JSALT Update 13 July 2022

Pre-Training Speech Models Team

Leveraging Cascaded-SpeechCLIP for ASR: Motivation

wav2vec-U 2.0 - “Towards End-to-end Unsupervised Speech Recognition” (Liu et al. 2022)

- Can identify phonemes from a speech signal with no parallel speech-text training data
- Assumes access to a large non-parallel text corpus in the same language as the speech
- Requires a pronunciation lexicon to phonemize the non-parallel text corpus and to predict words from the generated phoneme sequences

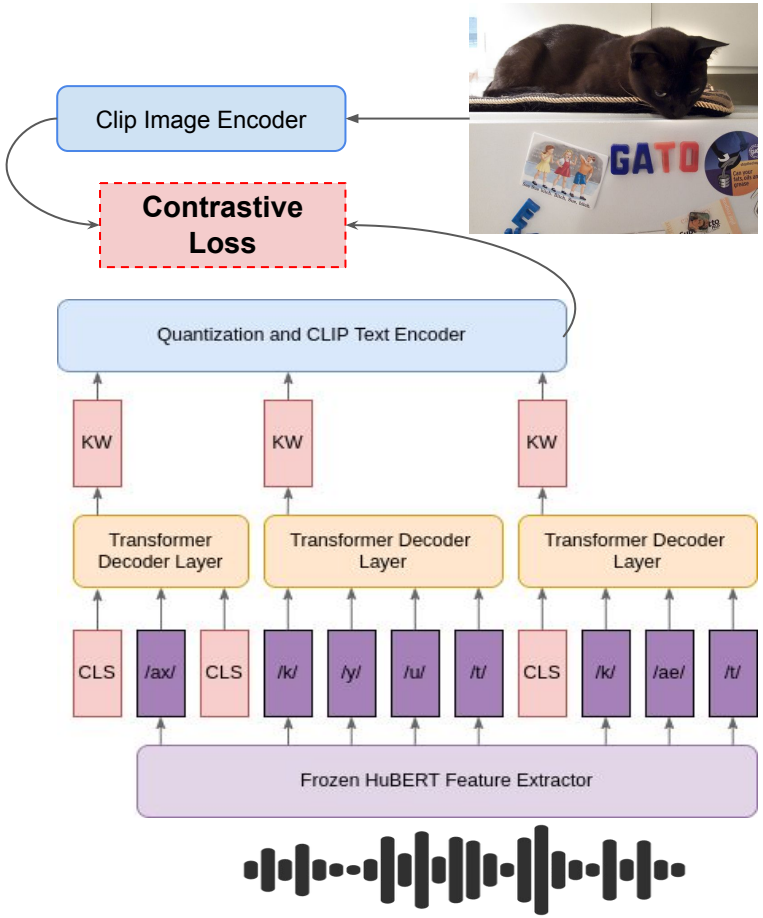
VG-HuBERT - “Word Discovery in Visually Grounded, Self-Supervised Speech Models” (Peng and Harwath, 2022)

- Can recognize words and segment them from speech audio with no text training data at all
- Skips over words without a visual referent, so great for nouns, okay for verbs, gets some prepositions, can’t recognize function words at all
- Assigns discovered words to clusters but cannot associate those clusters with a text label

Goal: Use Cascaded-SpeechCLIP (presented last week) to perform unsupervised ASR at the wordpiece level, rather than the phoneme level

Initial Architecture

- Variant of the architecture presented by David last week, with keywords predicted per segment rather than across the whole caption
  - Current model uses ground truth segmentations, but ultimate goal is to use automatic segmentation
- CLIP and HuBERT are frozen pre-trained models, while keyword prediction is trainable
  - HuBERT uses speech only
  - CLIP uses image/text caption pairs
- Learned keyword predictor maps from raw waveforms into clip wordpiece tokens
  - Output can be interpreted as an English transcription of the input speech



Initial Recognition Results

- Generated texts are not well-formed:

