CS 224S / LINGUIST 285
Spoken Language Processing

Andrew Maas
Stanford University
Spring 2022

Lecture 13: Foundation models and SpeechBrain training

Outline

- Foundation models
  - Wav2Vec 2.0
  - HuBERT
  - XLS-R cross-lingual features
  - Working example
- SpeechBrain homework overview

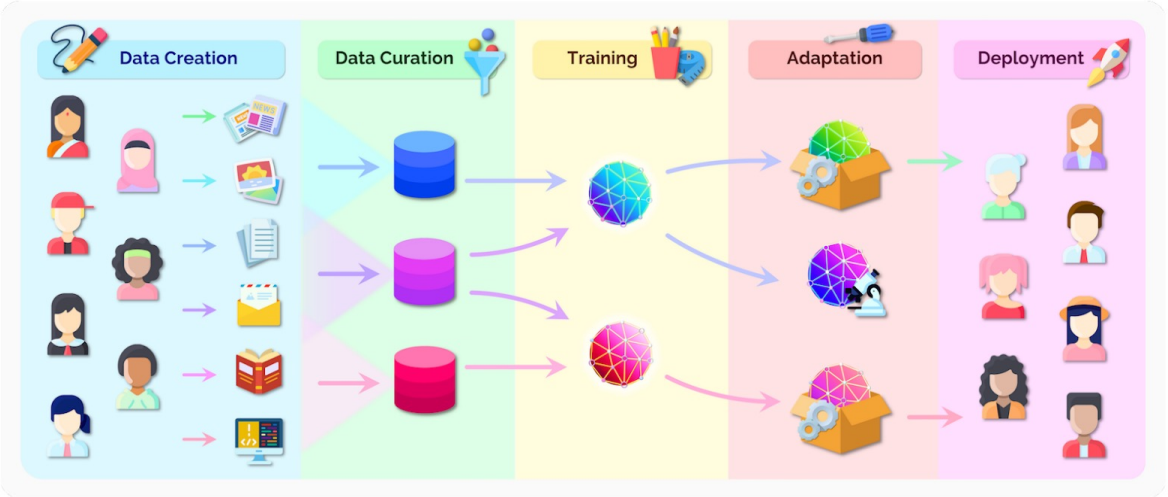
Reminders

- Homework 3 due **Wednesday** 11:59pm
  - Some challenges running part 5+6. Skip those if you like.
- Homework 4 out today. 2 Colab notebooks
  - Running inference + fine tuning SpeechBrain ASR model
  - Use your own audio with a voice cloning toolkit

- This Thursday 5/12 is our second guest lecture!
  - **Catalin Voss**. Building speech recognition systems for child reading products
  - *Lecture is Zoom only! No one in lecture hall*
    - 1% of your final grade for the course for just showing up!
    - If you can't attend synchronously, ask a question in advance on Ed
- Andrew OH on Zoom this Thursday. Link on Canvas

Foundation models

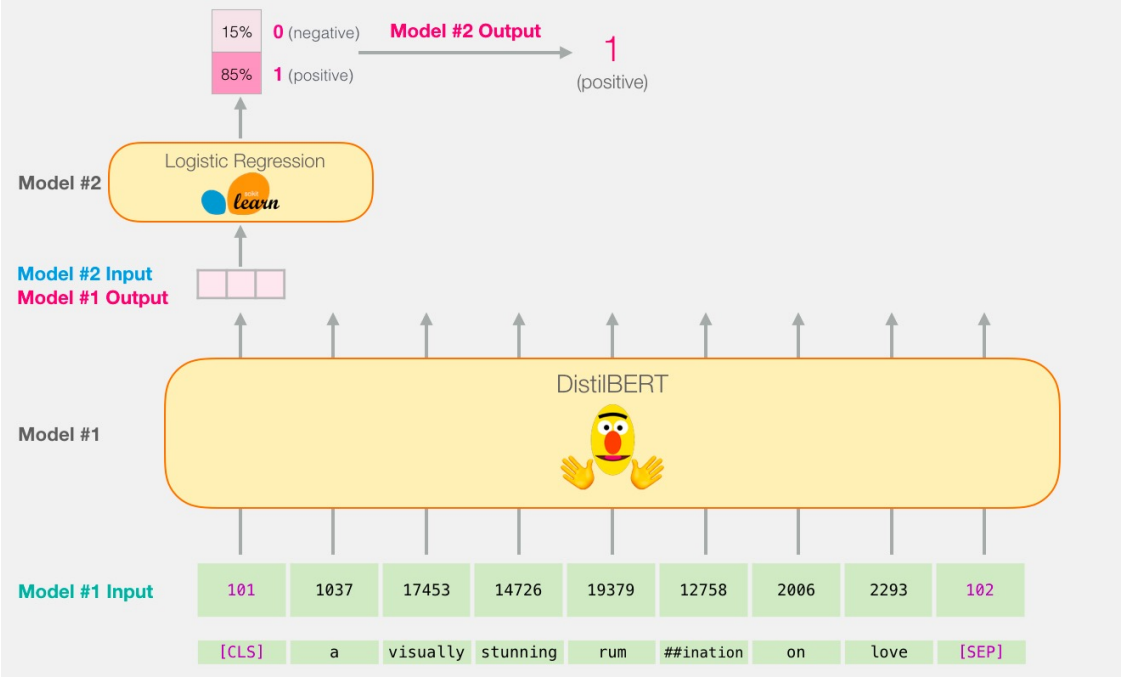
- Train large (neural) model on lots of data to cover input variation, use easy to impute labels. (self-supervision)
- Use embeddings from pre-trained model as features
  - Can collapse variable-length inputs to single vector
  - Depending on training objective, can encode rich attributes of audio



