Unsupervised Speech Recognition

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, Michael Auli

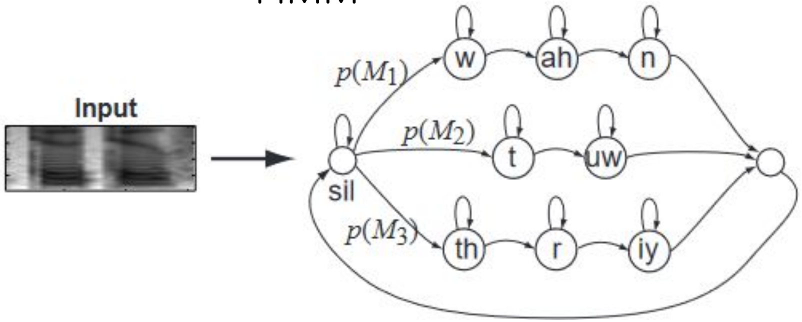
UBC DL-NLP Reading Group
Peter Sullivan 6/30/21

Outline

1. Background
2. Overview of wav2vec-u model
3. Experiments
4. Results
5. Discussion

Background (brief)

HMM



1. Hidden Markov Models (HMM)

   Find 'model' most likely to generate

2. wav2vec 2.0

   Semi-supervised technique (like BERT) to learn good representation of audio

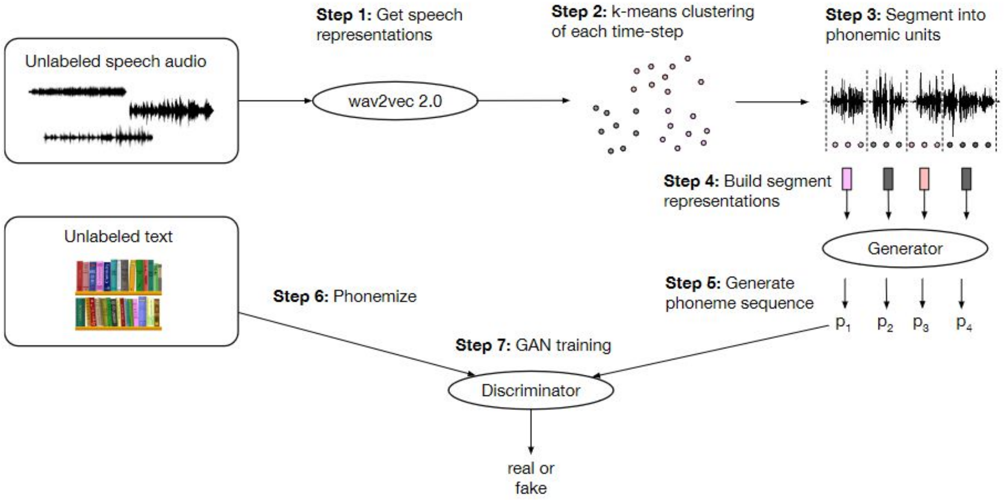
3. Generative Adversarial Network (GAN)

   Technique to generate convincing data by using a discriminator and generator

Image credits:
https://www.ee.columbia.edu/~dpwe/e6820/lectures/L09-asr.pdf

wav2vec-u high level overview

1. Train wav2vec 2.0 on untranscribed audio data
2. Cluster representation to identify phoneme-units per time
3. Use GAN to “create” phoneme transcriptions
    a. Use phoneme segments as input to GANN generator
    b. Use phonemized text data (from some other source) as true label for GANN
    c. Dephonemize to get text labels



