Session #19:
Self-Supervised Speech/Audio Models

Tuesday, November 1

CSCI 601.771: Self-supervised Statistical Models

Stakeholder

- Learning robust representations from speech audio alone, followed by fine-tuning on transcribed can outperform the best semi-supervised methods ,Also conceptually simpler.
- wav2vec 2.0 masks the speech input and solves the contrastive learning task
- A framework for self-supervised learning of representations of raw audio data.

Stakeholder: Method

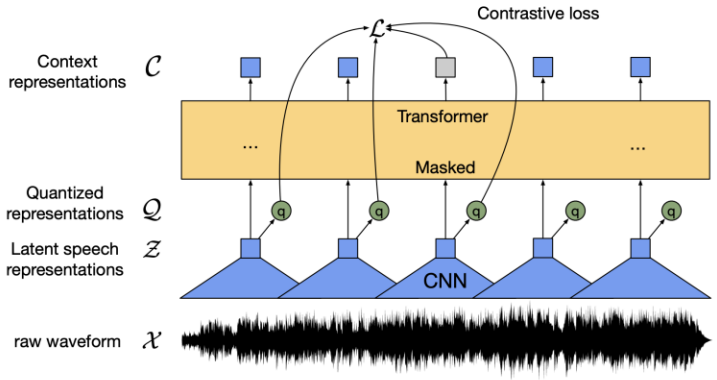
- Encodes speech audio through a multilayer convolutional neural network
- Latent representations are fed into the Transformer network to build context-based representations
- As part of training, we learn discrete speech units via softmax to represent latent representations in contrast tasks.

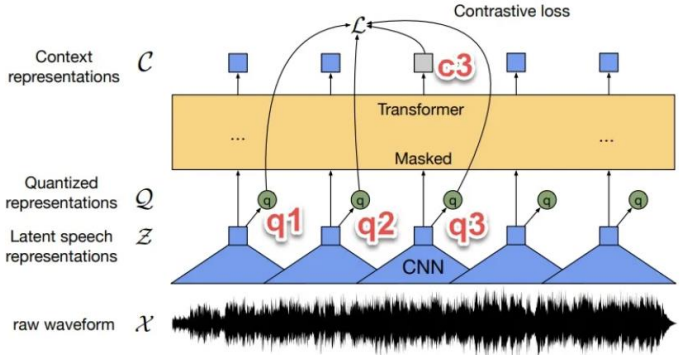
Stakeholder: Method

- The model is fine-tuned on labeled data
- Method end-to-end Both problems are solved in a straightforward manner.
- Other related work includes learning representations by auto-encoding input data or directly predicting future time steps
- Our results show that jointly learning discrete speech units with contextual representations achieves better results than fixed units learned in a prior step .
- The feasibility of ultra-low-resource speech recognition
- We achieve state-of-the-art sota on TIMIT phoneme recognition as well as Librispeech's 100-hour clean subset.

Steven Tan,Neha Verma ,Zhiqing Zhong

Stakeholder: Models

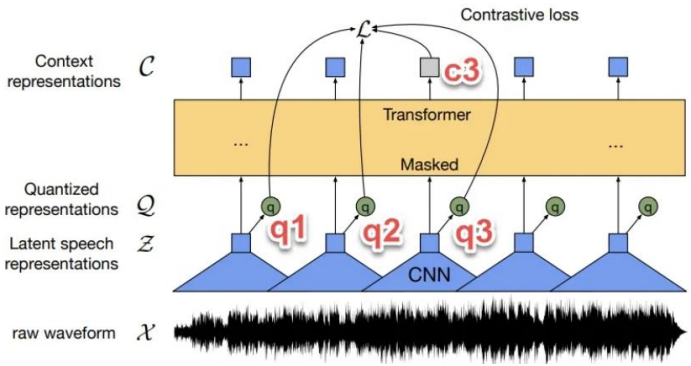




X = {x1, x2, ..., xT'}, this is the original wave form, e.g. sample rate = 16000;
Z = {z1, z2, ..., zT}, this is the "latent speech representation" (hidden layer speech representation) obtained after subsampling with 7-layer CNN; the combination of these 7-layer CNNs is called "" feature extractor".
Q = {q1, q2, ..., qT}, "quantized" from Z. For example, in the default code of fair seq, if there are two codebooks, each codebook has 320 codewords, and each codeword is represented by a 128-dimensional vector. That is to say, a tensor with shape (2, 320, 128) (similar to codeword embedding matrix) is what we need to learn.
C = {c1, c2, ..., cT}, is the "context representations" obtained by inputting Z through multiple layers of transformer encoders.