Correct: a man feeding a giraffe food with his mouth

Predicted: shirt hand giving have animals pizza taking giving have

- However, for each word appearing at least 10 times, we compute:

$$\text{Recognition Score}(w) = \max_{t \in \text{CLIP tokens}} \frac{\text{count}(G(w) = t)}{\text{count}(w)}$$

- Average Recognition Score: **61.46%**
- **31.15%** of words get a score > 75%

Pattern Type	Examples
Perfect Match	bathroom→bathroom (93.75%) **kitchen→kitchen (100%)** skateboard→skateboard (95.38%) vegetables→vegetables (82.76%) truck→truck (97.62%)
Semantically Related	elephants→cattle (89.29%) dark→seen (100%) **parked→stopped (92.68%)** rock→forest (100%)
Bucketing	street, train, traffic, intersection → cars meat, food, plate, sandwich → food soccer, tennis, court → frisbee woman, men, women, girls, guy → players **skis, snow, skier, skiing, ski → skiing**
Default Token ‘into’	a, on, **the**, has, white, in, his, **and**, at, with, is, their, its, up, **to**, from, that, **green**, next, front, various, **one**, some, …. → into (total: 66!)
Unexpected	benches→appears (76.92%) **snowy→following (85.71%)** glass→together (78.57%) player, baseball, bat, batter → toothbrush

Adding Form Supervision

Approach #1: Pretrained Language Model Loss (BERT)

- Map from predicted CLIP tokens to BERT token embeddings, use a pretrained BERT model with an LM head to get cross-entropy loss term
- Problem: generator is able to “cheat” these models by finding outputs with low loss that don’t correspond to correct transcriptions

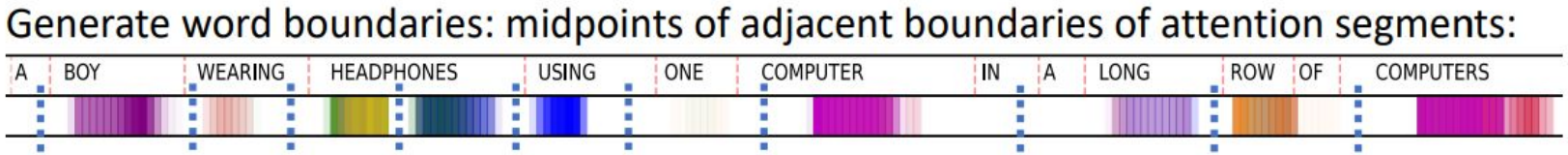
Candidate Caption	TinyBERT LM Loss	BERT-Base LM Loss
a man holding a cake that says happy birthday	13.9381	16.4122
group students enjoying among organized garden outside kitchen together together tracks	10.0006	11.4861

Approach #2: Adversarial Training (GAN)

- Inspired by wav2vec-U, but predicting word pieces instead of phonemes
  - Jiaotong has experimented with a word piece variant of wav2vec-U, but found the GAN supervision insufficient; adding our CLIP semantic loss term to Jiaotong’s model is a potentially fruitful collaboration
- Treat keyword prediction as our generator, initialize our discriminator from TinyBERT
- Non-parallel images captions are CLIP-tokenized English to use as real distribution

Planned Future Work

- Use VG-HuBERT to automatically segment incoming speech
  - Uncouple output sequence length from number of segments, to allow insertion of function words



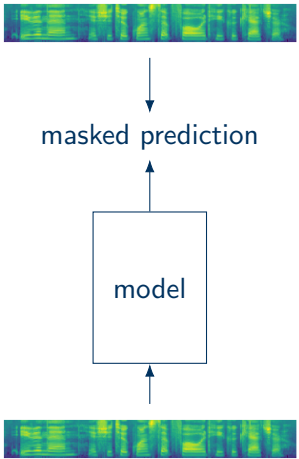
- Speech to Text translation into English with no source language text and no parallel sequences
  - The relationship between a waveform of me saying “cat”, the phoneme sequence /k/-/ae/-/t/, and the character sequence “c”-”a”-”t” is not arbitrary; a waveform of me saying “बिल्ली” cannot be easily mapped to “c”-”a”-”t”
  - The relationship between a waveform of me saying “cat” and the CLIP token ID 2368 is exactly as arbitrary as that between a waveform of me saying “बिल्ली” and the CLIP token ID 2368–a model which can learn to map “cat” to 2368 should be just as capable of learning to map “बिल्ली” to 2368

