Automatic Data Augmentation for Domain Adapted Fine-Tuning of Self-Supervised Speech Representations

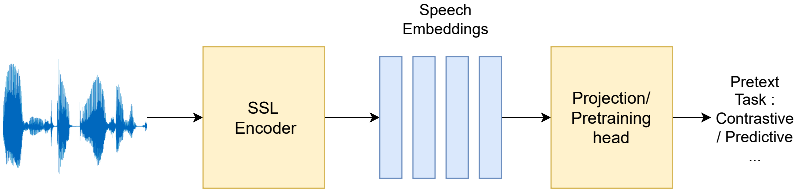
Salah Zaiem, Titouan Parcollet and Slim Essid
salah.zaiem@telecom-paris.fr

INTERSPEECH 2023

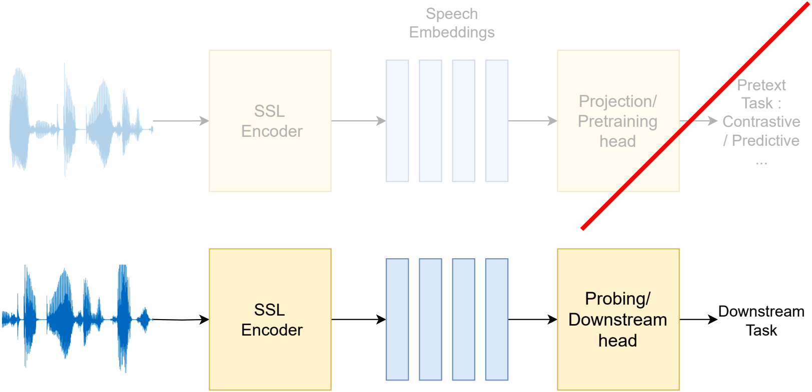
ADASP
audio data analysis signal processing
TELECOM Paris
IP PARIS
Samsung AI Center-Cambridge

Outline

Self supervised learning (SSL)



Self supervised learning (SSL)



Self-supervised Learning (SSL)

- Self-supervised models allowed substantial performance progress in ASR, especially in low-resource scenarios.
- With a few hours of annotated data, reasonable performances can be reached on a target domain.

Example: Fine-tuning Wav2vec2 Large on only 10 hours from LibriSpeech $\rightarrow$ 5% WER on *test-other*.

Domain shift for Self-Supervised Models

When encountering audios with conditions very different from pretraining ones, SSL based models suffer high ASR performance drops.

- Fine-tuning Wav2vec2 Large on only 10 hours from LibriSpeech $\rightarrow$ 5% WER on *test-other*.
- Fine-tuning on 10 hours from CommonVoice (CV) $\rightarrow$ $>$ 25% WER on CV test.

Domain shift for Self-Supervised Models

When encountering audios with conditions very different from pretraining ones, SSL based models suffer high ASR performance drops.

- Fine-tuning Wav2vec2 Large on only 10 hours from LibriSpeech $\rightarrow$ 5% WER on *test-other*.
- Fine-tuning on 10 hours from CommonVoice (CV) $\rightarrow$ $>$ 25% WER on CV test.

Domain shifts are diverse: linguistic, accent-related, prosodic, acoustic.. We will work on acoustic shifts.

Transfer learning

- In low-resource settings, transfer learning using data from other domains is very useful.
- It is less useful when the two domains are acoustically very different.

Question

How can we exploit large annotated datasets from other domains better ?

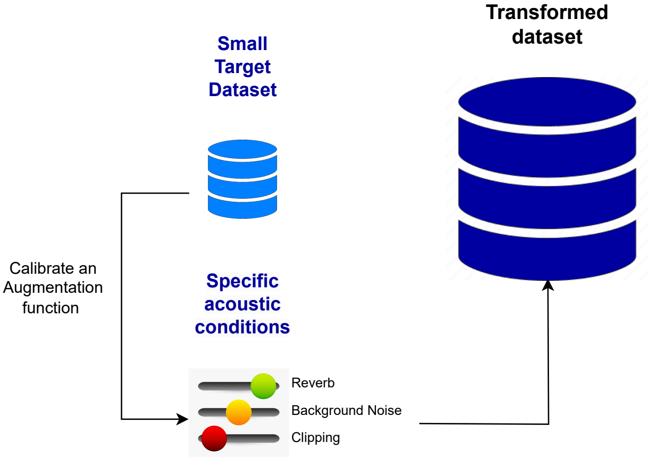
Idea

Small
Target
Dataset

Specific
acoustic
conditions

Large
"Clean"
Dataset

Idea



Transformed dataset
Small Target Dataset
Calibrate an Augmentation function
Specific acoustic conditions
Reverb
Background Noise
Clipping

Outline

Automatic Data Augmentation

Objective: Given a target annotated dataset (X, Y), obtain a data augmentation policy τ imitating its acoustic conditions.

