Self-supervised learning for speech processing

Facebook AI Research

Alexei Baevski, Alexis Conneau, Steffen Schneider, Henry Zhou, Abdelrahman Mohamed, Anuroop Sriram, Naman Goyal, Wei-Ning Hsu, Michael Auli, Kritika Singh, Yatharth Saraf, Geoffrey Zweig, Qiantong Xu, Tatiana Likhomanenko, Paden Tomasello, Ronan Collobert, Gabriel Synnaeve
Speech technology

Video captioning

Mobile devices

Home devices
Speech applications

- Speech to text/speech recognition - dictation etc.
- Text to speech - reading out aloud
- Keyword spotting - “Hey Alexa/Portal”
- Speaker identification - is it your voice?
- Language identification
- Speech translation
Overview

- Traditional speech recognition
- Self-supervised learning for speech processing
  - wav2vec 2.0
  - Cross-lingual training
  - Completely unsupervised speech recognition
Traditional speech recognition
I like black tea with milk
Traditional automatic speech recognition (ASR)

\[ W^* = \arg \max_W p(W|X) \]
\[ W^* = \arg \max_W p(F|W)p(W) \]
Traditional automatic speech recognition (ASR)

- Represent words as sequences of phonemes
- hello = h eh l ow
- Distinct units of sound to distinguish words
Traditional automatic speech recognition (ASR)

\[ W^* = \arg \max_W p(W|X) \]

\[ W^* = \arg \max_W \sum_Q p(F|Q)p(Q|W)p(W) \]
Traditional automatic speech recognition (ASR)

$W^* = \arg \max_W p(W|X)$

$W^* = \arg \max_W \sum_Q p(F|Q) p(Q|W) p(W)$

*Focus of this talk*
Feature representation

- Typical sample rates for speech: 8KHz, 16KHz.
- Traditionally: build spectrogram
**Spectrogram**

- Small window, e.g., 20ms of waveform
- Compute FFT and take magnitude
- Describes frequency content in local window
Spectrogram

- Concatenate frames from adjacent windows to form a spectrogram
Self-supervised speech representation learning
Training speech recognition models

I like black tea with milk

- Train on 1,000s of hours of transcribed data for good systems.
- Many languages, dialects, domains etc.
Supervised machine learning

(, cat)

potential train/test mismatch

Need to annotate lots of data!
Supervised machine learning

Need to annotate lots of data!

Not how humans learn!

potential train/test mismatch
Supervised machine learning
Learning good representations of audio data from unlabeled audio
I like tea

Speech recognition

Pre-trained model

0.1
0.5
...
-0.9
Ich mag Tee

Speech translation

Pre-trained model

0.1
0.5
...
-0.9

20
Audio event detection

Pre-trained model

"music"

0.1
0.5
...
-0.9
wav2vec 2.0

- Masked prediction with transformer, bi-directional contextualized representations (similar to BERT).

- But predict what? Learn an inventory of speech units with vector quantization via Gumbel softmax.

- Learning task: Joint VQ & context representation learning.

- Contrast true quantized latent with distractor latents.
wav2vec 2.0

- Masked prediction with transformer, bi-directional contextualized representations (similar to BERT).

- But predict what? Learn an inventory of speech units with vector quantization via Gumbel softmax.

- Learning task: Joint VQ & context representation learning.

- Contrast true quantized latent with distractor latents.
wav2vec 2.0

- Masked prediction with transformer, bi-directional contextualized representations (similar to BERT).

- But predict what? Learn an inventory of speech units with vector quantization via Gumbel softmax.

- Learning task: Joint VQ & context representation learning.

- Contrast true quantized latent with distractor latents.
Objective

Codebook diversity penalty to encourage more codes to be used
Masking

- Sample starting points for masks without replacement, then expand to 10 time-steps (1 time-step is 25ms but 10ms stride)

- Spans can overlap

- For a 15s sample, ~49% of the time-steps masked with an average span length of ~300ms
Fine-tuning

- Add a single linear projection on top into target vocab and train with CTC loss with a low learning rate (CNN encoder is not trained).
- Use modified SpecAugment in latent space to prevent early overfitting.
- Uses wav2letter decoder with the official 4gram LM and Transformer LM.
Results

High resource
(Librispeech 960h labeled)

<table>
<thead>
<tr>
<th>test other</th>
<th>ContextNet (supervised)</th>
<th>Noisy Student (60k-h unlabeled)</th>
<th>wav2vec (60k-h unlabeled)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>3.4</td>
<td>3.3</td>
<td></td>
</tr>
</tbody>
</table>
Results

High resource setup (Librispeech 960h labeled)

Word error rate

- ContextNet (supervised): 4.1
- Noisy Student (60k-h unlabeled): 3.4
- wav2vec (60k-h unlabeled): 3.3

Low resource setup (Librispeech 10min - 100h labeled)

Word error rate

- Noisy Student 100h: 8.6
- wav2vec 100h: 5
- wav2vec 1h: 7.6
- wav2vec 10m: 10.8
- wav2vec 10m + (60k-h unlabeled): 8.2

Note: 10 min labeled data
Effective both for high and low-resource settings!
Results