Compressing Self-Supervised Models

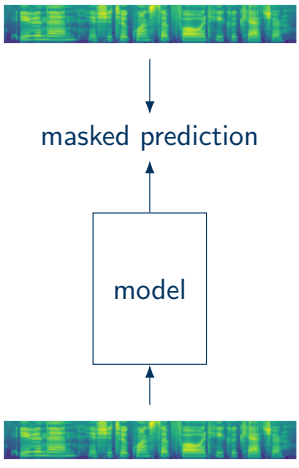
Tzu-Quan Lin[1], Tsung-Huan Yang[1], Tzu-Hsun Feng[1], Chun-Yao Chang[1], Guang-Ming Chen[1], Hao Tang[2], Hung-yi Lee[1]

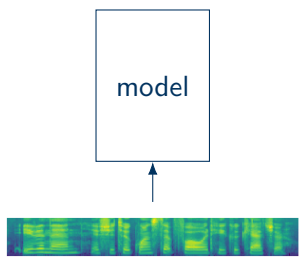
[1]National Taiwan University, [2]The University of Edinburgh

Self-supervised learning

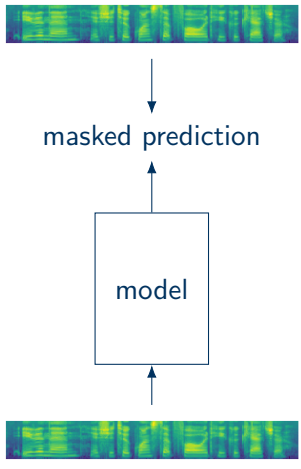


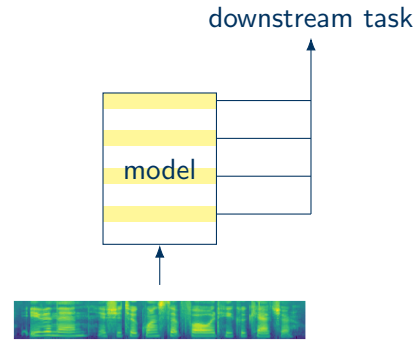
Self-supervised learning





Self-supervised learning





Goal

Making general-purpose speech models smaller

Goal

Making general-purpose speech models smaller

Goal

Making general-purpose speech models smaller

Scientific questions

$$\min_{f_{\text{small}}} \quad \Delta(f_{\text{small}}, f_{\text{large}})$$

$$\text{s.t.} \quad f_{\text{small}} \text{ has a short description length}$$

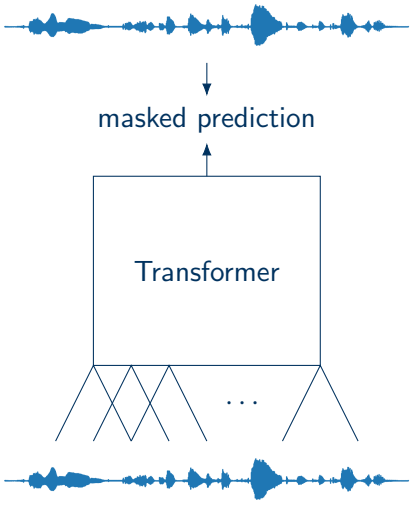
Scientific questions

$$\min_{f_{\text{small}}} \quad \Delta(f_{\text{small}}, f_{\text{large}})$$

$$\text{s.t.} \quad f_{\text{small}} \text{ has a short description length}$$

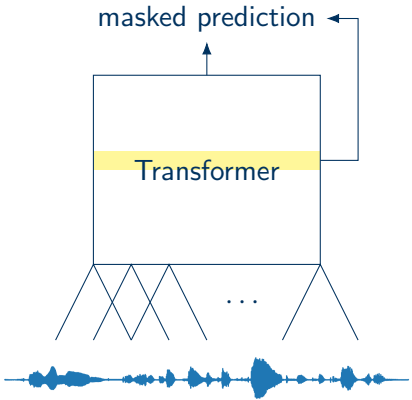
- Can we find a smaller model that performs equally well on **masked prediction**?
- Can the smaller model serve the needs of **downstream tasks**?