wav2vec-u high level overview

Pros

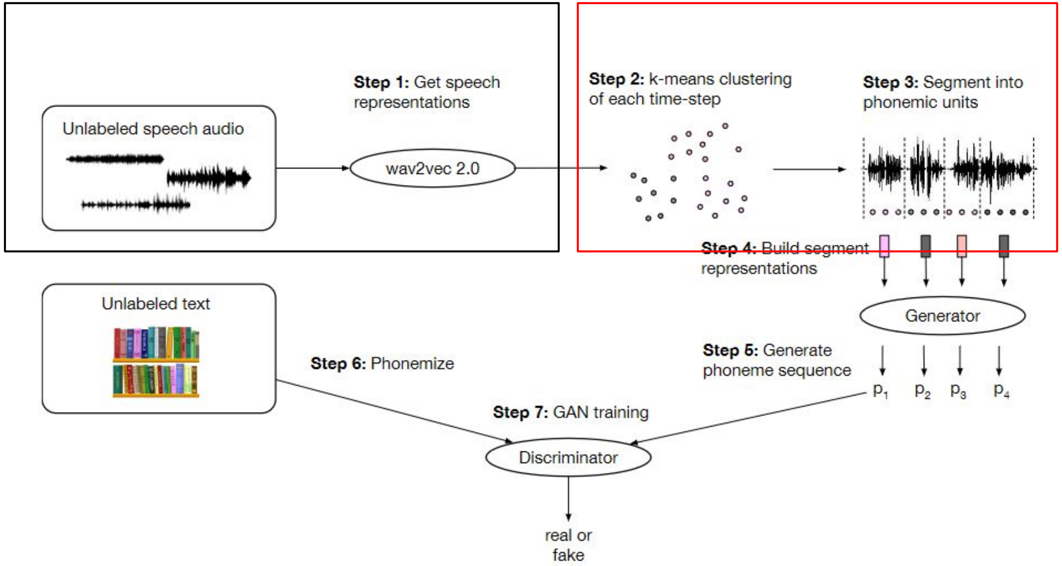
1. Ignoring cost of training wav2vec extremely **lightweight** model (12 hours on V100)
2. Reasonable performance compared with supervised
3. Fairseq implementation
4. Low data requirement!
5. Avoid transcribing!!!

Cons

1. Phonemization may prove problematic for some languages
2. Not great without self-training

wav2vec-u in detail

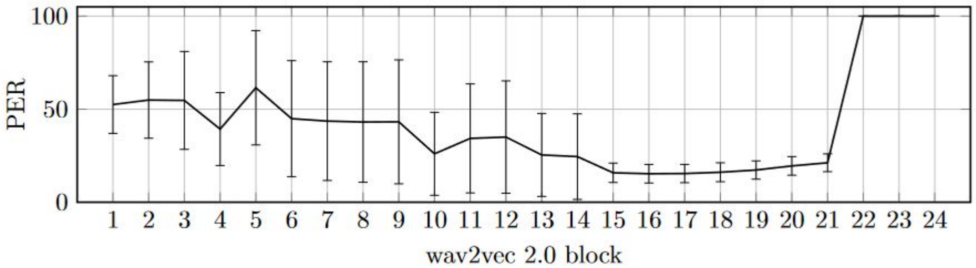
Assume we have a wav2vec 2.0 system

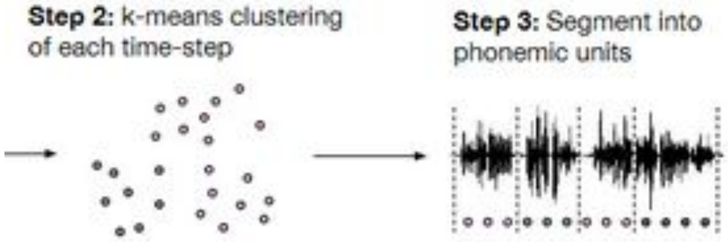


wav2vec-u phone segmentation

1. The output of the 15th layer of the wav2vec context network is chosen as the speech representation (c_1...c_t)
2. Use FAISS k-means (k=128) to cluster all speech representations
3. Set segment where cluster label changes between c_t and c_t+1

