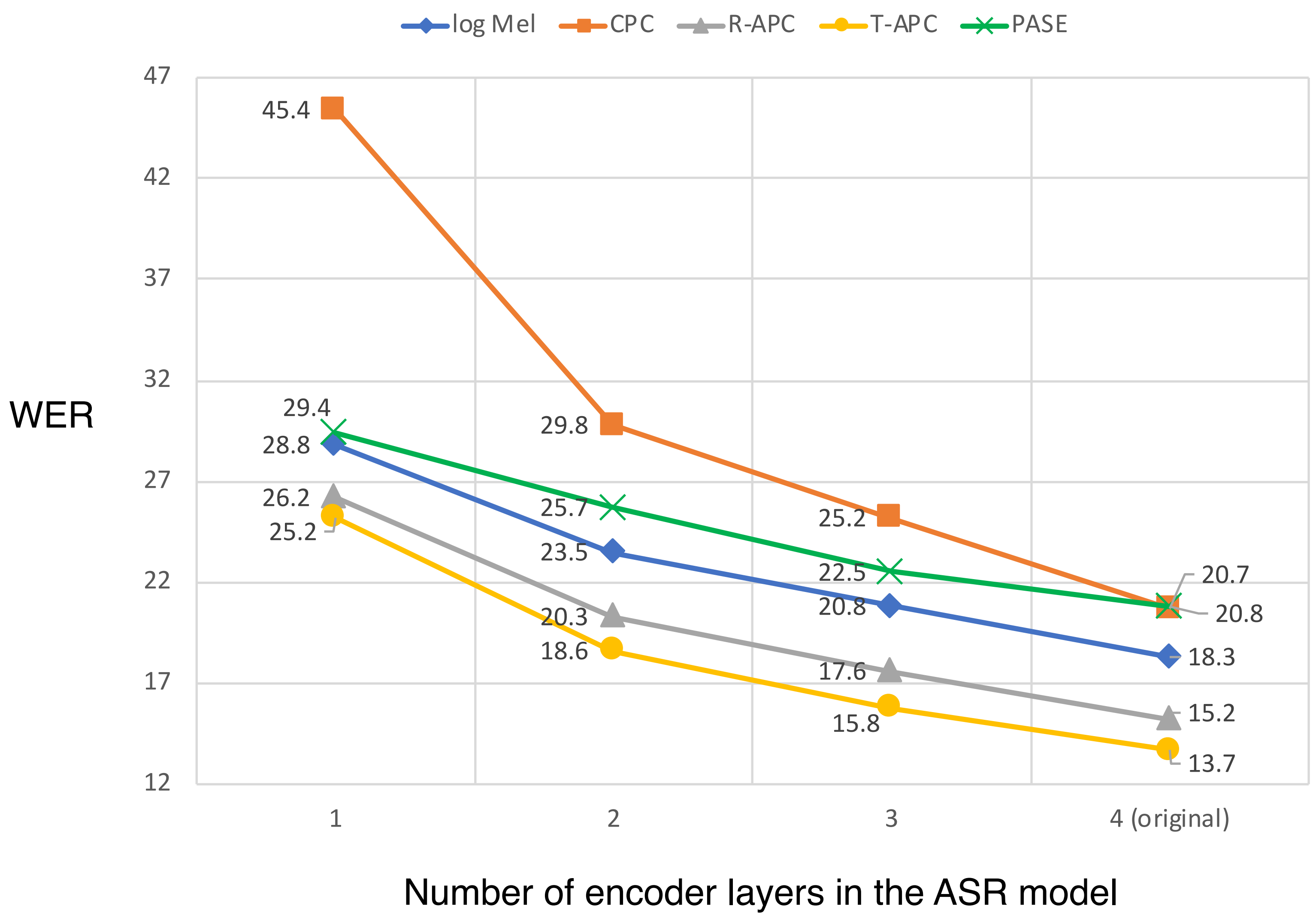


Recap: all feature extractors were pre-trained with 360 hours of LibriSpeech data; we did not fine-tune any feature extractor with the ASR model

Findings

- Full set:
  - 25% and 17% relative improvement for T-APC (13.7) and R-APC (15.2) over log Mel baseline (18.3), respectively
- As we decrease the amount of training data:
  - T-APC (yellow) and R-APC (gray) always outperform other methods
  - Gap between T-APC / R-APC and log Mel (blue) becomes larger
  - Using just half of si284, T-APC (16.4) already outperforms log Mel trained on full set (18.3)
- In the paper we also have the figure where all feature extractors were pre-trained on only 10 hrs of LibriSpeech data. **TLDR**: pre-training still helps even with just 10 hrs of pre-training data