Stanford HAI Foundation Models report

Foundation models

- Use as encoder for variable-length audio
- Different models can encode speech, multiple languages, etc.
- Basic intuition same as in NLP:



Visual BERT guide

Foundation models

- Using Wav2Vec 2.0 as “standard” example
- Self-supervised. Pre-trained models available.
- Convolutional layers reduce number of inputs.
  -> Transformer encoder

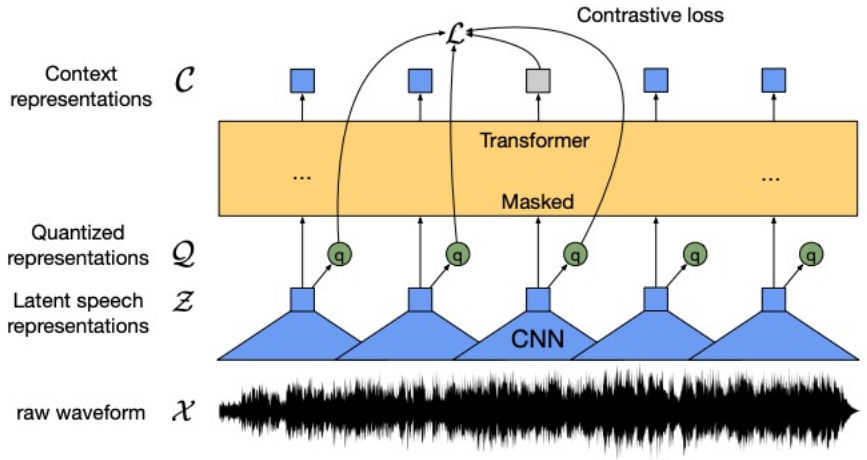


Figure 1: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units.

(Baevski et al, 2020)

Wav2Vec 2.0: Loss function

- Combination of predicting masked value + diversity loss

$$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d$$

Contrastive Loss. Given context network output $\mathbf{c}_t$ centered over masked time step t, the model needs to identify the true quantized latent speech representation $\mathbf{q}_t$ in a set of $K+1$ quantized candidate representations $\tilde{\mathbf{q}} \in \mathbf{Q}_t$ which includes $\mathbf{q}_t$ and K distractors [23, 54]. Distractors are uniformly sampled from other masked time steps of the same utterance. The loss is defined as

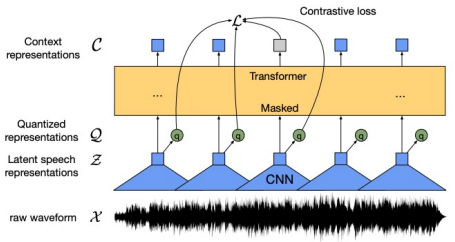
$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)} \quad (3)$$

Diversity Loss. The contrastive task depends on the codebook to represent both positive and negative examples and the diversity loss $\mathcal{L}_d$ is designed to increase the use of the quantized codebook representations [10]. We encourage the equal use of the V entries in each of the G codebooks by maximizing the entropy of the averaged softmax distribution $\mathbf{l}$ over the codebook entries for each

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v}$$

3.3 Fine-tuning

Pre-trained models are fine-tuned for speech recognition by adding a randomly initialized linear projection on top of the context network into C classes representing the vocabulary of the task [4]. For Librispeech, we have 29 tokens for character targets plus a word boundary token. Models are optimized by minimizing a CTC loss [14] and we apply a modified version of SpecAugment [41] by masking to time-steps and channels during training which delays overfitting and significantly improves the final error rates, especially on the Libri-light subsets with few labeled examples.