Phone Error Rate on Multilingual Librispeech by embedding block used

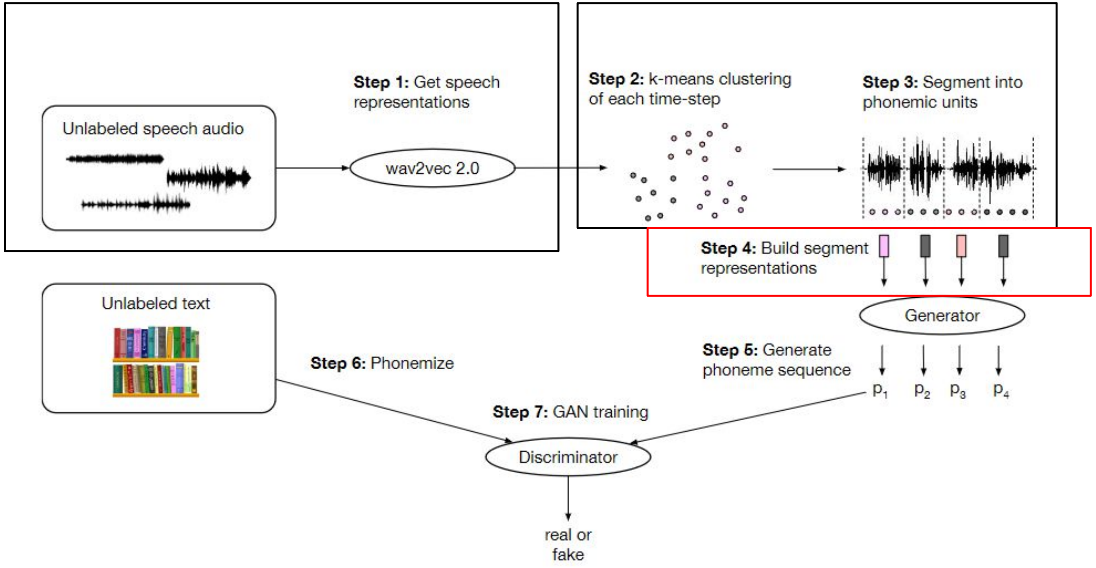




wav2vec-u in detail

Assume we have a wav2vec 2.0 system

and we have segmented all of the training utterances



wav2vec-u segmentation representation

1. Perform PCA (d=512) on speech representations in training
2. Mean pool PCAs of segment
3. Input to GAN generator

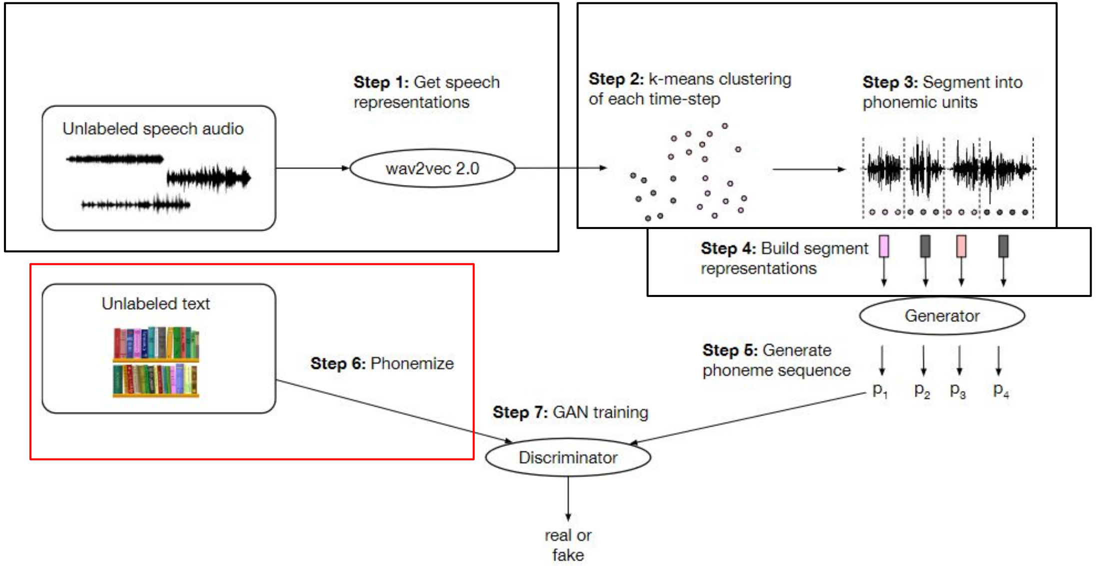
They also experimented with combining segments as well as viterbi decoding for identifying segments (instead of k-means)