An augmentation distribution τ is defined by a set of parameters defining how a chain of augmentations is sampled during training.

Conditional Independence based estimator

- Inspired by data augmentation selection for contrastive self-supervised pretraining.
- Precisely, we developed a conditional-independence based function that scores a candidate policy τ given a target dataset (X, Y).

S. Zaiem *and al.*, "Automatic Data Augmentation Selection and Parametrization in Contrastive Self-Supervised Speech Representation Learning," in Interspeech 2022

A few definitions and notations

Inputs: target dataset (X, Y), augmentation distribution τ

- f_τ the function that augments audio files sampling from τ.
- X' the dataset consisting of different views of X samples ($X \xrightarrow{f_\tau} X'$).

A few definitions and notations

Inputs: target dataset (X, Y), augmentation distribution τ

- f_τ the function that augments audio files sampling from τ.
- X' the dataset consisting of different views of X samples $(X \xrightarrow{f_\tau} X')$.
- For a point $x \in X'$, we will call z an ID of the original point in X it was generated from and Z the resulting set.
- $HSIC(X', Z)$ is the Hilbert-Schmidt Independence Criterion value for two sets (X', Z). It is positive a value. The lower, the more independent X' and Z are.

Gretton, A. *and al.* (2007). A Kernel Statistical Test of Independence. In NeurIPS

Computation steps

Inputs: target dataset (X, Y), augmentation distribution τ

1. Creating N views per audio: $X \xrightarrow{f_\tau(x)} X'$ with f_τ the function that augments audio segments sampling from τ.
2. Split the audio samples per downstream class (word identity). Segments obtained with force-alignment.

Computation steps

Inputs: target dataset (X, Y), augmentation distribution τ

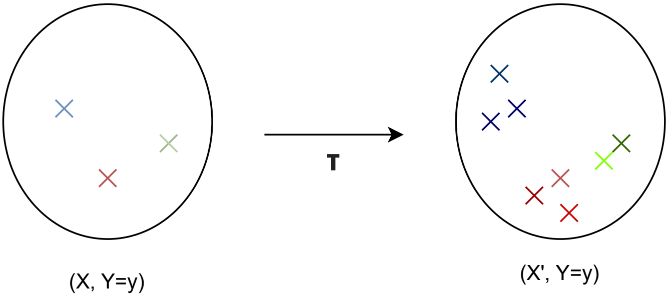
1. Creating N views per audio: $X \xrightarrow{f_\tau(x)} X'$ with f_τ the function that augments audio segments sampling from τ.
2. Split the audio samples per downstream class (word identity). Segments obtained with force-alignment.
3. With $\mathscr{C}$ the set of classes, and $HSIC_c(X', Z)$ the independence test value for points sharing the class (*i.e.* word) c, compute:

$$HSIC(X', Z|Y) = \frac{1}{M} \sum_{c \in \mathscr{C}} HSIC_c(X', Z) \times n_c.$$

The lower this value, the better τ is for acoustic condition cloning.

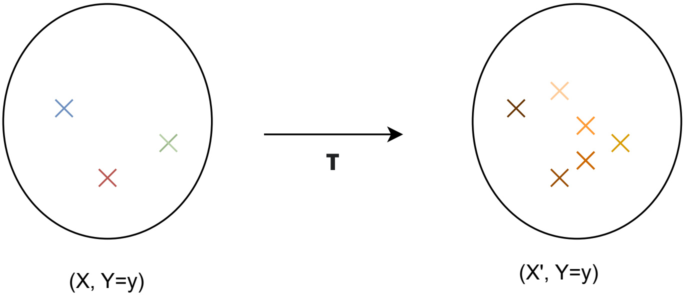
$$\tau^* = \arg\min_\tau HSIC(X', Z|Y)$$

Some intuition



No independence in the second circle $\longrightarrow$ high *HSIC* value

Some intuition



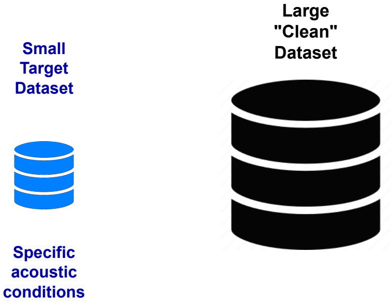
Independence $\longrightarrow$ low *HSIC* value

- Collapse is prevented by fixing limits to the augmentations sampled.

