Wav2vec related paper sharing

ZiXi Yan

2020/04/08

TRILLSSON: DISTILLED UNIVERSAL PARALINGUISTIC SPEECH REPRESENTATIONS

- Recent advances in self-supervision have dramatically im_x0002_proved the quality of speech representations. However, de_x0002_ployment of state-of-the-art embedding models on devices has been restricted due to their limited public availability and large resource footprint.

- In this work, we use CAP12 as the teacher and distill this model to several “lite” architectures based on the teacher-student distillation approach.

TRILLSSON: DISTILLED UNIVERSAL PARALINGUISTIC SPEECH REPRESENTATIONS

- 1. Audio Spectrogram Transformer (AST) is a Transformer-based model for audio classification. We train student models with different depths and widths.
- 2. EfficientNetv2 was designed by neural architecture search on image classification. The architecture is mobile friendly. Different versions of this architecture vary in terms of depths and filters.
- 3. Resnetish are modified ResNet-50 architectures designed to take audio spectral features as input. Different versions of this architecture include different depths and different number of filters per layer.

TRILLSSON: DISTILLED UNIVERSAL PARALINGUISTIC SPEECH REPRESENTATIONS

Table 3: Test performance on the NOSS Benchmark and extended tasks. "Prev. SoTA" are usually domain specific, but all other rows are linear models on time-averaged input. TRILLsson model sizes are shown without frontends. ↑ indicates higher values are better, and ↓ indicates lower is better. †We use a filtered subset of Voxceleb1 according to YouTube's privacy guidelines. We omit previous SoTA results on this dataset, since they used the entire dataset. **ASVSpoof uses equal error rate [18]. We report the best single-model performance (as compared to model ensembles). #Euphonia is the only non-public dataset. We use a larger dataset than was reported on in [4]. *We use the public Wav2Vec 2.0 model from Hugging Face [22]

Model (input size)	Params (M)	Size (MB)	Public	Voxceleb1† ↑	Voxforge ↑	Speech Commands ↑	ASVSpoof 2019** ↓	Euphonia# ↑	CREMA-D ↑	IEMOCAP ↑
Prev. SoTA	-	-	-	-	99.8 [4]	97.9 [23]	2.5 [4]	-	88.2 [4]	79.2 [4]
CAP12										
(full)	606	2,200	✗	51.0	99.7	97.1	2.5	46.9	88.2	75.5
(3s)	606	2,200	✗	47.9	99.4	97.1	3.8	46.9	88.1	74.3
(2s)	606	2,200	✗	48.1	99.4	97.0	6.9	46.9	85.3	72.7
Baselines										
Wav2Vec2 Sm. L6*	93.4	360	✓	17.9	98.5	**95.0**	6.7	48.2	77.4	65.8
Wav2Vec2 Sm.*	93.4	360	✓	1.7	95.9	89.3	11.2	50.0	58.0	52.4
TRILL	24.5	87	✓	13.8	84.5	77.6	6.3	47.0	65.7	55.4
YAMNet	3.7	17	✓	9.6	79.8	78.5	6.7	43.8	66.4	57.5
TRILLsson										
5 (AST)	88.6	314	✓	**46.2**	**99.7**	93.9	**5.4**	48.1	86.1	72.7
4 (AST)	63.4	224	✓	43.1	99.6	94.5	7.1	**50.7**	**86.2**	**73.2**
3 (EffNetv2)	21.5	99	✓	40.1	99.2	93.2	6.8	47.4	83.2	70.3
2 (EffNetv2)	8.1	42	✓	37.5	99.2	92.1	6.6	44.6	82.6	69.8
1 (ResNet)	5.0	22	✓	36.6	98.6	91.2	7.5	43.3	81.3	68.5

WavThruVec: Latent speech representation as intermediate features for neural speech synthesis