Method	Precision	Recall	F1
DAVEnet + peak detection (Harwath and Glass, 2019)	.893	.712	.792
CPC + peak detection (Kreuk et al., 2020)	.839	.836	.837
k-means on wav2vec 2.0 features	.935	.379	.539
wav2vec-U Viterbi prediction	.598	.662	.629

wav2vec-u in detail

Assume we have a wav2vec 2.0 system

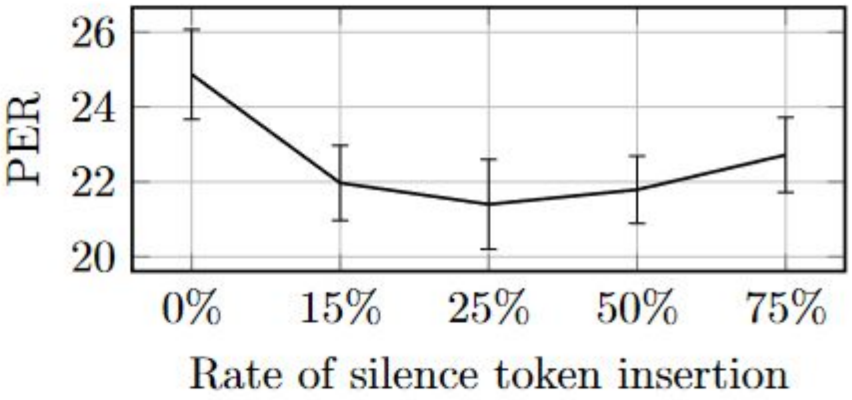
and we have segmented all of the training utterances



wav2vec-u text preparation

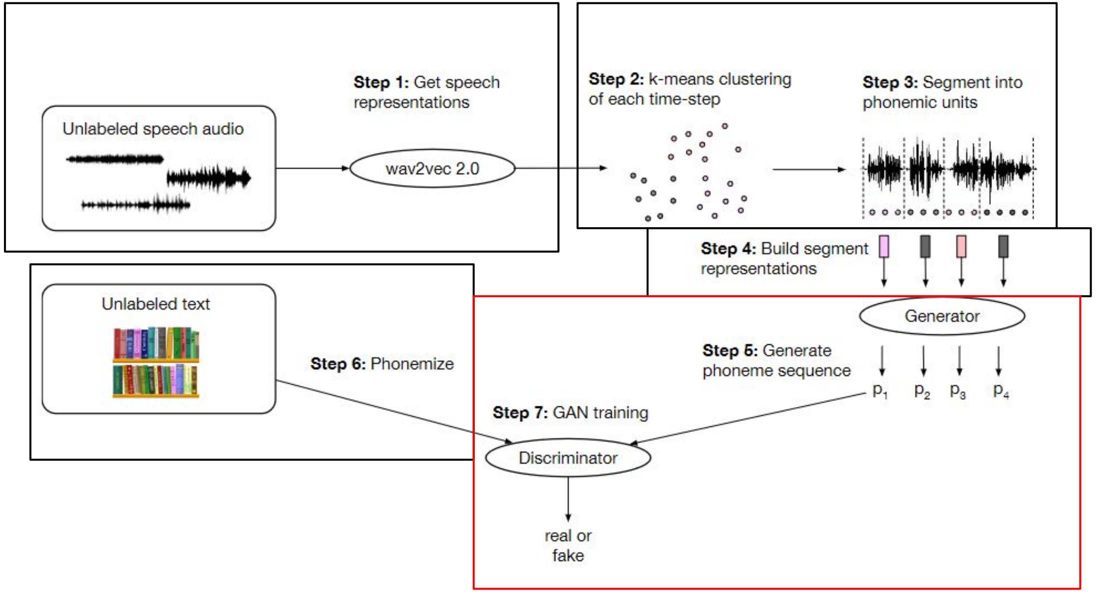
1. Take source text and apply off-the-shelf phonemizer
    a. For EN G2P for others Phonemizer
2. Insert random SIL tokens*
3. Use as “true labels” for GAN

	PER
Baseline	21.4 ± 1.2
- begin/end SIL tokens	25.8 ± 0.7
- audio silence removal	29.3 ± 2.0