(Baevski et al, 2020)

Wav2Vec 2.0: Masking & quantization

- Masking of input features during training. Pick an index, mask out inputs for M (randomly chosen) steps after that index
- Contrastive loss predicts correct quantized vector q_t from among K distractors (masked examples from the same utterance).

Contrastive Loss. Given context network output $\mathbf{c}_t$ centered over masked time step t, the model needs to identify the true quantized latent speech representation $\mathbf{q}_t$ in a set of $K+1$ quantized candidate representations $\tilde{\mathbf{q}} \in \mathbf{Q}_t$ which includes $\mathbf{q}_t$ and K distractors [23, 54]. Distractors are uniformly sampled from other masked time steps of the same utterance. The loss is defined as

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)} \quad (3)$$

- Quantization choices define q_t
- Quantization can be k-means. Chosen empirically to capture audio. Similar to HMM triphone clustering or BPE.

(Baevski et al, 2020)

Wav2Vec 2.0: ASR performance

- Great pre-training for end-to-end ASR models
- Performance ceiling effects on this dataset likely present

Table 2: WER on Librispeech when using all 960 hours of labeled data (cf. Table 1).

Model	Unlabeled data	LM	dev		test	
			clean	other	clean	other
Supervised						
CTC Transf [51]	-	CLM+Transf.	2.20	4.94	2.47	5.45
S2S Transf. [51]	-	CLM+Transf.	2.10	4.79	2.33	5.17
Transf. Transducer [60]	-	Transf.	-	-	2.0	4.6
ContextNet [17]	-	LSTM	1.9	3.9	1.9	4.1
Conformer [15]	-	LSTM	2.1	4.3	1.9	3.9
Semi-supervised						
CTC Transf. + PL [51]	LV-60k	CLM+Transf.	2.10	4.79	2.33	4.54
S2S Transf. + PL [51]	LV-60k	CLM+Transf.	2.00	3.65	2.09	4.11
Iter. pseudo-labeling [58]	LV-60k	4-gram+Transf.	1.85	3.26	2.10	4.01
Noisy student [42]	LV-60k	LSTM	1.6	3.4	1.7	3.4
This work						
LARGE - from scratch	-	Transf.	1.7	4.3	2.1	4.6
BASE	LS-960	Transf.	1.8	4.7	2.1	4.8
LARGE	LS-960	Transf.	1.7	3.9	2.0	4.1
	LV-60k	Transf.	1.6	3.0	1.8	3.3

(Baevski et al, 2020)

Wav2Vec 2.0: Low resource ASR

- Viable WER with just 10 mins – 1 hr of training data!

Table 1: WER on the Librispeech dev/test sets when training on the Libri-light low-resource labeled data setups of 10 min, 1 hour, 10 hours and the clean 100h subset of Librispeech. Models use either the audio of Librispeech (LS-960) or the larger LibriVox (LV-60k) as unlabeled data. We consider two model sizes: BASE (95m parameters) and LARGE (317m parameters). Prior work used 860 unlabeled hours (LS-860) but the total with labeled data is 960 hours and comparable to our setup.

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

(Baevski et al, 2020)

Hidden Unit BERT (HuBERT)

- Explicitly create and predict hidden cluster vectors
- More directly optimizes masked label prediction than wav2vec

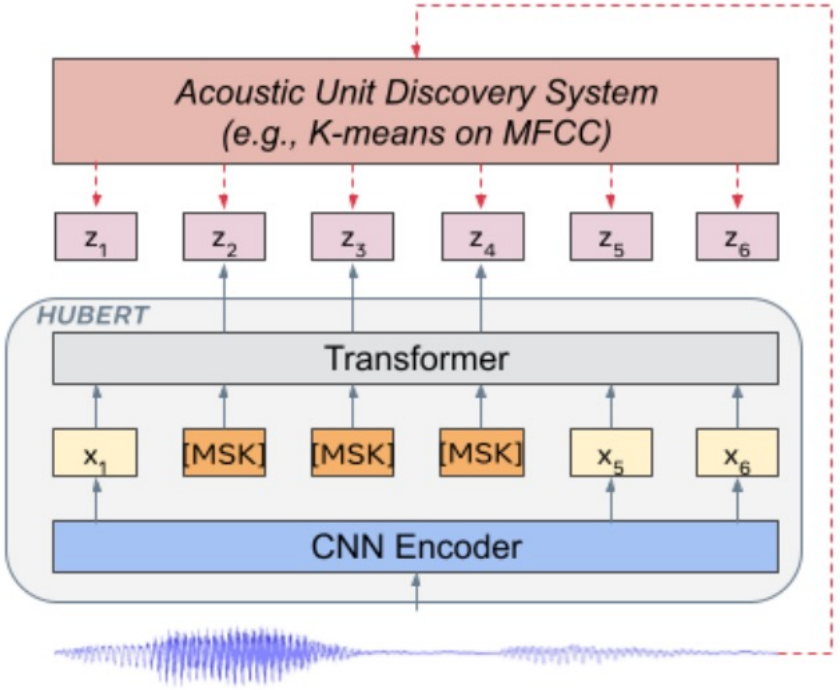


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

(Hsu et al., 2021)