Steven Tan,Neha Verma ,Zhiqing Zhong

Stakeholder:Models



Algorithm:
From X to Z, the original speech is represented as, the hidden layer speech representation;
From Z to Q, quantization operation;
A part of Z is given to the mask, for example, five 10-gram positions, these positions are replaced with a uniform vector; then Z_mask after the mask is thrown to the transformer encoder;
The result C predicted by the transformer is compared with the reference answer (= quantization of the (unmasked) Z sequence to get the Q sequence), and the "contrastive learning loss" is calculated, as well as several other Loss.

Steven Tan,Neha Verma ,Zhiqing Zhong

Stakeholder:Models(which has been masked?)

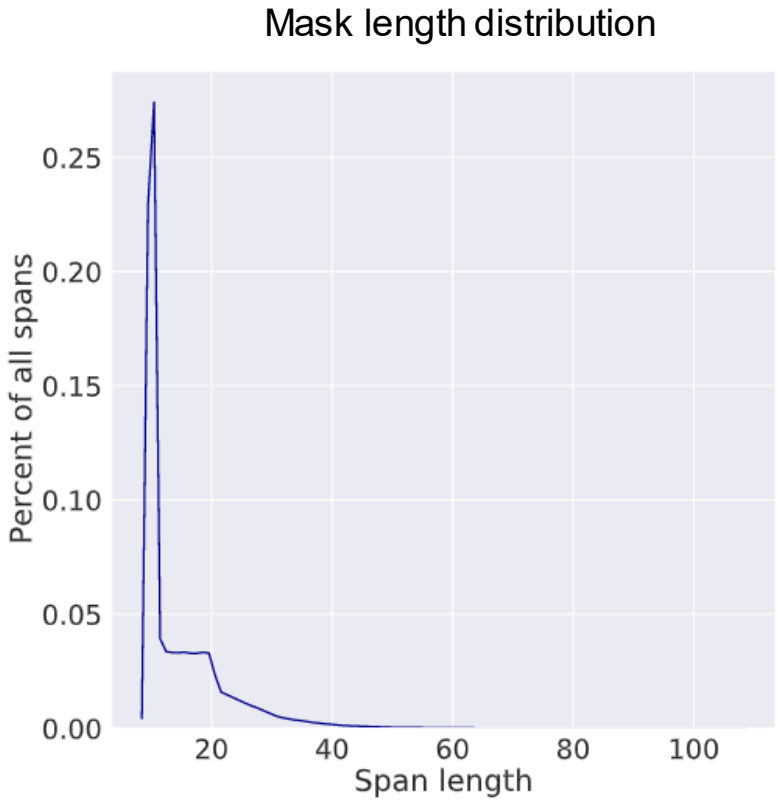
- In the above figure, the z3 corresponding to q3 in q1, q2, q3, q4, q5, is masked
- In order to predict q3, two interference terms are randomly found: q1, q2.
- After pretraining on unlabeled speech, the model is fine-tuned on labeled data with a connectionist temporal classification (CTC) loss for downstream speech recognition tasks .

Training

Mask:

P: portion of time steps to start masking (p=0.065)

M: consecutive time steps (M=10)