HuBERT



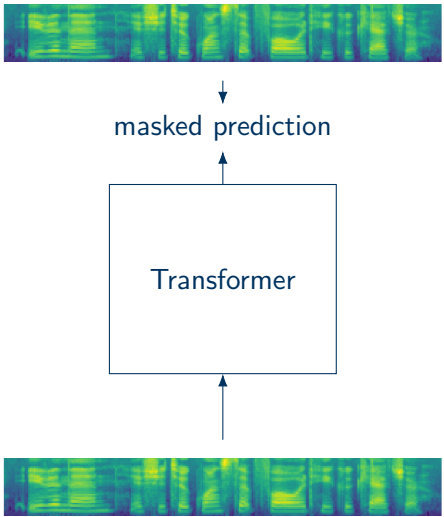
- Stage 1: predicting cluster IDs of MFCCs

HuBERT



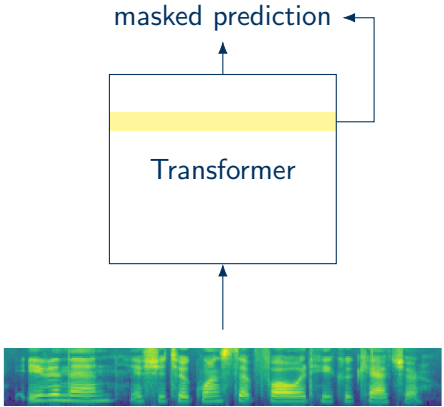
- Stage 1: predicting cluster IDs of MFCCs
- Stage 2: predicting cluster IDs of the 6th hidden layer from stage 1

MelHuBERT



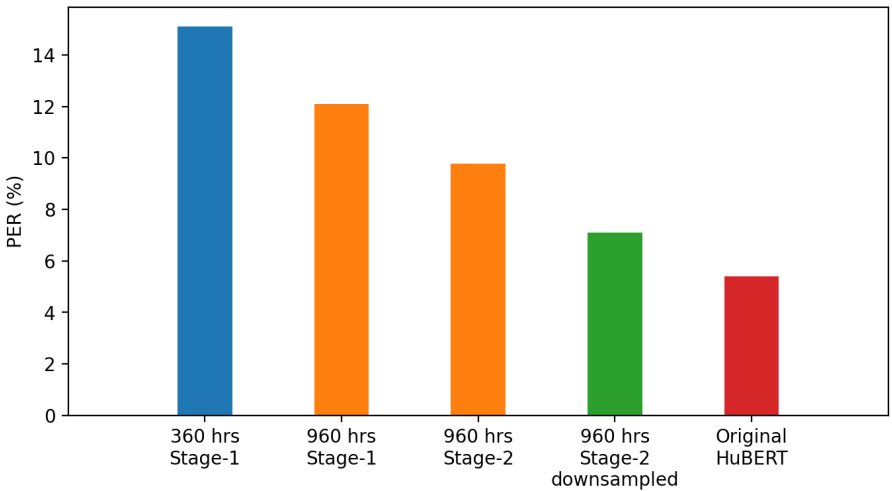
- Stage 1: predicting cluster IDs of log Mel features

MelHuBERT



- Stage 1: predicting cluster IDs of log Mel features
- Stage 2: predicting cluster IDs of the 8th hidden layer from stage 1