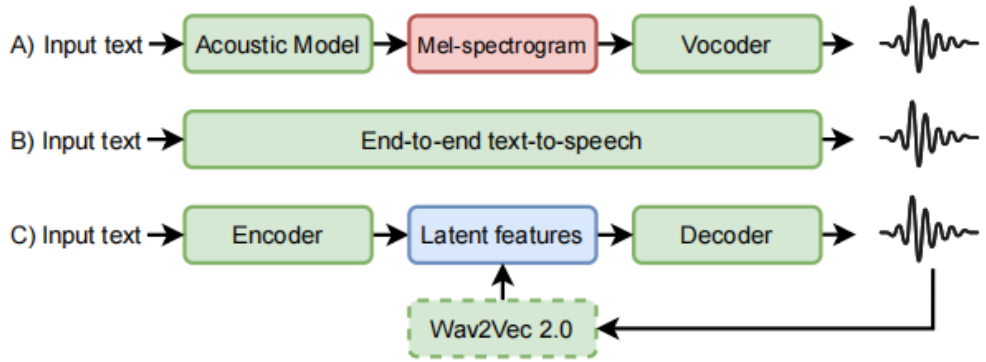

Figure 1: *A high-level comparison of TTS architectures: A) a traditional two-stage pipeline with mel-spectrogram as an intermediate speech representation; B) end-to-end TTS that generates waveform directly from input text; C) a proposed two-stage TTS that leverages latent speech representation from the external, pretrained model. Green blocks represent learnable neural modules, red represents predetermined features, while blue represents hidden representation. The dashed outline indicates that Wav2Vec is freezed during the training and its parameters are not updated.*

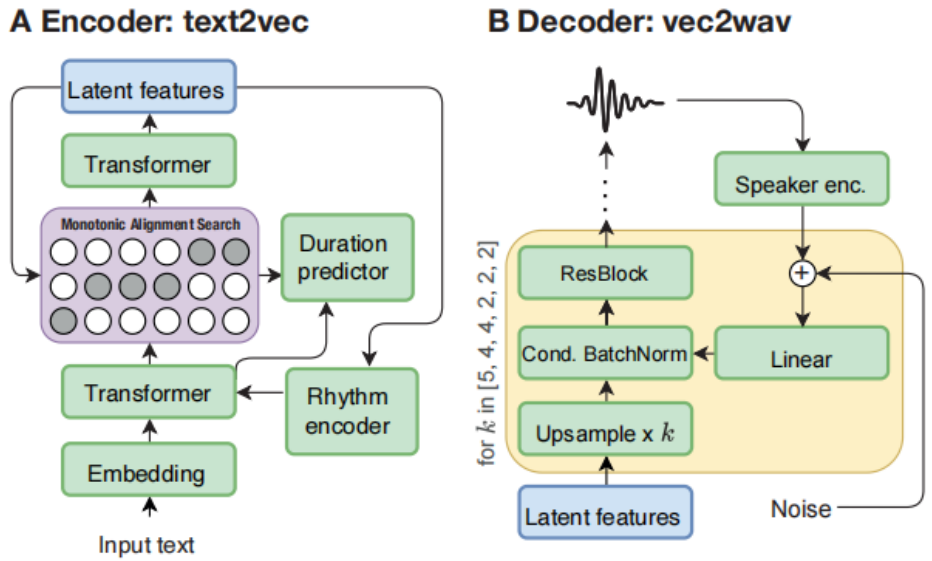

Figure 2: *Our architecture consists of an encoder (text2vec) and a decoder (vec2wav).*

WavThruVec: Latent speech representation as intermediate features for neural speech synthesis

Table 2: *Comparison of evaluated MOS and pronunciation errors (% correct) with 95% confidence intervals*

Model	MOS (CI)	% correct (CI)
Ground Truth	4.17 (±0.10)	–
Tacotron 2	3.92 (±0.13)	0.78 (±0.05)
FastPitch	3.67 (±0.12)	0.82 (±0.05)
VITS	3.99 (±0.12)	0.86 (±0.05)
WavThruVec	**4.09** (±**0.10**)	**0.89** (±**0.04**)