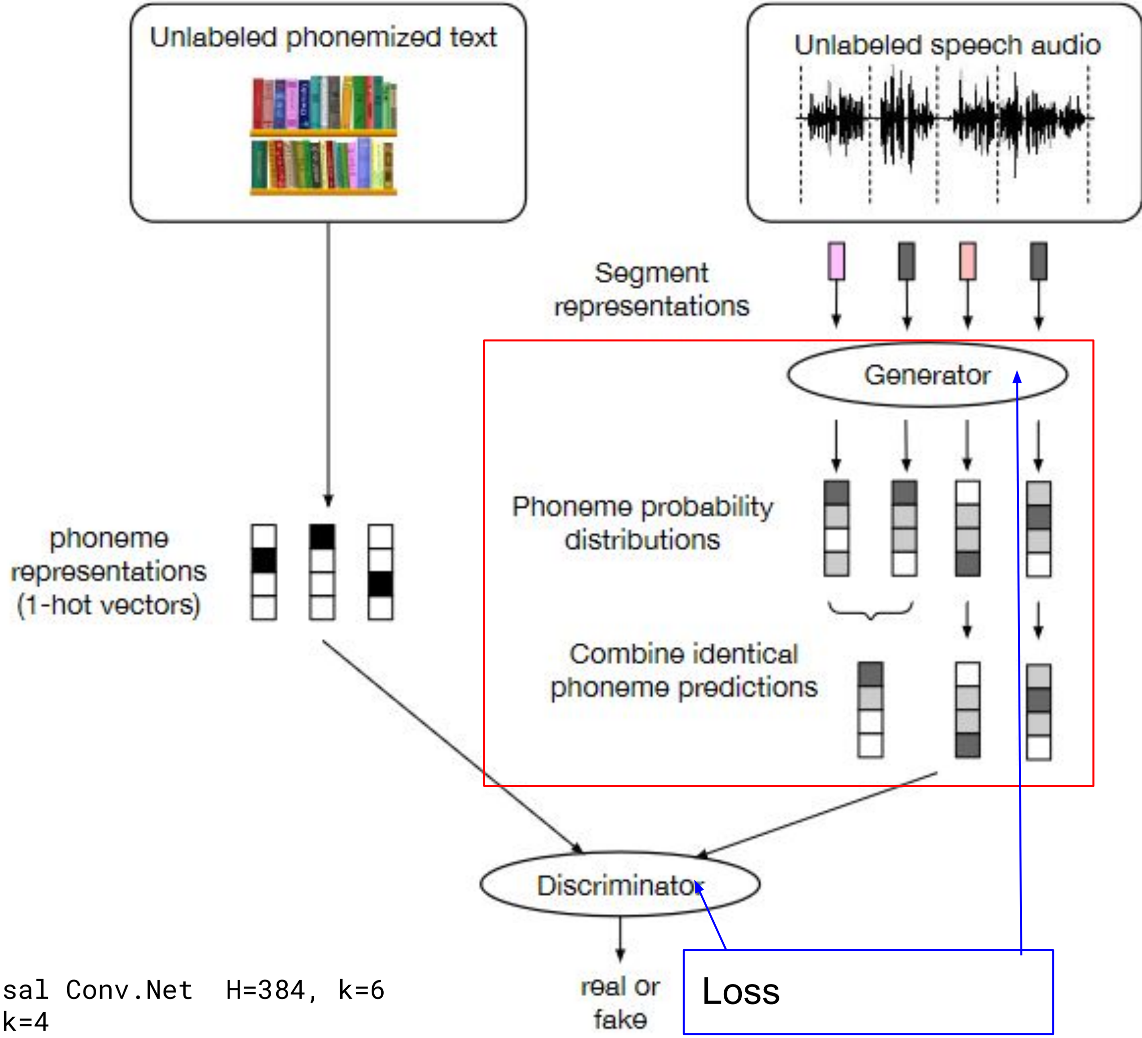
wav2vec-u in detail

Assume we have a wav2vec 2.0 system and we have segmented all of the training utterances



wav2vec-u GAN (1)

1. Generator takes segment reps and predicts phoneme distribution
2. Average nearby predictions with same argmax
3. Backprop see blue arrows (only random segment on Generator)
4. For output WFST decoding with Language Model