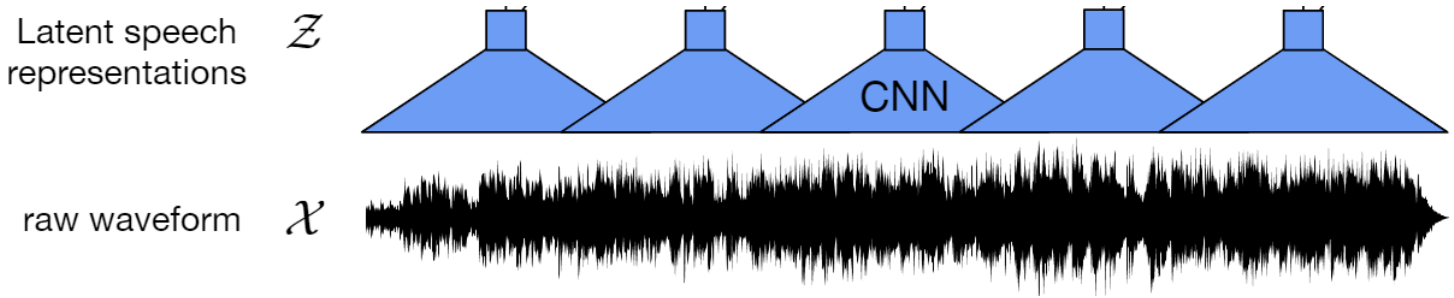
: Steven Tan, Neha Verma, Zhiqing Zhong

Different Masking Strategies

	avg WER	std
Baseline ($p = 0.075$)	7.97	0.02
Mask length $M = 8$	8.33	0.05
Mask length $M = 12$	8.19	0.08
Mask length $M = 15$	8.43	0.19
Mask probability $p = 0.065$	7.95	0.08
Mask probability $p = 0.06$	8.14	0.22
Mask w/o overlap, uniform(1,31)	8.39	0.02
Mask w/o overlap, uniform(10,30)	9.17	0.05
Mask w/o overlap, poisson(15)	8.13	0.04
Mask w/o overlap, normal(15, 10)	8.37	0.03
Mask w/o overlap, length 10	9.15	0.02
Mask w/o overlap, length 15	9.43	0.26

CNN for Feature Extraction

- 7-layer CNN block
- Receptive field 400 input sample/25 ms of audio



Training Objective

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

Lm: contrastive loss

C: contextual representation from transformer

Q: quantized candidate of input

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

K = 100 (# distractors),
temperature=0.1

Training Objective

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

Ld: diversity loss

V: entries

G: #codebooks

$$p_{g,v} = \frac{\exp(l_{g,v} + n_v)/\tau}{\sum_{k=1}^{V} \exp(l_{g,k} + n_k)/\tau},$$