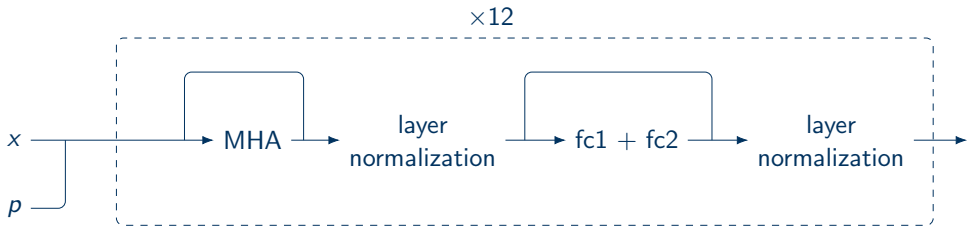
MelHuBERT progress



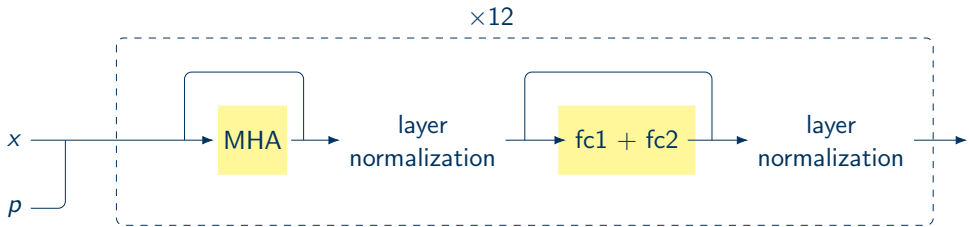
Approach

- Pruning
- Low-rank approximation
- Knowledge distillation

Transformer



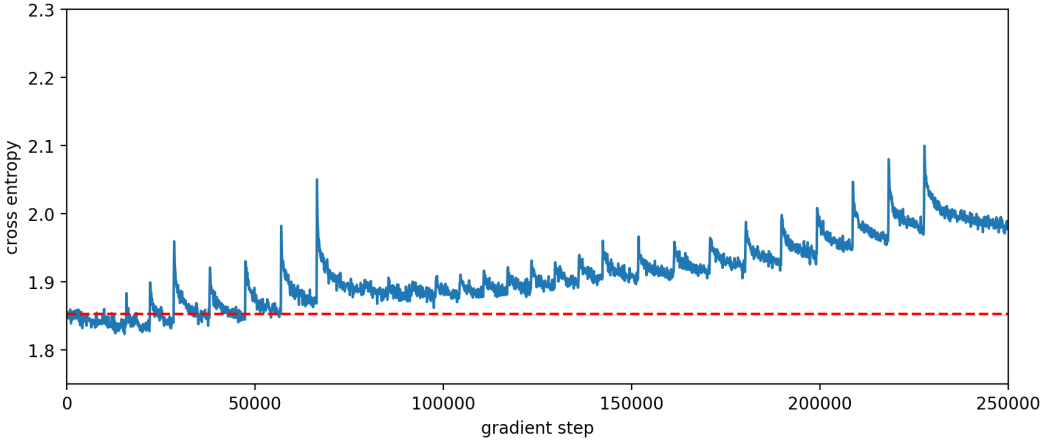
Transformer



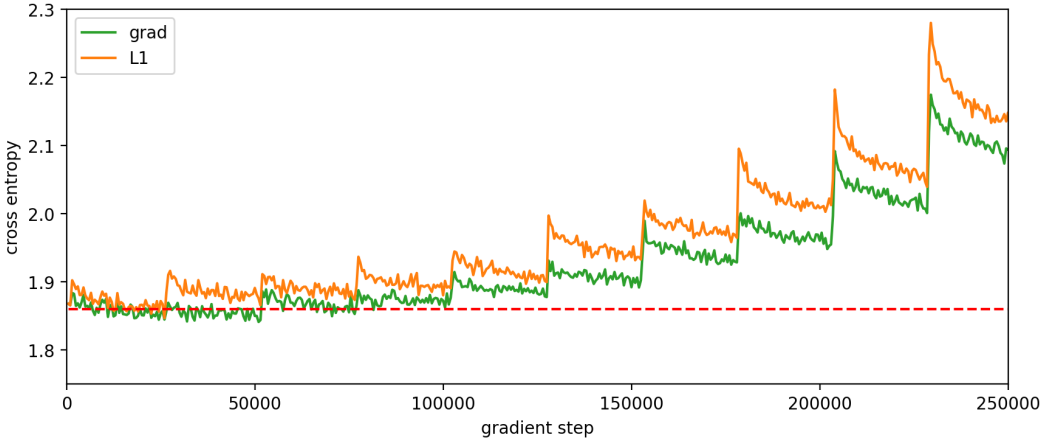
Pruning

- Prune the weights based on the ℓ_1 norm.
- Train the pruned model until convergence
- Repeat

