Towards Learning a Universal Non-Semantic Representation of Speech

Joel Shor\textsuperscript{1}, Aren Jansen\textsuperscript{2}, Ronnie Maoz\textsuperscript{1}, Oran Lang\textsuperscript{1}, Omry Tuval\textsuperscript{1}, Félix de Chaumont Quiry\textsuperscript{3}, Marco Tagliasacchi\textsuperscript{3}, Ira Shavit\textsuperscript{1}, Dotan Emanuel\textsuperscript{1}, Yinnon Haviv\textsuperscript{1}

\textsuperscript{1}Google Research, Israel
\textsuperscript{2}Google Research, Mountain View
\textsuperscript{3}Google Research, Zurich

joelshor@google.com

Abstract

The ultimate goal of transfer learning is to reduce labeled data requirements by exploiting a pre-existing embedding model trained for different datasets or tasks. The visual and language communities have established benchmarks to compare embeddings, but the speech community has yet to do so. This paper proposes a benchmark for comparing speech representations on non-semantic tasks, and proposes a representation based on an unsupervised triplet-loss objective. The proposed representation outperforms other representations on the benchmark, and even exceeds state-of-the-art performance on a number of translation tasks. The embedding is trained on a publicly available dataset, and it is tested on a variety of low-resource downstream tasks, including personalization tasks and medical domains. The benchmark, model, and evaluation code are publicly released.

1. Introduction

One of the most powerful uses of deep learning is finding a good representation for a given domain. Despite progress on representations in the visual domain \cite{1} and the language domain \cite{2}, no such universal representation exists for the speech domain. One reason is a lack of standard benchmark tasks to compare different methods; for example, existing speech representations tend to focus on one problem set such as speaker recognition or speech emotion recognition \cite{3}. In this paper, we propose a set of benchmark speech tasks that are diverse, to require that “good” representations contain general speech information, and targeted, to allow good performance when compared with task-specific representations.

We propose a specific set of publicly available tasks, called the “Non-Semantic Speech benchmark” (NOSS), to assess the general usefulness of speech representations on “non-semantic” tasks. What we call “non-semantic” tasks do not include tasks like automatic speech recognition and phone classification, which require sub-second granularity. They do include paralinguistic tasks such as speech emotion recognition, as well as tasks such as speaker identification, language identification, and medical diagnosis. In addition, we introduce a new set of intra-speaker sub-tasks from existing tasks where a model is trained and evaluated on speech from a single speaker. These intra-speaker tasks help measure which representations are most useful for personalization, which is an increasingly-relevant use-case as more computation is performed on-device.

A good speech representation should be high-performing on a diverse set of downstream tasks using simple models. In addition, it should be useful in transfer learning with small amounts of data for a new task. This use-case is relevant for model personalization, such as user-specific emotion recognition or speaker identification.

We introduce a representation, TRILL (TRIpLet Loss network), which is learned in a self-supervised manner on speech containing clips from AudioSet \cite{4}. Using the techniques of \cite{5}, the network represents audio such that segments which are closer in time are also closer in the embedding space. We demonstrate that this simple proxy objective is highly effective in learning a strong representation for multiple non-semantic speech tasks.

We evaluate TRILL and other representations on our benchmark by training small models built on top of the representations and comparing their performances. In addition, we explore transfer learning by fine-tuning TRILL using data from the downstream tasks. This is an advantage of learned representations over non-learned ones. Pre-training via transfer learning can sometimes outperform models trained on a single dataset \cite{1}, and this is also the case in our benchmark. Using transfer learning, we are able to achieve a new state-of-the-art in many of the tasks, surpassing previously published results which sometimes were hand-crafted for those specific datasets.

In summary, our contributions are:

1. We define a new benchmark for comparing representations on non-semantic speech tasks using previously published data. In addition, we add a sub-category of personalization tasks.
2. We demonstrate that a single representation learned in an unsupervised manners performs best on this benchmark. We compare it to existing representations, feature-based and learned.
3. We fine-tune our best performing representation, further boosting results. This method sets a new state-of-the-art on many previously published tasks.
4. We distill our learned representation to a model that can run inference and training on-device, and we open-source the original and distilled models.