DISC: 3-layer Causal Conv.Net H=384, k=6
GEN: 1-layer CNN k=4

wav2vec-u GAN (2) Loss

1. Alternating backprop (DISC and GEN)
2. Gradient Penalty (DISC)
   Helps with stability (soft enforce Lipschitz constraint)
3. Segment Smoothness (GEN)
   Penalize subsequent segments from being far apart
4. Phoneme diversity (GEN)
   Max batch-level entropy of phone distribution

$$\mathcal{L}_{gp} = \mathop{\mathbb{E}}_{\tilde{P} \sim \tilde{\mathcal{P}}} \left[\left(\|\nabla \mathcal{C}(\tilde{P})\| - 1 \right)^2 \right]$$

$$\mathcal{L}_{sp} = \sum_{(p_t, p_{t+1}) \in \mathcal{G}(S)} \|p_t - p_{t+1}\|^2$$

$$\mathcal{L}_{pd} = \frac{1}{|B|} \sum_{S \in B} -H_{\mathcal{G}}(\mathcal{G}(S))$$

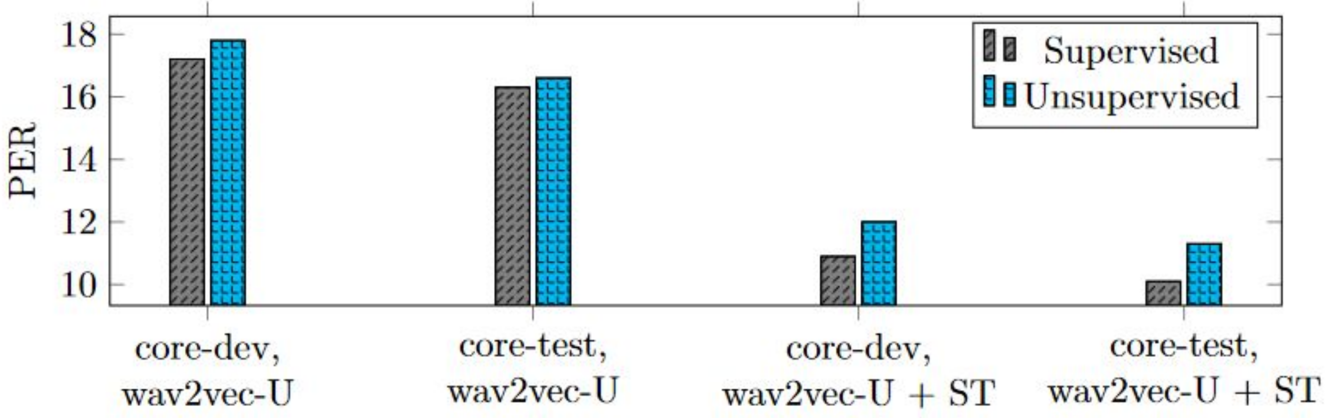
$$\min_{\mathcal{G}} \max_{\mathcal{C}} \mathop{\mathbb{E}}_{P^r \sim \mathcal{P}^r} [\log \mathcal{C}(P^r)] - \mathop{\mathbb{E}}_{S \sim \mathcal{S}} [\log (1 - \mathcal{C}(\mathcal{G}(S)))] - \lambda \mathcal{L}_{gp} + \gamma \mathcal{L}_{sp} + \eta \mathcal{L}_{pd}$$

Experiments

1. Unsupervised Validation
2. Self-training
    a. HMM model trained on pseudo-labels from wav2vec-u
3. Performance on the following datasets
    a. Librispeech (English - Character)
    b. TIMIT (English - Phoneme)
    c. MLS (Dutch, German, French, Spanish, Italian, Portuguese)
    d. Common Voice (Tatar and Kyrgyz)
    e. ALFFA (Swahili)
4. Experiment on amount of Data used

Unsupervised Validation

High level: Use Language Model Entropy and Vocabulary (phoneme) Usage Entropy to act as a proxy for labeled data during hyperparameter optimization. (See paper for details)



Self Training

High level:

Use pseudo-labels from GAN to train HMM, then relable with HMM and fine-tune wav2vec 2.0 model with CTC (HMM + fine-tune)