HuBERT: Predicting quantized vectors

- Procedure to iteratively update clustering / quantization
- Mix predicting quantized vectors for masked + unmasked outputs

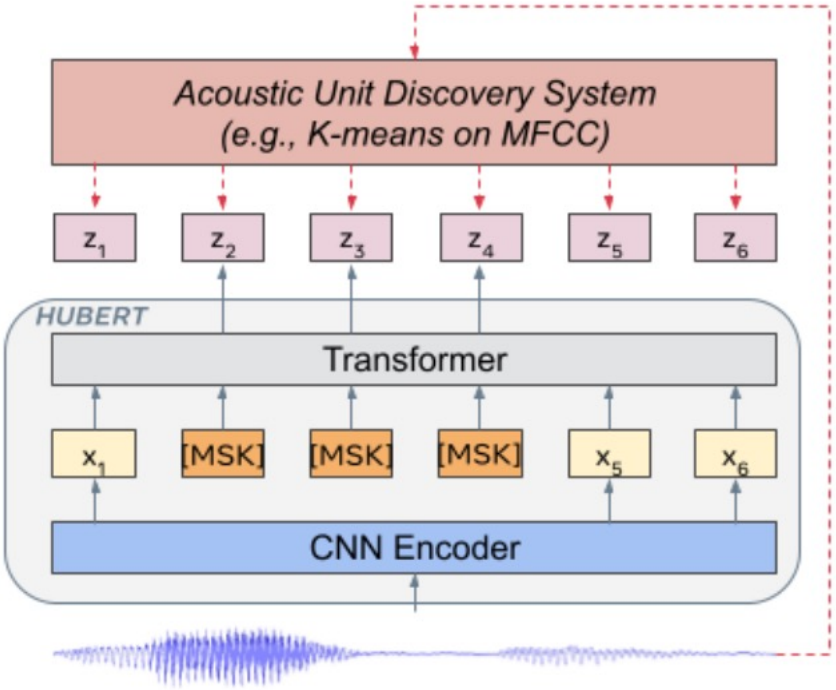


Fig. 1: The HuBERT approach predicts hidden cluster assignments of the masked frames (y_2, y_3, y_4 in the figure) generated by one or more iterations of k-means clustering.

		BASE	LARGE	X-LARGE
CNN Encoder	strides	5, 2, 2, 2, 2, 2, 2		
	kernel width	10, 3, 3, 3, 3, 2, 2		
	channel	512		
Transformer	layer	12	24	48
	embedding dim.	768	1024	1280
	inner FFN dim.	3072	4096	5120
	layerdrop prob	0.05	0	0
	attention heads	8	16	16
Projection	dim.	256	768	1024
Num. of Params		95M	317M	964M

TABLE I: Model architecture summary for BASE, LARGE, and X-LARGE HuBERT models

(Hsu et al., 2021)

HuBERT: A bit better on low resource

- Useful pre-training for ASR tasks. Quantized representations are somewhat easier to define for new tasks

Model	Unlabeled Data	LM	dev-clean	dev-other	test-clean	test-other
		10-min labeled				
DiscreteBERT [51]	LS-960	4-gram	15.7	24.1	16.3	25.2
wav2vec 2.0 BASE [6]	LS-960	4-gram	8.9	15.7	9.1	15.6
wav2vec 2.0 LARGE [6]	LL-60k	4-gram	6.3	9.8	6.6	10.3
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	4.6	7.9	4.8	8.2
HUBERT BASE	LS-960	4-gram	9.1	15.0	9.7	15.3
HUBERT LARGE	LL-60k	4-gram	6.1	9.4	6.6	10.1
HUBERT LARGE	LL-60k	Transformer	4.3	7.0	4.7	7.6
HUBERT X-LARGE	LL-60k	Transformer	4.4	6.1	4.6	6.8
		1-hour labeled				
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	13.8	29.1
DiscreteBERT [51]	LS-960	4-gram	8.5	16.4	9.0	17.6
wav2vec 2.0 BASE [6]	LS-960	4-gram	5.0	10.8	5.5	11.3
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	2.9	5.4	2.9	5.8
HUBERT BASE	LS-960	4-gram	5.6	10.9	6.1	11.3
HUBERT LARGE	LL-60k	Transformer	2.6	4.9	2.9	5.4
HUBERT X-LARGE	LL-60k	Transformer	2.6	4.2	2.8	4.8

(Hsu et al., 2021)

HuBERT or Wav2Vec: Great results!