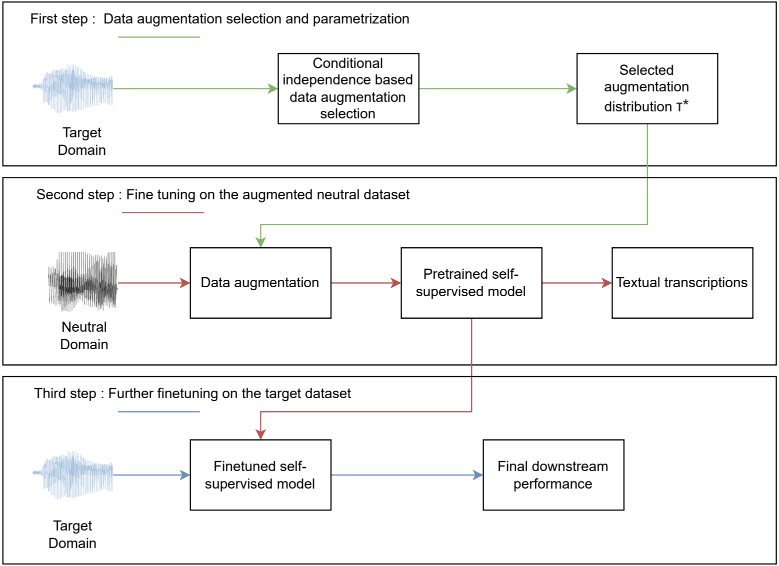
Outline

Inputs

- Annotated target dataset D_T, small size, and specific acoustic conditions.
- Large clean ("Neutral") dataset D_C



Steps


First step : Data augmentation selection and parametrization
Target Domain
Conditional independence based data augmentation selection
Selected augmentation distribution τ^*
Second step : Fine tuning on the augmented neutral dataset
Neutral Domain
Data augmentation
Pretrained self-supervised model
Textual transcriptions
Third step : Further finetuning on the target dataset
Target Domain
Finetuned self-supervised model
Final downstream performance

Settings

- SSL Model: Wav2vec 2.0 Large.
- Downstream head: linear decoder trained with CTC Loss. No language modelling for decoding.
- Downstream classes: 20 most common words longer than 5 characters. We cut at word level using forced alignment.

Baevski, A. *and al.* (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.

Selecting the Augmentation Distribution

An augmentation distribution τ is defined by a set of parameters defining how a chain of augmentations is sampled during pretraining.
Set of considered augmentations:

- Reverberation
- Low/High passing
- Pitch Shifting
- Gain
- Polarity Inversion
- Time dropping
- Coloured noise addition

Selecting the Augmentation Distribution

Every distribution τ is represented as a vector of $P = 17$ parameters.
Probabilities of applying an augmentation / controlling parameters.

Name	**Description**	**Range (Unit)**
Low Min	Lowpass minimal frequency cutoff	[100-500] (Hz)
Low Max	Lowpass maximal frequency cutoff	[1000-5000] (Hz)
High Min	Highpass minimal frequency cutoff	[1000,4000] (Hz)
High Max	Highpass maximal frequency cutoff	[4000,6000] (Hz)
Pitch min	Minimal pitch shift	[-6,-2] (semitones)
Pitch max	Maximal pitch shift	[2,6] (semitones)
Min SNR	Minimal SNR for coloured noise	[0,5] (dB)
Max SNR	Maximal SNR for coloured noise	[10,30] (dB)
Min Gain	Minimal gain	[-20,-10] (dB)
Max Gain	Maximal gain	[3,10] (dB)

Table: Augmentations, descriptions and parameter ranges

To minimize the described HSIC, we resort to a random search among the parameters.