APC for reducing downstream model size



Note: all models trained on full si284

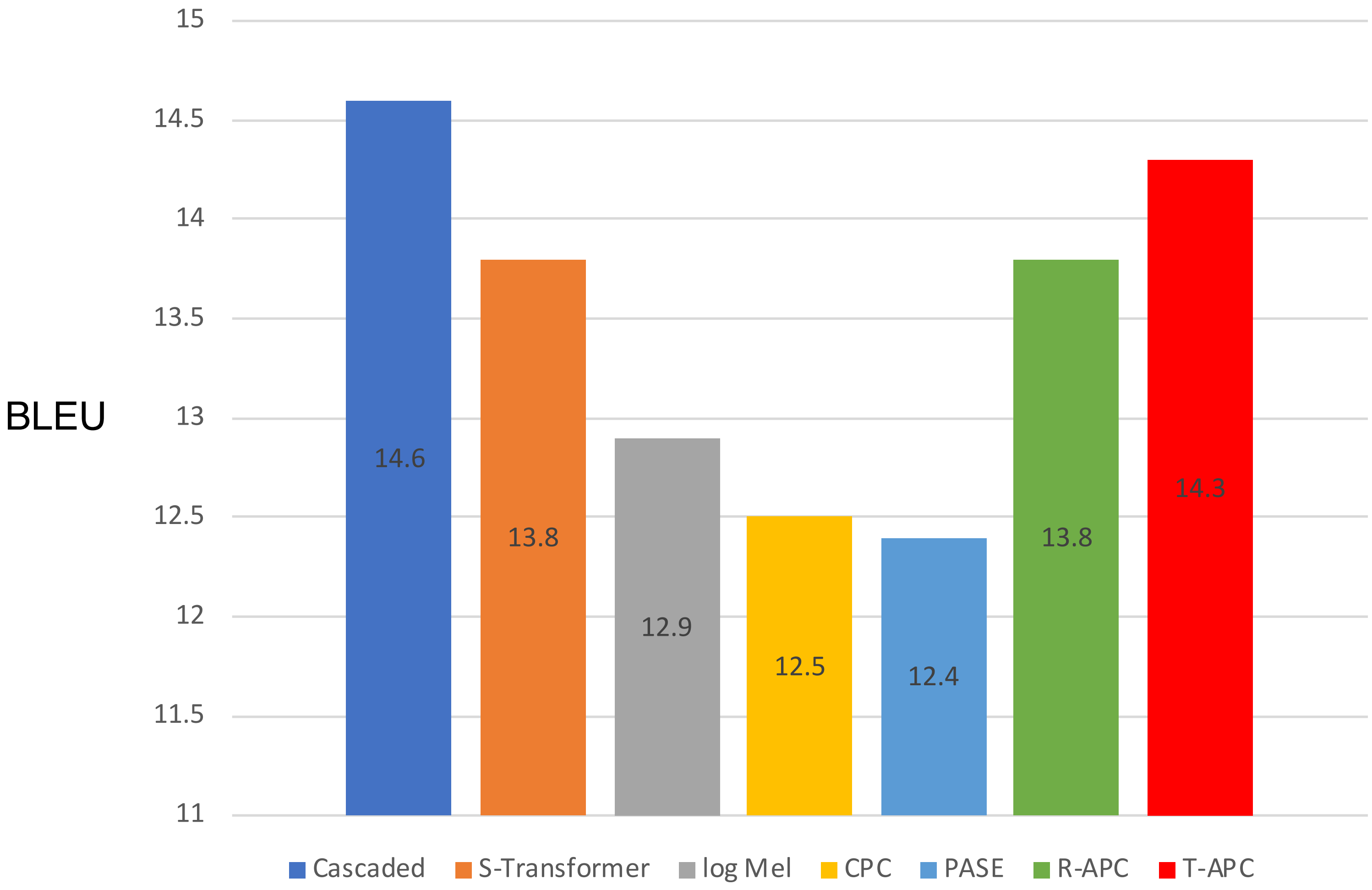
<u>Findings</u>

- T-APC (yellow) and R-APC (gray) always outperform other methods
- T-APC with just 2 layers (18.6) performs similar to log Mel with 4 layers (18.3)

Speech Translation

- Considered dataset: LibriSpeech En-Fr
  - Training set has around 100 hrs of audio
  - Report BLEU scores on test set
- Downstream speech translation model
  - RNN-based seq2seq with attention model [Berard et al., 2018]
- Also compare with two other baselines
  - Cascaded system (ASR + MT)
  - S-Transformer (end-to-end SOTA) [Di Gangi et al., 2019]

Speech translation results



Findings

- 11% and 7% relative improvement for T-APC (14.3) and R-APC (13.8) over log Mel (12.9), respectively
- T-APC (14.3) outperforms end-to-end SOTA S-Transformer with log Mel input (13.8)
  - Since S-Transformer is larger than our RNN-based seq2seq model, this result also suggests that using APC features can reduce downstream model size for speech translation
- T-APC (14.3) is close to cascaded system (14.6)

Conclusions

Empirically demonstrate that APC is a simple yet effective pre-training strategy for speech

- Can leverage large quantities of unlabeled data
- Architecture-agnostic: any autoregressive model can be used as backbone; in this paper we explored Transformer and RNN
- Learns general speech representations that can be transferred to different speech applications and outperform log Mel baseline and other self-supervised representations
- Allows to train downstream models more (labeled) data- and model-efficient

Thank you!

Questions?

Slides: http://people.csail.mit.edu/andyyuan/docs/icassp-20.generative.slides.pdf

Code: https://github.com/iamyuanchung/Autoregressive-Predictive-Coding

References

- [Doersch et al., 2015] Unsupervised visual representations learning by context prediction, ICCV
- [Devlin et al., 2019] BERT: Pre-training of deep bidirectional Transformers for language understanding, NAACL-HLT
- [Oord et al., 2018] Representation learning with contrastive predictive coding, arXiv
- [Chung et al., 2019] An unsupervised autoregressive model for speech representation learning, Interspeech
- [Schneider et al., 2019] wav2vec: Unsupervised pre-training for speech recognition, Interspeech
- [Liu et al., 2020] Mockingjay: Unsupervised speech representation learning with deep bidirectional Transformer encoders, ICASSP
- [Wang et al., 2020] Unsupervised pre-training of bidirectional speech encoders via masked reconstruction, ICASSP
- [Pascual et al., 2019] Learning problem-agnostic speech representations from multiple self-supervised tasks, Interspeech
- [Chorowski et al., 2015] Attention-based models for speech recognition, NIPS
- [Berard et al., 2018] End-to-end automatic speech translation of audiobooks, ICASSP
- [Di Gangi et al., 2019] Adapting Transformer to end-to-end spoken language translation, Interspeech