Model	LM	core-dev	core-test	all-test
wav2vec-U	4-gram	17.0	17.8	16.6
+ HMM	4-gram	13.7	14.6	13.5
+ HMM + HMM	4-gram	13.3	14.1	13.4
+ HMM resegment + GAN	4-gram	13.6	14.4	13.8
+ fine-tune	4-gram	12.0	12.7	12.1
+ fine-tune	-	12.1	12.8	12.0
+ fine-tune + fine-tune	-	12.0	12.7	12.0
+ HMM + fine-tune	-	11.3	11.9	11.3
+ HMM + fine-tune	4-gram	11.3	12.0	11.3

Results - Librispeech

Model	Unlabeled data	LM	dev		test	
			clean	other	clean	other
960h - Supervised learning						
DeepSpeech 2 (Amodei et al., 2016)	-	5-gram	-	-	5.33	13.25
Fully Conv (Zeghidour et al., 2018)	-	ConvLM	3.08	9.94	3.26	10.47
TDNN+Kaldi (Xu et al., 2018)	-	4-gram	2.71	7.37	3.12	7.63
SpecAugment (Park et al., 2019)	-	-	-	-	2.8	6.8
SpecAugment (Park et al., 2019)	-	RNN	-	-	2.5	5.8
ContextNet (Han et al., 2020)	-	LSTM	1.9	3.9	1.9	4.1
Conformer (Gulati et al., 2020)	-	LSTM	2.1	4.3	1.9	3.9
960h - Self and semi-supervised learning						
Transf. + PL (Synnaeve et al., 2020)	LL-60k	CLM+Transf.	2.00	3.65	2.09	4.11
IPL (Xu et al., 2020b)	LL-60k	4-gram+Transf.	1.85	3.26	2.10	4.01
NST (Park et al., 2020)	LL-60k	LSTM	1.6	3.4	1.7	3.4
wav2vec 2.0 (Baevski et al., 2020c)	LL-60k	Transf.	1.6	3.0	1.8	3.3
wav2vec 2.0 + NST (Zhang et al., 2020b)	LL-60k	LSTM	1.3	2.6	1.4	2.6
Unsupervised learning						
wav2vec-U LARGE	LL-60k	4-gram	13.3	15.1	13.8	18.0
wav2vec-U LARGE + ST	LL-60k	4-gram	3.4	6.0	3.8	6.5
	LL-60k	Transf.	3.2	5.5	3.4	5.9

Results - TIMIT

Matched:
Unlabeled text include transcriptions of audio

Unmatched:
Different split no overlap with text and audio

Model	LM	core-dev	core-test	all-test
Supervised learning				
LiGRU (Ravanelli et al., 2018)	-	-	14.9	-
LiGRU (Ravanelli et al., 2019)	-	-	14.2	-
Self and semi-supervised learning				
vq-wav2vec (Baevski et al., 2020b)	-	9.6	11.6	-
wav2vec 2.0 (Baevski et al., 2020c)	-	7.4	8.3	-
Unsupervised learning - matched setup				
EODM (Yeh et al., 2019)	5-gram	-	36.5	-
GAN* (Chen et al., 2019)	9-gram	-	-	48.6
GAN + HMM* (Chen et al., 2019)	9-gram	-	-	26.1
wav2vec-U	4-gram	17.0	17.8	16.6
wav2vec-U + ST	4-gram	11.3	12.0	11.3
Unsupervised learning - unmatched setup				
EODM (Yeh et al., 2019)	5-gram	-	41.6	-
GAN* (Chen et al., 2019)	9-gram	-	-	50.0
GAN + HMM* (Chen et al., 2019)	9-gram	-	-	33.1
wav2vec-U*	4-gram	21.3	22.3	24.4
wav2vec-U + ST*	4-gram	13.8	15.0	18.6

Results - MLS

Model	Labeled data used	LM	de	nl	fr	es	it	pt	Avg
Labeled training hours (full)			2k	1.6k	1.1k	918	247	161	
Supervised learning									
Pratap et al. (2020)	full	5-gram	6.49	12.02	5.58	6.07	10.54	19.49	10.0
Unsupervised learning									
wav2vec-U	0h	4-gram	32.5	40.2	39.8	33.3	58.1	59.8	43.9
wav2vec-U + ST	0h	4-gram	11.8	21.4	14.7	11.3	26.3	26.3	18.6