Oracle Experiments

We apply a known augmentation distribution to the test splits of Librispeech and use it as target domain. (10 times)

1. Sample an augmentation distribution τ.
2. Apply τ on LibriSpeech test splits to create the testing sets and on *dev-clean* to create D_T.

Oracle Experiments

We apply a known augmentation distribution to the test splits of Librispeech and use it as target domain. (10 times)

1. Sample an augmentation distribution τ.
2. Apply τ on LibriSpeech test splits to create the testing sets and on *dev-clean* to create D_T.
3. Apply our method on D_T to obtain a distribution τ^*.
4. Apply τ^* on LibriSpeech *train-clean-100* to obtain D_{CT}, then use D_{CT} for training.

Oracle Experiments

We apply a known augmentation distribution to the test splits of Librispeech and use it as target domain. (10 times)

1. Sample an augmentation distribution τ.
2. Apply τ on LibriSpeech test splits to create the testing sets and on *dev-clean* to create D_T.
3. Apply our method on D_T to obtain a distribution τ^*.
4. Apply τ^* on LibriSpeech *train-clean-100* to obtain D_{CT}, then use D_{CT} for training.

Warning

Only **one** fine-tuning is done in the controlled Oracle experiment as we only have a test target dataset.

Oracle Experiments

Split	No Aug	Default	Random	CI Augment	Topline
test-clean	33.81	29.86	29.91	**27.20**	26.11
test-other	44.12	43.89	42.48	**40.68**	36.92

Table: Mean WER results on distorted versions of LibriSpeech test splits

- Default: All augmentations with default parameters.
- Random: Mean of 9 runs with random augmentation policies during training.
- **Topline**: Applying the test distortions on the train data.

Experiments on Real Distorted Datasets

Conditions for the datasets:

- Small target distorted dataset => interesting challenge
- Textual correspondance with the neutral dataset (read speech)
- Coherent and regular acoustic conditions.

We took the samples from the most productive contributors of CommonVoice 11.0, and selected two contributors respecting the conditions. (7 and 9 hours of data)

Experiments on Real Distorted Datasets

Two steps fine-tuning, first on LibriSpeech *train-clean-100* (distorted or not), and second on the contributor data.

Contributor	**Without Augmentations**			**With Augmentations**		
	Train100	Only Contrib.	Train100 + Contrib.	Default	Random	CI Augment
Contributor 1	102.52	73.0	27.71	27.95	27.33	**24.27**
Contributor 2	96.49	98.92	20.48	20.76	22.23	**16.49**

Table: WER Results on the two considered CommonVoice contributors.

Varying the available annotation

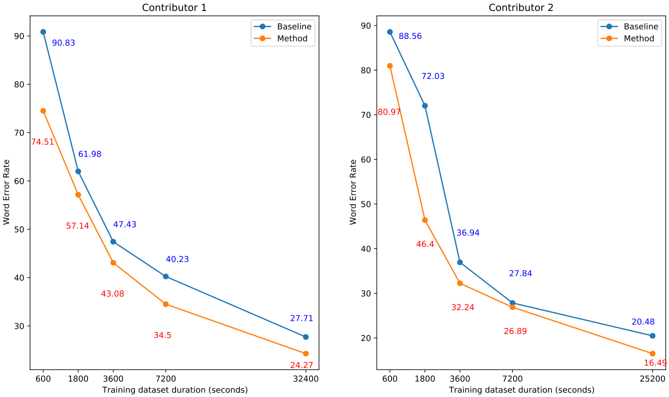


Figure: Effect of choosing suitable augmentations on the performance depending on the quantity of in-domain training data for each of the two considered contributors.

Conclusion

Our approach allows for automatic data augmentation for better adaptation of self-supervised speech models.

- Main strengths : efficient and good in very low resource scenarios.
- Limitations: Limited to acoustic mismatch and need to see the effect of bigger "Neutral" datasets.

Thank You

Thank you all for your attention !

Please feel free to ask any question.

Oracle Experiments

The oracle experiments has two main advantages:

1. Ensuring that the distortions in the target set can be replicated with the considered set of augmentations.
2. Allows to compare the obtained distribution $\tau*$ with the one used to create the target.

After results

- We observe a Spearman correlation score of 0.51 between the HSIC scores and the distances between vectors of probabilities.
- Furthermore, the application probabilities of the 10 (top 5%) best scoring distributions are 15% closer to the target ones than those of the 10 worst scoring ones.