HUBERT: HOW MUCH CAN A BAD TEACHER BENEFIT ASR PRE-TRAINING?

Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, Abdelrahman Mohamed

P. R. Sullivan

UBC NPL-DL, August 11th 2021

Table of Contents

Motivation

Self-supervised Speech Training is an ongoing competitive area of research with many models trying to replicate the impact of BERT to text-based NLP.

Goals:

- Overview of (yet another) Self-Supervised Speech Training model (HUBERT)
- Compare and contextualize HUBERT to other similar Self-Supervised models (wav2vec 2.0 [3], DeCoAR [7], Mockingjay [8] etc.)

HUBERT - High Level

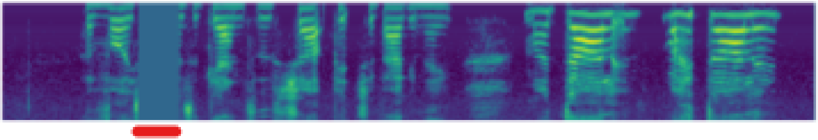
Similar to wav2vec 2.0 with Self Training [12], the goal is to internalize a language model alongside the acoustic model. Architecturally HUBERT is wav2vec 2.0 using an ensemble of bad teacher-self training, instead of contrastive loss.

Hidden Unit BERT Key ideas:

- **Masking:** Similar to wav2vec 2.0, randomly pick time steps and then mask *length* extra frames.
- **Self-Training:** Pseudo-label with frame-level predictions.
- **K-means Teachers:** Generate Pseudo-labels through k-means on either MFCC or the wav2vec feature extractor representation.
- **Multiple Teachers:** Have sets of Pseudo-labels generated by varying k in k-means clustering (*Multi-task Learning*)

Table of Contents

Masking



1

- Similar to Specaugment [10] but only across time, and other standard masking approachs [3]
- Mask applied AFTER wav2vec 2.0 feature extractor (a series of CNNs) and passed to Transformer encoder (similar to wav2vec 2.0 architecture)

[1] image from [10]

Pseudo Labeling

- Based on the wav2vec 2.0 feature extractor, we can generate per-frame pseudo-labels z using k-means (GMM could also be used), with the set of pseudo-labels Z.
- Multiple teachers (sets of pseudo-labels) can be generated using different granularity of k.

Loss Function

Instead of Contrastive Loss [11, 3, 2, 9], HUBERT applies Cross Entropy loss on frame level predictions using the pseudo labels.
Here the loss for the masked M region:

$$L_m(f; X, \{Z^{(k)}\}_k, M) = \sum_{t \in M} \sum_k \log p_f^{(k)}(z_t^{(k)} \mid \tilde{X}, t),$$

- p_f Our prediction function, see next slide.
- k the indices of the 'bad teacher'
- $\tilde{X}$ our corrupted sequence
- t timestep/frame, M masked region
- Unmasked loss is identical but $t \notin M$

Projection Function

The output from the Transformer portion of HUBERT is fed to a projection function:

$$p_f^{(k)}(c \mid \tilde{X}, t) = \frac{\exp(\mathrm{sim}(A^{(k)} o_t, e_c)/\tau)}{\sum_{c'=1}^{C} \exp(\mathrm{sim}(A^{(k)} o_t, e_{c'})/\tau)},$$

Basically the softmax distribution over the possible code generations. Calculated using the cosine similarity between the projected (using matrix A) output from HUBERT (o_t) and the different code embeddings (e_c)

To fine-tune, remove p_f and just feed O to CTC [5, 4] similar to wav2vec 2.0 [3]

Table of Contents

Experiments

- Identify balance between unmasked loss and masked loss α
- Identity impact of teacher quality on WER
- Hyperparameter Configuration
- Evaluate teacher ensembles
- Evaluate iterative teacher generation
- Compare with alternative models

Balance of prediction function

teacher	C	dev-other WER (%) $\alpha = 1.0$	dev-other WER (%) $\alpha = 0.5$	dev-other WER (%) $\alpha = 0.0$
K-means	50	18.68	31.07	94.60
	100	17.86	29.57	96.37
	500	18.40	33.42	97.66

Where α is the ratio of Masked ($\alpha = 1$) vs. Unmasked ($\alpha = 0$).
Takeaway: Computing loss on unmasked region doesn't help.

Evaluation of 1st Gen Teachers

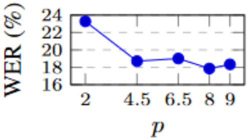
Performance of teachers (generated from 39d MFCC) is evaluated. Phone purity, Cluster Purity, and Phone-normalized Mutual Information are reported to indicate how well teachers learn frame-level phone information.

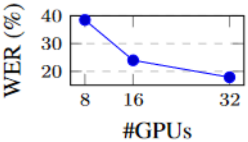
teacher	C	Phn Pur.(%)	Cls Pur.(%)	PNMI	WER (%)
Random (bottom line)	100	17.56	1.08	0.000	100.00
Chenone (top line)	8976	79.19	22.02	0.809	10.38
K-means	50	31.76	15.12	0.227	18.68
	100	33.26	9.08	0.243	17.86
	500	35.27	2.66	0.276	18.40
GMM	100	35.50	11.14	0.303	16.95

Takeaway: OK performance, not near the Supervised ASR prediction (Chenone top-line).
Ultimately use 100C k-means for simplicity.

Hyperparameter Tuning

teacher	C	dev-other WER (%) steps=100k	250k	400k	800k
K-means	50	18.68	13.65	12.40	11.82
	100	17.86	12.97	12.32	11.68
[10]	13.5k	26.6			





Takeaway: Longer training (top), larger probability to mask (bottom left), and larger batch size (bottom right) all are useful.