- Overall ASR performance is great for pre-training + fine tuning approaches on reasonable benchmark.

Model	Unlabeled Data	LM	dev-clean	dev-other	test-clean	test-other
		10-hour labeled				
SlimIPL [54]	LS-960	4-gram + Transformer	5.3	7.9	5.5	9.0
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	5.4	13.3
DiscreteBERT [51]	LS-960	4-gram	5.3	13.2	5.9	14.1
wav2vec 2.0 BASE [6]	LS-960	4-gram	3.8	9.1	4.3	9.5
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	2.4	4.8	2.6	4.9
HUBERT BASE	LS-960	4-gram	3.9	9.0	4.3	9.4
HUBERT LARGE	LL-60k	Transformer	2.2	4.3	2.4	4.6
HUBERT X-LARGE	LL-60k	Transformer	2.1	3.6	2.3	4.0
		100-hour labeled				
IPL [12]	LL-60k	4-gram + Transformer	3.19	6.14	3.72	7.11
SlimIPL [54]	LS-860	4-gram + Transformer	2.2	4.6	2.7	5.2
Noisy Student [61]	LS-860	LSTM	3.9	8.8	4.2	8.6
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	5.0	12.1
DiscreteBERT [51]	LS-960	4-gram	4.0	10.9	4.5	12.1
wav2vec 2.0 BASE [6]	LS-960	4-gram	2.7	7.9	3.4	8.0
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	1.9	4.0	2.0	4.0
HUBERT BASE	LS-960	4-gram	2.7	7.8	3.4	8.1
HUBERT LARGE	LL-60k	Transformer	1.8	3.7	2.1	3.9
HUBERT X-LARGE	LL-60k	Transformer	1.7	3.0	1.9	3.5

(Hsu et al., 2021)

Research trend: Foundation model features

TABLE II: An overview of the recent audio self-supervised learning methods. The "speech" column distinguishes whether a method addresses speech tasks or for general purpose audio representations. The "framework" type refers to Figure 1.

Model	Speech	Input format	Framework	Encoder	Loss	Inspired by
LIM [36]	✔	raw waveform	(d)	SincNet	BCE, MINE or NCE loss	SimCLR
COLA [36]	✗	log mel-filterbanks	(d)	EfficientNet	InfoNCE loss	SimCLR
CLAR [33] (semi)	✗	raw waveform log mel-spectrogram	(d)	1D ResNet-18 ResNet-18	NT-Xent + cross-entropy	SimCLR
Fonseca et al. [36]	✗	log mel-spectrogram	(d)	ResNet, VGG, CRNN	NT-Xent loss	SimCLR
Wang et al. [88]	✗	raw waveform + log mel-filterbanks	(d)	CNN ResNet	NT-Xent loss + cross-entropy	SimCLR
BYOL-A [89]	✗	log mel-filterbanks	(b)	CNN	MSE loss	BYOL
Speech2Vec [48]	✔	mel-spectrogram	(a)	RNN	MSE loss	Word2Vec
Audio2Vec [91]	✔✗	MFCCs	(a)	CNN	MSE loss	Word2Vec
Carr [67]	✔	MFCCs	(a)	Context-free network	Fenchel-Young loss	-
Ryan [68]	✗	constant-Q transform spectrogram	(a)	AlexNet	Triplet loss	- -
Mockingjay [92]	✔	mel-spectrogram	(a)	Transformer	L1 loss	BERT
TERA [93]	✔	log mel-spectrogram	(a)	Transformer	L1 loss	BERT
Audio ALBERT [94]	✔	log mel-spectrogram	(a)	Transformer	L1 loss	BERT
DAPC [95]	✔	spectrogram	(a)	Transformer	Modified MSE loss + orthogonality penalty	BERT
PASE [96]	✔	raw waveform	(a)	SincNet + CNN	L1, BCE loss	BERT
PASE+ [97]	✔	raw waveform	(a)	SincNet + CNN + QRNN	MSE, BCE loss	BERT
CPC [40]	✔	raw waveform	(a)	ResNet + GRU	InfoNCE loss	-
CPC v2 [59]	✔	raw waveform	(a)	ResNet + Masked CNN	InfoNCE loss	-
CPC2 [98]	✔	raw waveform	(a)	ResNet + LSTM	InfoNCE loss	-
Wav2Vec [84]	✔	raw waveform	(a)	1D CNN	Contrastive loss	-
VQ-Wav2Vec [85]	✔	raw waveform	(a)	1D CNN + BERT	Contrastive loss	BERT
Wav2Vec 2.0 [81]	✔	raw waveform	(a)	1D CNN + Transformer	Contrastive loss	BERT
HuBERT [99]	✔	raw waveform	(c)	1D CNN + Transformer	Contrastive loss	BERT