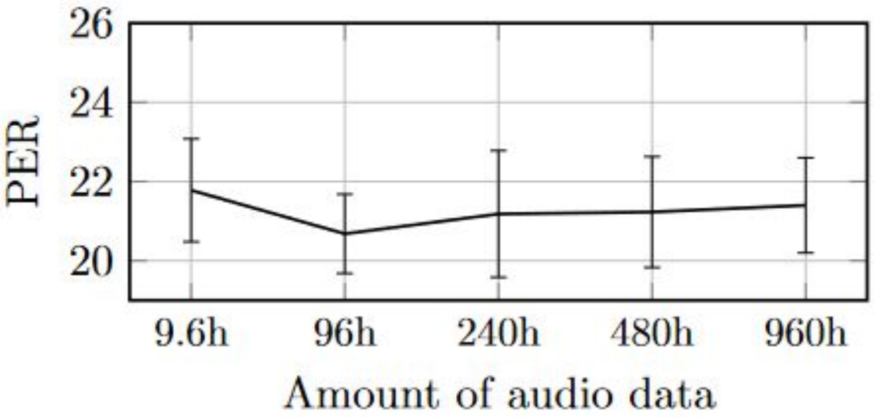
Results - Low Resource

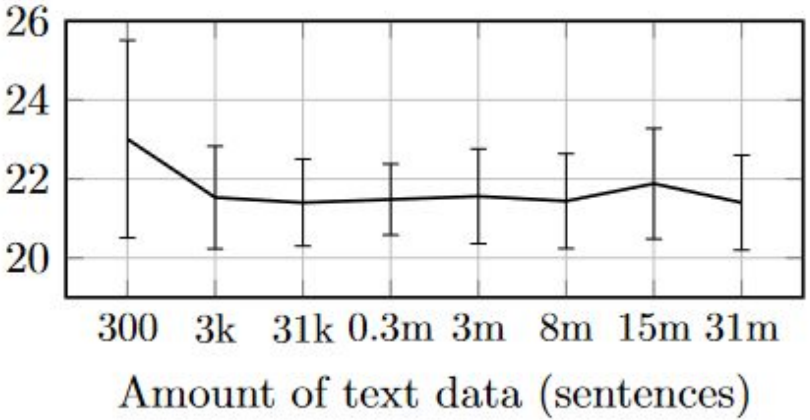
Model	tt	ky
Supervised learning		
Fer et al. (2017)	42.5	38.7
m-CPC (Rivière et al., 2020)	42.0	41.2
XLSR-53 (Conneau et al., 2020)	5.1	6.1
Unsupervised learning		
wav2vec-U	25.7	24.1
wav2vec-U + HMM	13.7	14.9

Model	sw
Supervised learning	
Besacier et al. (2015)	27.36
Unsupervised learning	
wav2vec-U	52.6
wav2vec-U + ST	32.2

Tatar (4.6h)
Kyrgyz (1.8h)
Swahili (9.2h)

Results - Data Quantity





Ablation

Ablation	mean PER ± std	%-converged (PER < 40)
Baseline	21.4 ± 1.2	100%
9.6h audio, 3k text	21.2 ± 1.1	100%
96h audio, 3k text	21.1 ± 1.3	95%
w/o clustering, pca, mean pool	-	0%
w/o clustering	-	0%
w/o 2nd stage mean pool	-	0%
w/o PCA	-	0%
64 clusters	23.1 ± 0.7	100%
256 clusters	22.3 ± 1.1	100%
256 PCA	21.6 ± 1.1	100%
768 PCA	28.0 ± 1.5	90%
use full phone set	23.51 ± 1.3	100%

Discussion

1. Importance of Phonemization (Dialectal Arabic?)
2. Data quantity (good performance with <100hr)
3. Training speed is fast
    a. 12hrs for GAN training
    b. 80k updates for 100hr finetuning (~Librispeech)
    c. 18k updates for 1hr finetuning (~TIMIT)
4. Return of the HMMs ???

Github