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

Training Objective

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

Ld: diversity loss

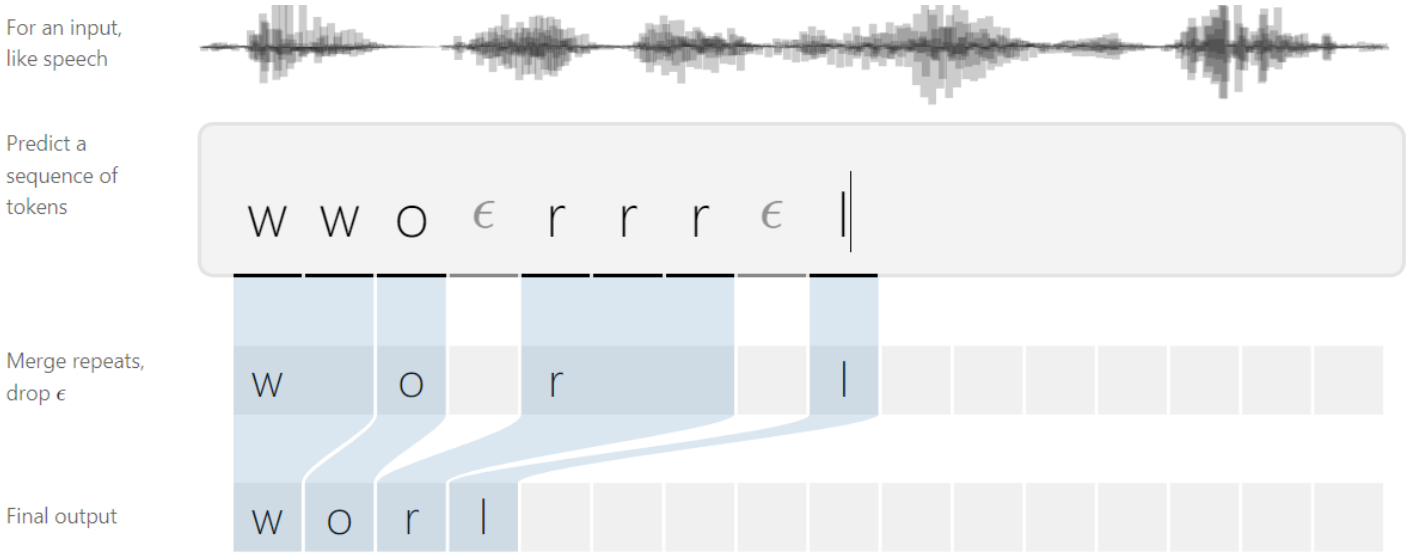
V=320

G=2

Entry dimension: input dim / G = 768 /2 =384

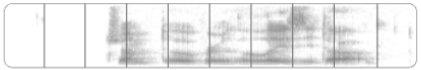
Finetuning

- CTC loss

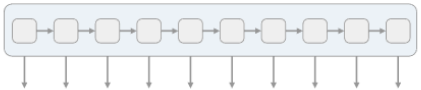


✍: Steven Tan, Neha Verma, Zhiqing Zhong

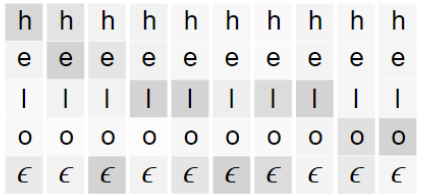
CTC loss



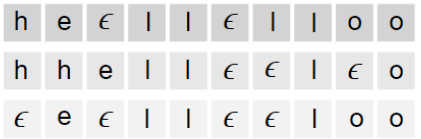
We start with an input sequence, like a spectrogram of audio.



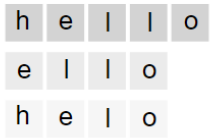
The input is fed into an RNN, for example.



The network gives $p_t(a \mid X)$, a distribution over the outputs {h, e, l, o, ϵ} for each input step.



With the per time-step output distribution, we compute the probability of different sequences



By marginalizing over alignments, we get a distribution over outputs.

Our favorite: intractable marginalization!

$$p(Y \mid X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X)$$

The CTC conditional **probability**

marginalizes over the set of valid alignments

computing the **probability** for a single alignment step-by-step.

Inference: beam-search

$$Y^* = \underset{Y}{\operatorname{argmax}} \; p(Y \mid X) \cdot p(Y)^{\alpha} \cdot L(Y)^{\beta}$$

The CTC conditional probability.

The language model probability.

The "word" insertion bonus.

✍: Steven Tan, Neha Verma, Zhiqing Zhong

Datasets and Evaluation

1. Pretraining
    1. Librispeech (no labels): 960 hrs
    2. LibriVox: 53.2K hrs
2. Language Model (for Speech Recognition)
    1. Librispeech LM corpus

2. Fine-tuning
    1. 960 hrs of Librispeech (labels)
    2. Librispeech subsets
        1. Train-clean-100 hrs
        2. Libri-light 10 hrs
        3. Libri-light 1 hr
        4. Libri-light 10 min
    3. TIMIT Acoustic-Phonetic Speech Corpus (5 hrs)

Model varieties

1. Size
    1. Base
        1. 12 Transformer layers
    2. Large
        1. 24 Transformer layers

1. LM used
    1. 4-gram
    2. Transformer-based
        1. 20-layer

Results: Low-Resource Labeled Data

Model	Unlabeled data	LM	dev clean	dev other	test clean	test 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
100h labeled						
Hybrid DNN/HMM [34]	-	4-gram	5.0	19.5	5.8	18.6
TTS data augm. [30]	-	LSTM			4.3	13.5
Discrete BERT [4]	LS-960	4-gram	4.0	10.9	4.5	12.1
Iter. pseudo-labeling [58]	LS-860	4-gram+Transf.	4.98	7.97	5.59	8.95
	LV-60k	4-gram+Transf.	3.19	6.14	3.72	7.11
Noisy student [42]	LS-860	LSTM	3.9	8.8	4.2	8.6
BASE	LS-960	4-gram	2.7	7.9	3.4	8.0
		Transf.	2.2	6.3	2.6	6.3
LARGE	LS-960	Transf.	2.1	4.8	2.3	5.0
	LV-60k	Transf.	1.9	4.0	2.0	4.0

- Very good WER for ultra-low resource 10 min recording
- New state-of-the-art on Librispeech train-clean-100

✍: Steven Tan, Neha Verma, Zhiqing Zhong

Results: High-Resource Labeled Data (Librispeech)

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

- New state-of-the-art
- Other architectures from scratch work better
  - but self-supervision technique is very helpful

Results: TIMIT Phoneme Recognition

- Task: transcribe speech using 39 phones

Table 3: TIMIT phoneme recognition accuracy in terms of phoneme error rate (PER).