(Liu *et al*, 2022)

Using HuBERT: Emotion recognition

- Pre-train with huBERT, fine tune on emotion classifier

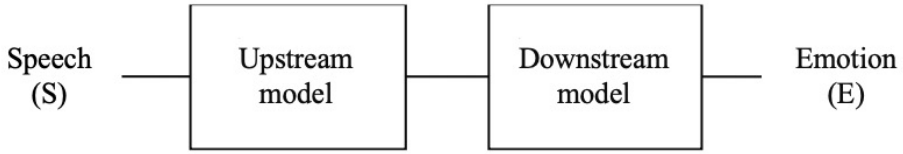


Figure 1. Proposed SER model architecture

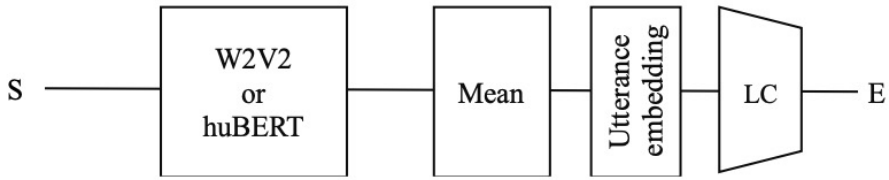
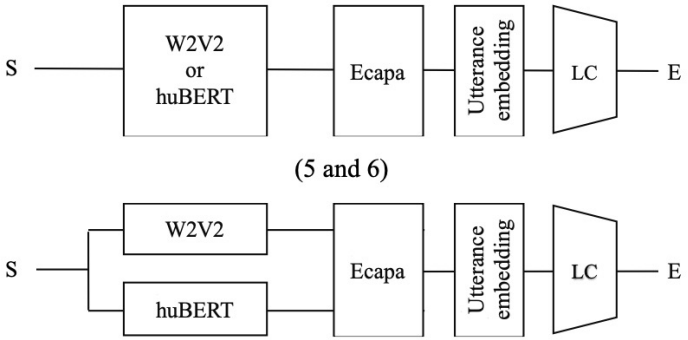


Figure 2. Upstream model fine-tuning process



(Morais et al., 2022)

Using HuBERT: Emotion recognition

- Pre-trained features much better than filterbanks
- Useful features even without fine tuning.
  See paper for ablations

#	Method	Modalities	(UACC %)
1	Sajjad et al. [23]	Audio	72.25
2	Wang et al. [24]	Audio	73.30
3	Liu et al. [25]	Audio	70.78
4	Zhao et al. [26]	Audio	71.70
5	Wu et al. [6]	Audio + Text	78.30
	Ours (exp. 7)	Audio	**77.76**

Table 2: SOTA results for SER for the cases of audio-only and audio+text input modalities.

(Morais et al., 2022)

XLS-R: Large multi-lingual wav2vec 2

- Train wav2vec 2 on many languages. Quantized outputs are language agnostic
- Publicly released encoder model on HuggingFace

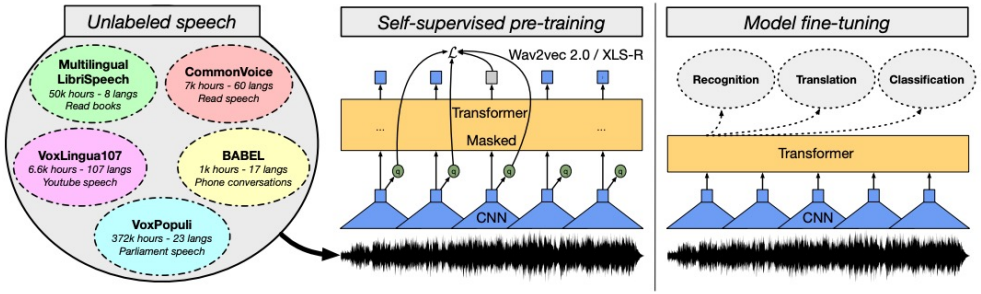


Figure 1: **Self-supervised cross-lingual representation learning.** We pre-train a large multilingual wav2vec 2.0 Transformer (XLS-R) on 436K hours of unannotated speech data in 128 languages. The training data is from different public speech corpora and we fine-tune the resulting model for several multilingual speech tasks.