Ensembling Teachers

K-means with multiple k sizes vs. Taking window-3 MFCC K-means, derivative subspaces are split and quantized with 100 dim codebook, Tied vs. Untied indicate whether or not separate A matrices are used for teachers.

teacher	WER (tied)	WER (untied)
K-means {50,100}	18.17	17.81
K-means {50,100,500}	17.46	17.56
Product K-means-0-100	19.26	N/A
Product K-means-1-100	17.64	N/A
Product K-means-2-100	18.46	N/A
Product K-means-{0,1,2}-100	17.63	16.73

Takeaway: Using multiple teachers works best! Subspace approach on derivates works better!

Gen-2 Teachers

Use a k-means on MFCC to train a model, use this model as 2nd Gen teacher. Note need to grab from appropriate layer L, since last layer doesn't encode phone information.

feature	$C = 100$ / $C = 500$ Phn Pur. (%)	Cls Pur. (%)	PNMI	WER (%)
L-12	39.17 / 44.01	14.77 / 6.04	0.338 / 0.402	15.14 / 15.47
L-9	46.20 / 55.11	19.65 / 7.56	0.436 / 0.535	13.73 / 13.50
L-6	53.32 / 63.28	23.75 / 9.95	0.504 / 0.614	12.74 / 12.05
L-3	43.58 / 48.64	16.70 / 6.62	0.411 / 0.476	14.88 / 13.88
L-0	37.87 / 42.77	14.37 / 4.86	0.338 / 0.406	16.23 / 15.56

Takeaway: Bootstrapping approach seems to work well and allows for larger cluster size in second gen.

Final Results

Under a very low resource setting it performs comparable to wav2vec 2.0 [3]

D_l	dev-clean / dev-other / test-clean / test-other WER (%)		
	DiscreteBERT [10]	wav2vec 2.0 [11]	HUBERT-it2 (400k)
10m	15.7 / 24.1 / 16.3 / 25.2	8.9 / 15.7 / 9.1 / 15.6	9.1 / 15.0 / 9.7 / 15.3
1h	8.5 / 16.4 / 9.0 / 17.6	5.0 / 10.8 / 5.5 / 11.3	5.6 / 10.9 / 6.1 / 11.3
10h	5.3 / 13.2 / 5.9 / 14.1	3.8 / 9.1 / 4.3 / 9.5	3.9 / 9.0 / 4.3 / 9.4

Takeaway: Simpler than wav2vec 2.0 but still good coverage.

Table of Contents

Other Models

- DeCoAR 2[6] Uses an autoregressive (predict ahead) loss,transformer architecture, quantization and diversity loss.
- Mockingjay[8] Uses a reconstruction loss with a *Masked Acoustic Model* close to BERT's MLM
- DiscreteBERT[1] Modifies BERT to use a convolutional input instead of positional embeddings. Uses MLM. Different projection function to HUBERT.

Drawbacks of paper

- BERT is a misnomer?
- Fails to outperform wav2vec 2.0 with ST [12] in high resource setting.
- Lots of experiments that aren't actually evaluated in final setting (only uses 1 normal k-means, and 2 rounds of pre-training).
- Says they want to try to learn LM with training from continuous input, but relies on an external LM in the end!

References I

[1] BAEVSKI, A., AULI, M., AND MOHAMED, A.
Effectiveness of self-supervised pre-training for speech recognition.
arXiv preprint arXiv:1911.03912 (2019).

[2] BAEVSKI, A., SCHNEIDER, S., AND AULI, M.
vq-wav2vec: Self-supervised learning of discrete speech representations.
arXiv preprint arXiv:1910.05453 (2019).

[3] BAEVSKI, A., ZHOU, Y., MOHAMED, A.-R., AND AULI, M.
wav2vec 2.0: A framework for self-supervised learning of speech representations.
Advances in Neural Information Processing Systems 33 (2020).

References II

[4] GRAVES, A., FERNÁNDEZ, S., GOMEZ, F., AND SCHMIDHUBER, J.
Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks.
In *Proceedings of the 23rd international conference on Machine learning* (2006), pp. 369–376.

[5] GRAVES, A., AND JAITLY, N.
Towards end-to-end speech recognition with recurrent neural networks.
In *International conference on machine learning* (2014), pp. 1764–1772.

[6] LING, S., AND LIU, Y.
Decoar 2.0: Deep contextualized acoustic representations with vector quantization.
arXiv preprint arXiv:2012.06659 (2020).

References III

[7] LING, S., LIU, Y., SALAZAR, J., AND KIRCHHOFF, K.
Deep contextualized acoustic representations for semi-supervised speech recognition.
In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 6429–6433.

[8] LIU, A. T., YANG, S.-W., CHI, P.-H., HSU, P.-C., AND LEE, H.-Y.
Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders.
In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 6419–6423.

[9] OORD, A. V. D., LI, Y., AND VINYALS, O.
Representation learning with contrastive predictive coding.
arXiv preprint arXiv:1807.03748 (2018).

References IV

[10] PARK, D. S., CHAN, W., ZHANG, Y., CHIU, C.-C., ZOPH, B., CUBUK, E. D., AND LE, Q. V.
Specaugment: A simple data augmentation method for automatic speech recognition.
arXiv preprint arXiv:1904.08779 (2019).

[11] SCHNEIDER, S., BAEVSKI, A., COLLOBERT, R., AND AULI, M.
wav2vec: Unsupervised pre-training for speech recognition.
arXiv preprint arXiv:1904.05862 (2019).

[12] XU, Q., BAEVSKI, A., LIKHOMANENKO, T., TOMASELLO, P., CONNEAU, A., COLLOBERT, R., SYNNAEVE, G., AND AULI, M.
Self-training and pre-training are complementary for speech recognition.
arXiv preprint arXiv:2010.11430 (2020).