Robust Speaker Recognition with Transformers Using wav2vec 2.0

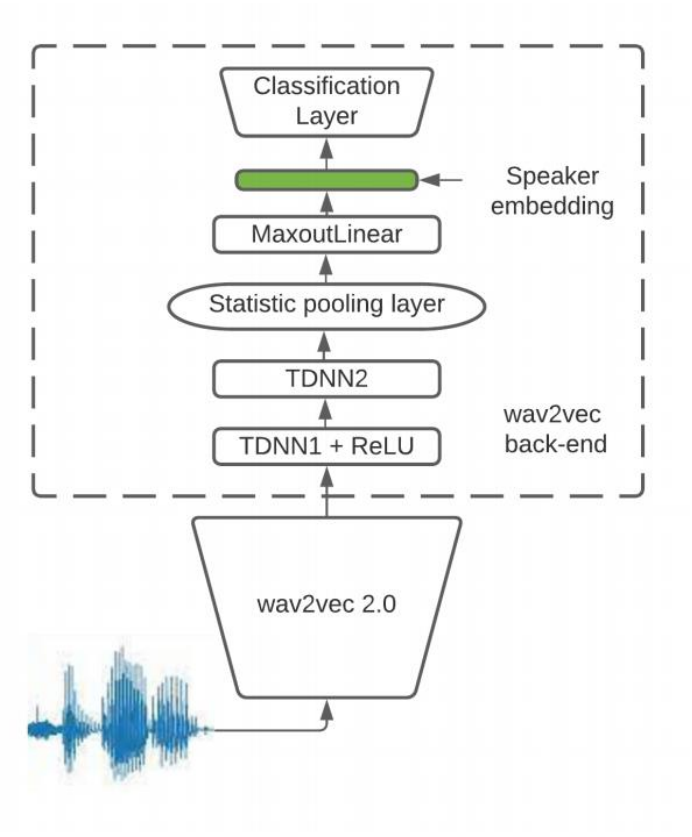

Figure 1: *Wav2vec 2.0 based speaker embeddings extractor*

Table 1: *Results of speaker verification on VC1-O (cleaned) and SRE'18 dev sets in dependence of wav2vec 2.0 encoder output layer selection for wav2vec-TDNN(XLSR_53)*[3]

Layer	Train set	VC1-O (cleaned)		SRE'18 dev	
		EER	DCF(0.01)	EER	DCF(0.01)
3	VC1	2.54	0.29	13.58	0.62
6		1.82	0.22	10.19	0.51
9		1.76	0.197	10.5	0.48
12		1.71	0.21	10.58	0.52
18		**1.61**	**0.17**	**9.97**	**0.44**
24		na[4]	na	na	na

Table 2: *Results of speaker verification on VC1-O (cleaned) test and SRE'18 dev sets in dependence of wav2vec 2.0 encoder output layer selection for wav2vec-TDNN(XLSR_53)*[3]

Layer	Train set	VC1-O (cleaned)		SRE'18 dev	
		EER	DCF(0.01)	EER	DCF(0.01)
3	VC1 + augs	3.47	0.327	12.22	0.55
6		2.37	**0.227**	**9.78**	**0.45**
9		2.23	0.267	10.88	0.48
12		2.38	0.321	10.34	0.45
18		**2.21**	0.243	11.06	0.54
24		16.62	0.99	30	1

Robust Speaker Recognition with Transformers Using wav2vec 2.0

Table 4: *Speaker recognition evaluations on different test protocols for baseline systems and proposed wav2vec 2.0 based systems in terms of EER[%] / minDCF(0.05)*

System	#Params, M	Test datasets						
		SRE'18 dev	SRE'16 eval	SRE'19 eval	VC1-O (cleaned)	VOiCES dev	CTS'20 progress [1]	SRE'21 eval [2]
Baseline encoders								
ResNet101	27.5	3.28/0.118	5.01/0.237	2.39/0.134	1.78/0.105	1.81/0.110	2.75/0.097	5.41/0.344
ECAPA-TDNN	29	4.14/0.152	8.59/0.337	2.97/0.165	1.87/0.123	2.02/0.123	2.91/0.109	6.26/0.398
ResNet101 + ECAPA TDNN	56.5	3.17/0.114	4.87/0.221	2.12/0/122	1.35/0.086	1.31/0.081	2.71/0.085	4.74/0.299
New encoders								
wav2vec-TDNN (XLSR_53)	98	3.07/0.137	4.18/0.206	2.34/0.142	0.82/0.052	0.99/0.06	**2.25/0.080**	4.43/0.283
wav2vec-TDNN (XLS-R_1B)	265	**2.94/0.083**	**3.13/0.161**	**1.71/0.097**	**0.69/0.040**	1.02/0.057	3.61/0.080	**3.59/0.281**