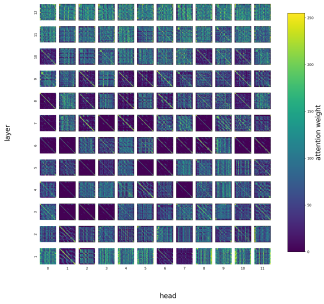
Weight Pruning

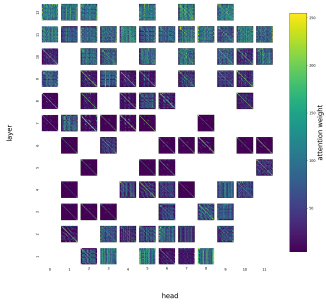


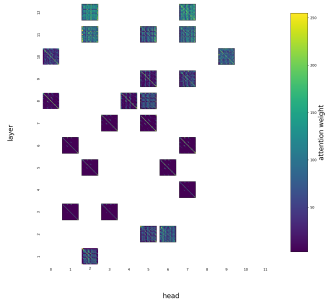
Head Pruning



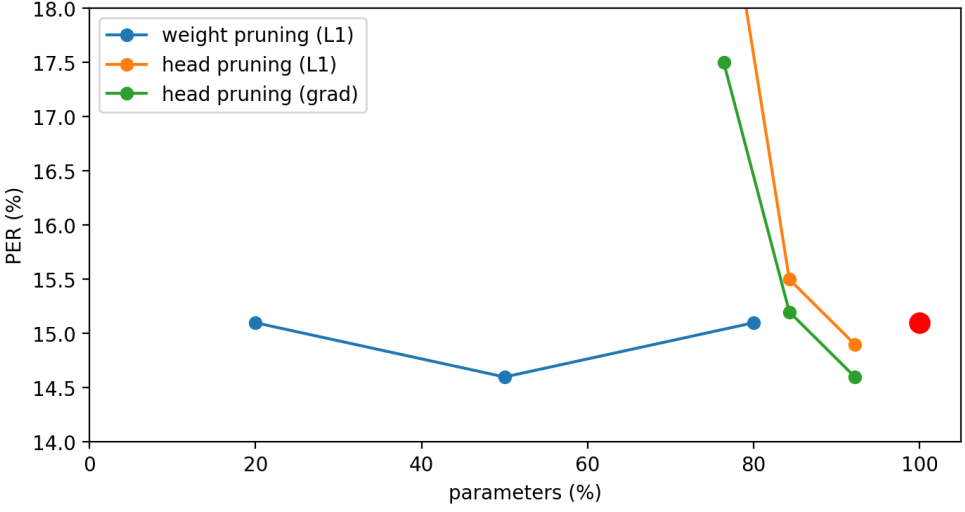
Head Pruning





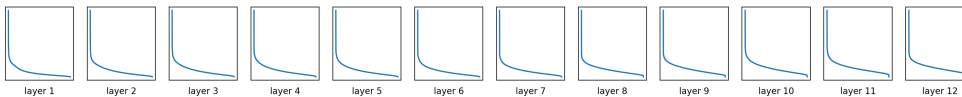


Phone Recognition

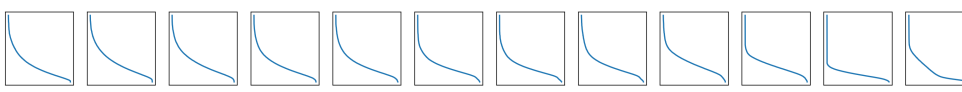


Low-rank approximation

fc1



fc2



What's left

- Finish MelHuBERT stage-2 training
- Finalize low-rank approximation
- Explore knowledge distillation
- Port techniques from the 360-hour model to the 960-hour model.
- Define several evaluation metrics.