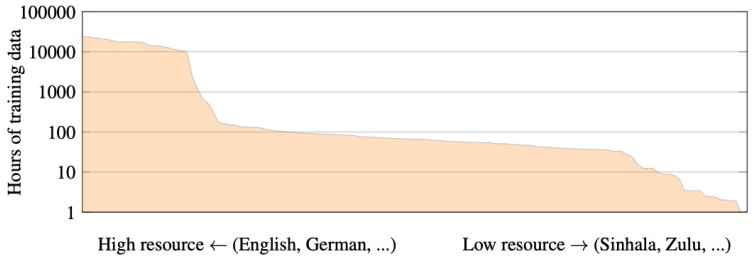


Figure 2: Illustration of the unlabeled training data distribution across the 128 languages of XLS-R.

(XLS-R Blog Post, 2021)

XLS-R: Low resource ASR + classification tasks

- Useful feature encodings for several tasks across languages

Table 11: LibriSpeech ASR results in terms of WER. Models are fine-tuned with 10min, 1h or 10h of annotated data. We compare XLS-R to wav2vec 2.0 (Baevski et al., 2020b) with the same number of parameters (317M). We do not use any language model for these experiments. Cross-lingual training with higher capacity such as for XLS-R (1B) obtains competitive performance.

Model	dev		test	
	clean	other	clean	other
10 min labeled				
wav2vec 2.0 LV-60K (0.3B)	31.7	35.0	32.1	34.5
XLS-R (0.3B)	33.3	39.8	34.1	39.6
XLS-R (1B)	**28.4**	**32.5**	**29.1**	**32.5**
1h labeled				
wav2vec 2.0 LV-60K (0.3B)	13.7	**16.9**	13.7	**17.1**
XLS-R (0.3B)	17.1	23.7	16.8	24.0
XLS-R (1B)	**13.2**	17.0	**13.1**	17.2
10h labeled				
wav2vec 2.0 LV-60K (0.3B)	**5.7**	**9.2**	**5.6**	**9.4**
XLS-R (0.3B)	8.3	15.1	8.3	15.4
XLS-R (1B)	5.9	10.5	5.9	10.6

Table 12: Language identification on VoxLingua107. We report the error rate on the development set spanning 33 languages.

	Error Rate (%)		
	0...5 sec	5...20 sec	Average
Previous work			
Valk & Alumäe (2020)	12.3	6.1	7.1
Ravanelli et al. (2021)	-	-	6.7
This work			
wav2vec 2.0 LV-60K (300M)	11.5	6.3	7.2
XLS-R (0.3B)	**9.1**	**5.0**	**5.7**

Table 13: Speaker identification accuracy on VoxCeleb1 in terms of accuracy. We report baselines from previous work as well as XLS-R (B).

	Accuracy (%)
Previous work	
CNN (Nagrani et al. (2017))	80.5
SUPERB-Hubert Large (Yang et al., 2021)	90.3
This work	
XLS-R (0.3B)	**95.8**

(XLS-R Blog Post, 2021)

XLS-R: multi-lingual ASR

- Fine tuning from self-supervised pre-trained features
- Decoding uses language model. Comparing to SOTA

Table 7: Speech recognition results on BABEL in terms of word error rate (WER) on Assamese (as), Tagalog (tl), Swahili (sw), Lao (lo) and Georgian (ka).

	as	tl	sw	lo	ka
Labeled data	55h	76h	30h	59h	46h
Previous work					
Alumäe et al. (2017)	-	-	-	-	32.2
Ragni et al. (2018)	-	40.6	35.5	-	-
Inaguma et al. (2019)	49.1	46.3	38.3	45.7	-
XLSR-10 (Conneau et al., 2021)	44.9	37.3	35.5	32.2	-
XLSR-53 (Conneau et al., 2021)	44.1	33.2	26.5	-	31.1
This work					
XLS-R (0.3B)	42.9	33.2	24.3	31.7	28.0
XLS-R (1B)	40.4	30.6	21.2	30.1	25.1
XLS-R (2B)	**39.0**	**29.3**	**21.0**	**29.7**	**24.3**

(XLS-R Blog Post, 2021)

Outline

- Foundation models
  - Wav2Vec 2.0
  - HuBERT
  - XLS-R cross-lingual features
  - **Working example**
- SpeechBrain homework overview

Most common project feedback: Try foundation features!