SHAS: Approaching optimal Segmentation for End-to-End Speech Translation

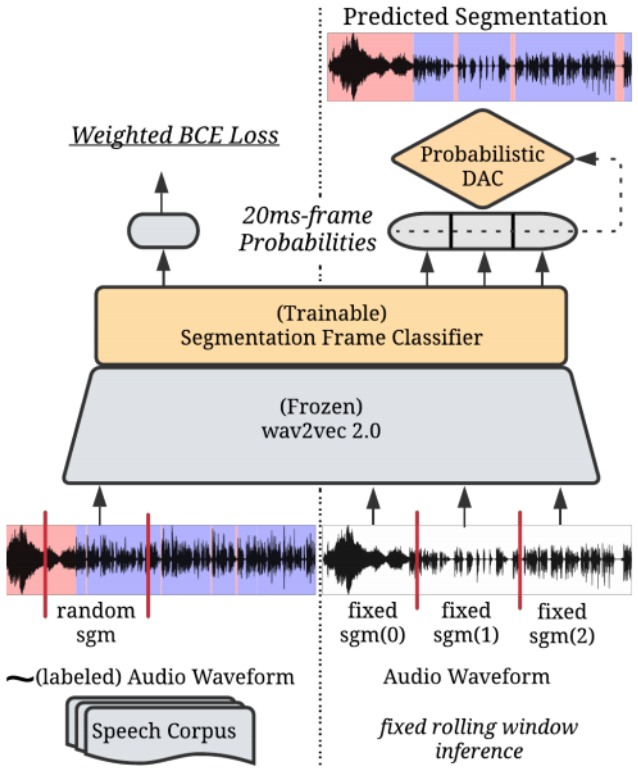

Figure 1: *Supervised Hybrid Audio Segmentation (SHAS). Left: Training procedure. Right: Segmentation at inference.*

SHAS: Approaching optimal Segmentation for End-to-End Speech Translation

Algorithm 1 Probabilistic DAC

```
procedure RECURSIVE_SPLIT(sgm)
    if len(sgm) < max then
        append sgm to segments
    else
        j ← 0
        indices ← argsort probs[sgm]
        while True do
            sgm_a, sgm_b ← split sgm at indices[j]
            sgm_a ← trim(probs[sgm_a], thr)
            sgm_b ← trim(probs[sgm_b], thr)
            if len(sgm_a) > min and len(sgm_b) > min then
                RECURSIVE_SPLIT(sgm_a)
                RECURSIVE_SPLIT(sgm_b)
                break
            j ← j + 1
procedure PROBABILISTIC_DAC(probs, max, min, thr)
    segments ← empty List
    sgm ← Tuple[0, len(probs)]              ▷ init single segment
    RECURSIVE_SPLIT(sgm)
    return segments
```

SHAS: Approaching optimal Segmentation for End-to-End Speech Translation

Table 1: *BLEU scores of SHAS, manual segmentation, and other methods. In parenthesis is the percentage of manual BLEU score retained.* ***(i)****: Main results on MuST-C en-de tst-COMMON and mTEDx x-en test.* ***(ii)****: Cross-domain results on Europarl-ST test.*

Segm. Methods	**en-de**	**es-en**	**fr-en**	**it-en**	**pt-en**	**Average**	**Europarl en-de**
Manual	26.99 (100.)	31.94 (100.)	36.69 (100.)	27.15 (100.)	34.88 (100.)	31.53 (100.)	28.83 (100.)
Length-based (fixed)	22.34 (82.8)	27.71 (86.8)	30.57 (83.3)	23.66 (87.1)	30.21 (86.6)	26.90 (85.3)	22.35 (80.3)
Pause-based (VAD)	22.78 (84.4)	27.03 (84.6)	30.01 (81.8)	21.77 (80.2)	26.58 (76.2)	25.63 (81.3)	18.58 (66.8)
Hybrid VAD-DAC	24.19 (89.6)	29.38 (92.0)	31.85 (86.8)	24.35 (89.7)	30.92 (88.6)	28.14 (89.2)	25.06 (90.1)
Hybrid VAD-STRM	23.75 (88.0)	29.54 (92.5)	31.79 (86.6)	24.72 (91.0)	31.15 (89.3)	28.19 (89.4)	24.08 (86.5)
Hybrid W2V-DAC	24.49 (90.7)	29.69 (93.0)	33.48 (91.3)	24.82 (91.4)	32.48 (93.1)	28.99 (91.9)	24.92 (89.5)
Hybrid W2V-STRM	24.47 (90.7)	29.50 (92.4)	33.40 (91.0)	25.24 (93.0)	32.35 (92.7)	28.99 (91.9)	23.27 (83.6)
SHAS	**25.67 (95.1)**	30.50 (95.5)	35.08 (95.6)	26.38 (97.2)	33.20 (95.2)	30.17 (95.7)	**26.40 (94.9)**
↪ Multilingual	25.61 (94.9)	**30.82 (96.5)**	**35.28 (96.2)**	**26.56 (97.8)**	**33.53 (96.1)**	**30.36 (96.3)**	26.24 (94.3)

Thank