	dev PER	test PER
CNN + TD-filterbanks [59]	15.6	18.0
PASE+ [47]	-	17.2
Li-GRU + fMLLR [46]	–	14.9
wav2vec [49]	12.9	14.7
vq-wav2vec [5]	9.6	11.6
This work (no LM)		
LARGE (LS-960)	7.4	8.3

Results: Ablations

- Quantizing happens only for latents as targets, not for latents as inputs
- Continuous input – retain more info
- Quantized output – more robust training

Table 4: Average WER and standard deviation on combined dev-clean/other of Librispeech for three training seeds. We ablate quantizing the context network input and the targets in the contrastive loss.

	avg. WER	std.
Continuous inputs, quantized targets (Baseline)	7.97	0.02
Quantized inputs, quantized targets	12.18	0.41
Quantized inputs, continuous targets	11.18	0.16
Continuous inputs, continuous targets	8.58	0.08

Empiricists

https://colab.research.google.com/drive/1gR5rMqy3H5D74qVGlZCE2iSJT4E5VNyp?usp=sharing

Reviewer: Strengths

- Extensively open-sourced codes and models.
- Great results on low-resource labelled data, huge potential impact for uncommon languages.
- A lot of moving parts work together well.

- Quantization module: a great way to downscale feature representation

- Motivation: consider real world
- Experiments based on public dataset

Reviewer: Weaknesses

- Paper is too dense: hard to follow all the techniques.
- No evaluation for the pretrained model.
- No reasoning for using relative positional embedding vs absolute in contextualized representations.
- Tons of other hyperparameter choices mentioned with explanations or ablation studies either missing or left in appendix.

- Not end-to-end, needs secondary recognition model.
- Relies heavily on the pre-training model, which is very large and hard to train.

Reviewer: Weakness

- Limited argumentation
  - Why such design decision has made
  - Quantization: yes, discretize to smaller space, and then …
- Less clarity
  - Denotes not clear: G, V
- Not easy to reproduce to evaluation
  - Large computational resources

- Aayush, Tianqi, Yongrui

Visionary: Few-shot speech recognition for close languages

- Paper shows impressive performance on low-resource languages and phenome recognition.
  - However, it doesn't show the verbal equivalent of zero or few-shot performance.
  - Additionally, it doesn't show how proficiency in one language may generalize to proficiency in similar languages.
- Create a study of the few-shot performance of wav2vec and see how finetuning on one language generalizes to similar languages.
  - Ex: Fine-tune on Spanish and see if it has generalization properties in Italian.
  - Ex: Japanese has 2 written forms expressing different aspects of the language and it doesn't necessarily correlate 1-to-1 with spoken Japanese.
- Motivation: For languages with no written form, you may be able to use a similar language that has written language and then generalize.

Visionary: Improvement from Error Analysis

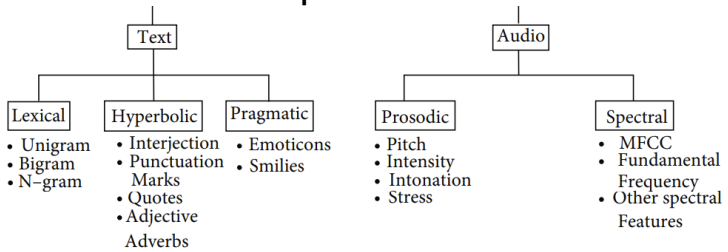
Time of Trained Labeled Data	Error Types
10 minutes	- Phonetic spelling errors in general (omit silent and repeated characters) *i.e. are -> ar*
1 hour	- Phonetic spelling errors for less common words *i.e. soul -> sol*
10 hours	- Articles mistaken *i.e. a -> the, in -> and* - Alternative spellings *i.e color vs colour*
100 hours	- Phonetic spelling errors mostly for person names *i.e. christie -> cristy* - Incorrect spacing *i.e anyone vs any one*
960 hours	- Similar to 100 hours but error rate to 2% + rare words

- Emphasis on silent/repeated-letter words
- Why articles are mistaken

Visionary: Downstream Emotion and Sarcasm Detection

- Strength in self-supervised learning with unlabeled data: Limited emotion and sarcasm detection datasets with annotation

- Additional information in audio not present in text



- Transfer learning: Use the trained weights of contextual representations as starting points for emotion and sarcasm detection models