- If your project would benefit from a good audio encoding, try some of what's available via HF
- Potentially good way to work with smaller datasets

Example: Using HF transformers API

https://huggingface.co/docs/transformers/model_doc/hubert

Wav2Vec2FeatureExtractor

class transformers.**Wav2Vec2FeatureExtractor** <source>

```
( feature_size = 1, sampling_rate = 16000, padding_value = 0.0,
return_attention_mask = False, do_normalize = True, **kwargs )
```

Parameters

- **feature_size** (`int`, defaults to 1) — The feature dimension of the extracted features.
- **sampling_rate** (`int`, defaults to 16000) — The sampling rate at which the audio files should be digitalized expressed in Hertz per second (Hz).
- **padding_value** (`float`, defaults to 0.0) — The value that is used to fill the padding values.
- **do_normalize** (`bool`, *optional*, defaults to `True`) — Whether or not to zero-mean unit-variance normalize the input. Normalizing can help to significantly improve the performance for some models, *e.g.*, wav2vec2-lv60.
- **return_attention_mask** (`bool`, *optional*, defaults to `False`) — Whether or not **call()** should return `attention_mask`.

Example: Using HF transformers API

- Create huBERT audio features from pre-trained model
- https://huggingface.co/docs/transformers/model_doc/hubert

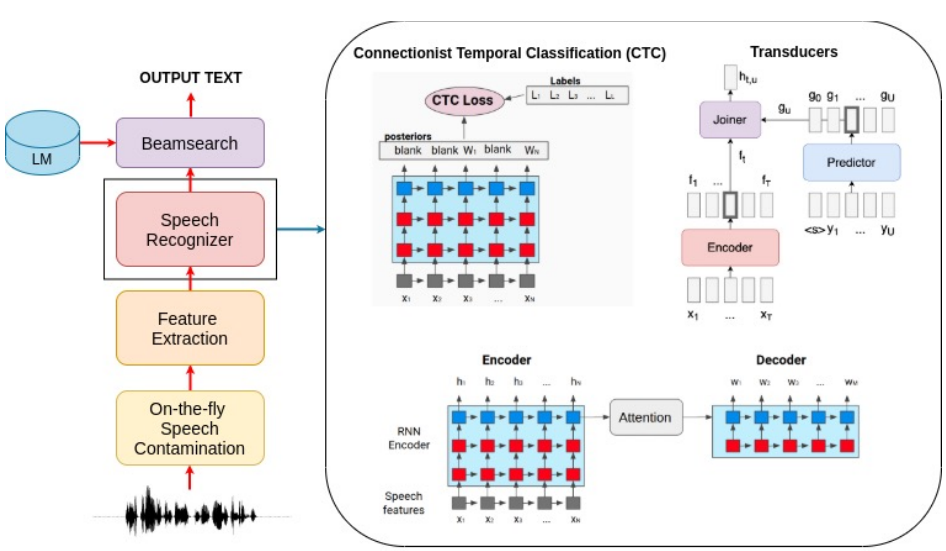
Public notebook with examples

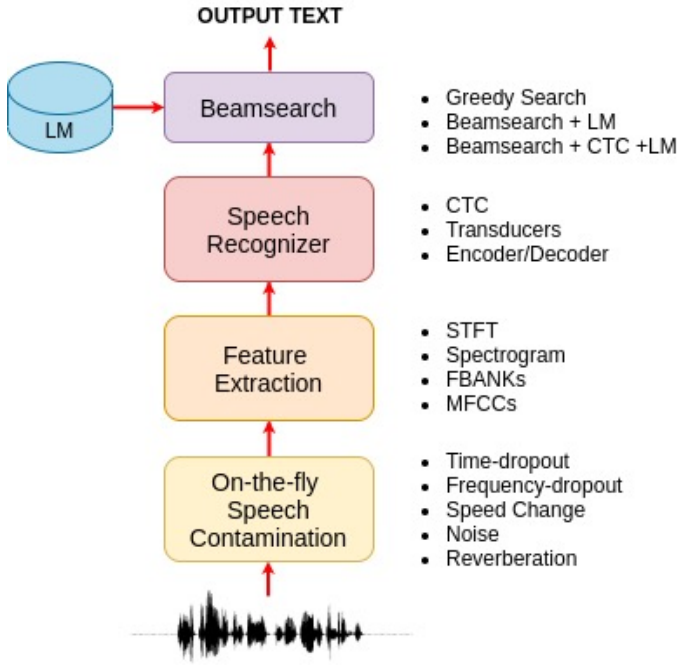
Outline

- Foundation models
  - Wav2Vec 2.0
  - HuBERT
  - XLS-R cross-lingual features
  - Working example
- **SpeechBrain homework overview**

SpeechBrain overview

- Full ASR toolkit. Integrated with PyTorch. Still figuring out APIs / tools interface





Appendix