2. Background

Transfer learning and domain adaptation have been extensively studied in machine learning \cite{12}. Recent research has mostly focused on deep representation learning methods, either supervised, semi-supervised, or unsupervised. Successful representations improve the sample efficiency of ML algorithms by extracting most information out of the raw signal from the new
There are many methods for learning audio representations. The work in [16] trained an embedding for audio classification on AudioSet. Other work has demonstrated the value of supervised [18,17], semi-supervised [19], or unsupervised representations. The unsupervised audio representation literature is especially diverse (L^2 [20], AuDeep [21], Autoregressive Predictive Coding [22], Contrastive Predictive Coding [23], metric learning [5], autoencoding [24]). However, these methods were evaluated on just one or a limited set of downstream tasks.

In other domains, training a strong representation requires a very large, general dataset. AudioSet [4] is the largest dataset for general purpose audio machine learning, serving as an audio equivalent of ImageNet. Even when restricted to only the samples with speech tags, it surpasses all datasets in size and variability. It has been used to learn a general purpose audio embedding in [16], and can be used for multiple speech tasks.

### 3. Non-Semantic Speech Benchmark (NOSS)

To standardize the assessment of non-semantic speech representations, we introduce NOSS. This section describes the benchmark in detail (summarized in Table 1). These tasks reflect different properties of the speech signal, and they vary in size and difficulty. Personalization and on-device training is increasingly important, so we include “Intra Speaker” tasks for the datasets when applicable. Intra-speaker tasks are an important addition because they also test task adaptation for small amounts of data, and that representations do not just rely on speaker identity.

### 4. Experiments

#### 4.1. TRILL representation

Non-semantic aspects of the speech signal (e.g., speaker identity, language, and emotional state) generally change more slowly than the phonetic and lexical aspects used to explicitly convey meaning. Therefore, we expect a good representation for non-semantic downstream tasks to be considerably more stable in time than what is required for ASR applications. To take advantage of this intuition, we follow the work of [5] (Section 3.2.5) and use temporal proximity as a self-supervision signal.

More formally, consider a large, unlabeled speech collection represented as a sequence of spectrogram context windows $X = x_1 x_2 \ldots x_N$, where each $x_i \in \mathbb{R}^{F \times T}$. Our goal is to learn a map $g : \mathbb{R}^{F \times T} \rightarrow \mathbb{R}^d$ from spectrogram context windows to d-dimensional space such that $\|g(x_i) - g(x_j)\| \leq \|g(x_i) - g(x_k)\|$ when $|i - j| \leq |i - k|$. We can express this desired relationship as a learning objective using triplet loss-based metric learning as follows. First, we sample from $X$ a large collection of example triplets of the form $z_i = (x_i, x_j, x_k)$ (the so-called anchor, positive, and negative examples), where $|i - j| \leq \tau$ and $|i - k| > \tau$ for some suitably chosen time scale $\tau$. The loss incurred by each triplet is then given by

$$L(z) = \sum_{i=1}^{N} \left( \|g(x_i) - g(x_j)\|_2^2 - \|g(x_i) - g(x_k)\|_2^2 + \delta \right)_{+},$$

where $\|\cdot\|_2$ is the $L_2$ norm, $[\cdot]_+$ is standard hinge loss, and $\delta$ is a
4.2. Other representations

We compared TRILL to several representations used frequently on non-semantic tasks. These include classical methods like Mel spectrograms and OpenSmile [30], which is the de-facto standard in emotion recognition [3]. For Mel spectrograms, we tested many different configurations and selected the best one on each task, and for OpenSmile we used the ComParE16 acoustic parameter set [31].

We also compared to existing learned representations. YAMNet [32] is a state-of-the-art network for audio classification [16], trained on AudioSet. VGGish [18] is an audio embedding produced by training a modified VGGNet model to predict video-level tags from the Youtube-8M dataset. We also compared to our own network with random initialization, a technique which has been shown to produce good embeddings [33].

4.3. Experimental Method

4.3.1. Evaluation methods

In order to evaluate the usefulness of the representations described in Section 4.2, we train small models to solve the downstream NOSS tasks (Section 3). For each representation / task pair, we explore different downstream models, representation aggregation techniques, and normalization methods.

We train shallow models using Scikit-Learn [34]. We experimented with logistic regression, random forest classifiers, and linear discriminator analysis (LDA) as the downstream models. Despite the shallowness of the above models, we achieve competitive results with the best previously-reported results on some of the benchmark tasks.

The embeddings were aggregated over time using average pooling, except for VoxCeleb which used NetVLAD aggregation [28]. Datasets which have multiple utterances per speaker were normalized for each speaker using $L^2$ normalization.

Some of the downstream tasks have fixed canonical splits. For the inter-speaker tasks on those datasets, we report numbers on the canonical splits (Speech Commands and VoxCeleb). For the other datasets, and for all the intra-speaker tasks, we perform five random train / test splits and report the average. For intra-speaker tasks (Section 3), we train and test on one speaker at a time, then average results across splits and across speakers.