Effects of model size and amount of unlabeled data

Word error rate on test-other

Labeled data

- Base (100m)
- Large (300m)
- + 60k-h

<table>
<thead>
<tr>
<th>Labeled data</th>
<th>10m</th>
<th>1h</th>
<th>10h</th>
<th>100h</th>
<th>960h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base (100m)</td>
<td>12.9</td>
<td>9.3</td>
<td>7.8</td>
<td>6.3</td>
<td>4.8</td>
</tr>
<tr>
<td>Large (300m)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results

Effects of model size and amount of unlabeled data

Word error rate on test-other

Labeled data

- Base (100m)
- Large (300m)
- + 60k-h

10m: 12.9, 10.8
1h: 9.3, 7.6
10h: 7.8, 6.1
100h: 6.3, 5
960h: 4.8, 4.1
Results

Effects of model size and amount of unlabeled data

Word error rate on test-other

Labeled data

<table>
<thead>
<tr>
<th>Labeled data</th>
<th>Base (100m)</th>
<th>Large (300m)</th>
<th>+ 60k-h</th>
</tr>
</thead>
<tbody>
<tr>
<td>10m</td>
<td>12.9</td>
<td>10.8</td>
<td>8.2</td>
</tr>
<tr>
<td>1h</td>
<td>9.3</td>
<td>7.6</td>
<td>5.8</td>
</tr>
<tr>
<td>10h</td>
<td>7.8</td>
<td>6.1</td>
<td>4.9</td>
</tr>
<tr>
<td>100h</td>
<td>6.3</td>
<td>5</td>
<td>4.0</td>
</tr>
<tr>
<td>960h</td>
<td>4.8</td>
<td>4.1</td>
<td>3.3</td>
</tr>
</tbody>
</table>
Examples (10 min labeled data)

HYP (no LM): she SESED and LUCHMAN GAIVE A SENT won by her GENTAL argument
HYP (w/ LM): she ceased and LUCAN gave assent won by her gentle argument
REF: she ceased and lakshman gave assent won by her gentle argument

HYP (no LM): but NOT WITH STANDING this boris EMBRAED him in a QUIAT FRENDDLY way and CISED him THRE times
HYP (w/ LM): but NOT WITHSTANDING this boris embraced him in a quiet friendly way and kissed him three times
REF: but notwithstanding this boris embraced him in a quiet friendly way and kissed him three times
wav2vec on Hugging Face

- Hugging Face is a popular NLP model zoo
- Hugging Face community fine-tuned our models to do speech recognition in 73 languages.
Pre-training and self-training

- Self-training very successful in speech recognition: generate pseudo-labels
Pre-training and self-training

- Self-training very successful in speech recognition: generate pseudo-labels
Pre-training and self-training

- Self-training very successful in speech recognition: generate pseudo-labels
Pre-training and self-training

- Self-training very successful in speech recognition: generate pseudo-labels

Supervised model

- I like tea
- Hello!

Semi-Supervised model

- What time is it?
The diagram shows the Word Error Rate (WER) on the test-other set for various speech recognition systems and techniques on the Librispeech benchmark with 960 hours of labeled data. The data is based on Papers with Code as of 25 October 2020.

- **Deep Speech 2** (Baidu '15) achieved a WER of 13.25.
- **Fully Conv ASR** (FB '18) had a WER of 10.47.
- **tdnn / Kaldi** (18) resulted in a WER of 7.63.
- **SpecAugment** (Google '19) showed a WER of 5.8.
- **RWTH Hybrid** ('19) had a WER of 5.
- **Pseudo-labeling** (FB '20) achieved a WER of 4.
- **Conformer** (Google '20) resulted in a WER of 3.9.
- **Noisy Student** (Google '20) had a WER of 3.4.
- **wav2vec 2.0** (FB, 2020) showed a WER of 3.3.
- **wav2vec 2.0 + Conf. + NST** (Google, 2020) had a WER of 2.6.
- **wav2vec 2.0** (FB, 2020) resulted in a WER of 8.6.
- **wav2vec 2.0 + SelfTrain** (FB, 2020) had a WER of 5.2.

The diagram illustrates the significant progress in reducing WER over time, with newer techniques like wav2vec 2.0 and its variants performing notably better than earlier systems like Deep Speech 2 and SpecAugment.
Results based on wav2vec 2.0
XLSR: cross lingual speech representation learning with wav2vec
Why cross-lingual self-supervised learning

- Little labeled data -> little unlabeled data
- Leverage unlabeled data from high-resource languages
- To improve performance on low-resource languages
- One model for each of the 6500 languages, for each domain? No.
- Instead: one pertained model for all languages
XLSR: cross lingual speech representation learning with wav2vec
XLSR: Results - cross-lingual transfer

Cross-lingual transfer = Train data from high-resource languages benefits low-resource languages.

CommonVoice results:

<table>
<thead>
<tr>
<th>Model</th>
<th>Phoneme Error Rate (PER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLSR-Mono</td>
<td>High-resource languages</td>
</tr>
<tr>
<td>XLSR-10 (Base)</td>
<td>Low-resource languages</td>
</tr>
<tr>
<td>XLSR-10 (Large)</td>
<td>Low-resource languages</td>
</tr>
</tbody>
</table>
XLSR: Results - multilingual fine-tuning

Multilingual finetuning leads to *one model for all languages* with little loss in performance
XLSR: Results - multilingual fine-tuning

Multilingual finetuning leads to one model for all languages with little loss in performance.

CommonVoice results:

- XLSR-Mono
- XLSR-10
- Multilingual finetuning

Phoneme Error Rate (PER)
XLSR: Analysis of discrete latent speech representations

PCA visualization of latent discrete representations from the multilingual codebook

Similar languages tend to share discrete tokens and thus cluster together
Unsupervised Speech Recognition
Unsupervised speech recognition

- Entirely remove need for labeled data
- Unsupervised machine translation works*, what about speech?
- Key problem: what are the units in the speech audio?
wav2vec Unsupervised: Key ideas

- Learn good representations of speech audio
- Unsupervised segmentation of the speech audio into phonemic units
- Learn mapping between speech segments and phonemes using adversarial learning
wav2vec Unsupervised

Unlabeled speech audio

Unlabeled text
wav2vec Unsupervised

Step 1: Learn speech representations

Unlabeled speech audio

wav2vec 2.0

Unlabeled text
wav2vec Unsupervised

Step 1: Learn speech representations

wav2vec 2.0

Step 2: k-means cluster representations

Unlabeled speech audio

Unlabeled text
wav2vec Unsupervised

Step 1: Learn speech representations

wav2vec 2.0

Step 2: k-means cluster representations

Step 3: Segment into phonemic units

Unlabeled speech audio

Unlabeled text
wav2vec Unsupervised

Step 1: Learn speech representations

Step 2: k-means cluster representations

Step 3: Segment into phonemic units

Step 4: Build segment representations
wav2vec Unsupervised

- Step 1: Learn speech representations
- Step 2: k-means cluster representations
- Step 3: Segment into phonemic units
- Step 4: Build segment representations
- Step 5: Generate phoneme sequence

Unlabeled speech audio

Unlabeled text
wav2vec Unsupervised

Step 1: Learn speech representations

Step 2: k-means cluster representations

Step 3: Segment into phonemic units

Step 4: Build segment representations

Step 5: Generate phoneme sequence

Step 6: Phonemize

Step 7: GAN training

Discriminator

real or fake

Generator

p_1, p_2, p_3, p_4

Unlabeled speech audio

Unlabeled text
Simple segmentation

k-means:
GAN:
gold:

---

...
Text data pre-processing

he spoke soothingly
Text data pre-processing

Unlabeled text

he spoke soothingly

Phonemize

hh iy s ow k s uw dh ih ng l iy
Text data pre-processing

Unlabeled text

he spoke soothingly

Phonemize

sil hh iy s ow k s uw dh ih ng l iy sil

Silence insertion
GAN inputs

Unlabeled phonemized text

phoneme representations (1-hot vectors)

Unlabeled speech audio

Segment representations

Generator

Phoneme probability distributions

Combine identical phoneme predictions

Discriminator

real or fake
Generator / Discriminator

- Generator: 1 layer CNN with 90k parameters
  w2v features frozen
- Discriminator: 3 layer CNN
- Train time: 12-15h on a single V100
Training details

- Unsupervised metric for early stopping, hyper-parameter selection
- Self-training after GAN training (HMM and fine-tuning w2v)
Comparison to prior unsupervised work

Phoneme error rate

TIMIT Benchmark
Comparison to best supervised systems

Amount of labeled data used

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>960 hrs.+</td>
<td>13.25</td>
<td>10.47</td>
<td>7.53</td>
<td>5.8</td>
<td>5</td>
<td>4</td>
<td>3.4</td>
<td>3.3</td>
<td>2.6</td>
<td>5.9</td>
</tr>
<tr>
<td>1 hr.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 hrs.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Librispeech benchmark
Other languages

MLS benchmark, wav2vec-U used only 100h of unlabeled data but there is up to 2k hours for some languages.
Low-resource languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Supervised (Besacier et al. '15)</th>
<th>wav2vec-U</th>
<th>wav2vec-U + ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kyrgyz</td>
<td>24.75</td>
<td></td>
<td>16.5</td>
</tr>
<tr>
<td>Tatar</td>
<td>37.5</td>
<td>12.5</td>
<td></td>
</tr>
<tr>
<td>Swahili</td>
<td>33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*wav2vec-U uses much less speech audio than prior work: 1.8h vs. 17h for Kyrgyz, 4.6h vs. 17h for Tatar
Discussion

- Very lightweight approach (except for wav2vec 2.0)
- Why does it work? Good audio features are main driver of performance
- Phonemizer still required
- Segment construction
Conclusion

- Pre-training for speech works very well in both low-resource and high-resource setup.

- Cross-lingual training improves low-resource languages.

- Enable speech models with very little or even no labeled training data.

- Make speech technology more ubiquitous and robust.

- Code and models are available in the fairseq GitHub repo + Hugging Face.
Thank you