Table 2: Average performance of the different embeddings on a number of downstream tasks.

|                  | VoxCeleb1* | VoxForge | SpeechCommands | CREMA-D | SAVEE | DementiaBank |
|------------------|------------|----------|----------------|---------|-------|--------------|
| Prev SOTA        | -          | 89.0 [25]| 91.1 [26]      | 74.0    | 61.0  | 80.6 [28]    |
| Mel / MFCC       | 12.2       | 79.2     | 47.7           | 52.8    | 55.6  | 67.7         |
| OpenSmile        | 2.5        | 78.0     | 36.5           | 53.7    | 62.6  | 68.8         |
| Random Network   | 11.0       | 73.0     | 42.0           | 52.1    | 48.6  | 67.9         |
| YAMNet top       | 3.1        | 67.0     | 40.7           | 52.2    | 45.4  | 64.8         |
| YAMNet layer 10  | 16.6       | 86.1     | 73.1           | 66.4    | 62.3  | 70.0         |
| VGGish top       | 12.5       | 80.8     | 28.3           | 51.3    | 49.8  | 68.3         |
| VGGish FCN 1     | 14.8       | 85.1     | 52.7           | 55.7    | 57.7  | 68.7         |
| TRILL top        | 16.4       | 83.8     | 60.4           | 64.9    | 53.7  | 68.2         |
| TRILL layer 19   | 17.7       | 88.1     | 74.0           | 67.8    | 67.8  | 67.2         |
| TRILL layer 19, MobileNet 2048d | 17.9 | 83.4     | 74.9           | 68.1    | 60.0  | 68.1         |
| TRILL finetuned  | 17.6       | 94.1     | 91.2           | 69.5    | 68.6  | 73.1         |

*Results in our study use a small subset of Voxceleb1 filtered according to YouTube’s privacy guidelines. Interested readers can run our study on the full dataset using TensorFlow Datasets and our open-source evaluation code. State-of-the-art performance on the un-filtered dataset can be found in [6]. † Also uses video features. ‡ Also uses text features.
Table 3: Intra-speaker task performance

| CREMA-D | SAVEE | Speech Commands |
|---------|-------|-----------------|
| Mel / MFCC | 47.7  | 77.2 | 73.3  |
| OpenSmile | 47.6  | 67.8 | 72.2  |
| Random Network | 43.4  | 73.9 | 69.0  |
| YAMNet top | 39.1  | 64.4 | 70.3  |
| YAMNet layer 10 | 50.8  | 79.2 | 77.6  |
| VGGish top | 43.1  | 68.7 | 70.2  |
| VGGish FCN 1 | 50.5  | 77.9 | 73.1  |
| TRILL top | 56.8  | 83.9 | 72.2  |
| TRILL layer 19 | 57.0  | 84.7 | 74.7  |
| TRILL distilled | 56.8  | 84.7 | 74.7  |

Figure 1: Effect of model on accuracy ($R^2 = 91\%$). A linear regression on the observed accuracies, with both the model and task as the explanatory variables. The effect a model has on the accuracy is the coefficient associated with the model in the regression. For a given task, when changing from one model to another, the resulting change in accuracy is expected to be the difference in y values in this figure.

We experimented with fine-tuning the entire model end-to-end. For benchmarks tasks with relatively small amounts of data, we applied early stopping.

We also experimented with intra-speaker fine-tuning. For this we used the CREMA-D dataset, with fixed train/dev/test splits of 80%/10%/10%. We first fine-tuned a common model on the training partitions of all the speakers, then further fine-tuned on each speaker separately.

5. Results

TRILL outperforms previously reported results on three of the six benchmark tasks, and is competitive with previous best on two of the three remaining tasks. On these two tasks, the best previously reported numbers use other modalities in addition to audio (visual or textual features) (Table 2). Of the representations we compared against, TRILL performed the best on five-of-six tasks (Table 2) and two-of-three intra-speaker tasks (Table 2). We successfully distilled TRILL to a much smaller model that can be used on a mobile device. The distilled model has no performance degradation on five-of-nine tasks, statistically insignificant degradation on one, and minor degradation on the remaining tasks. We compare across all tasks by fitting a linear regression on the observed accuracies, with both the model and task as the explanatory variables (see Figure 1).

The distilled model results are presented in the one-before-last line of Table 2. With the exception of VoxForge language identification and SAVEE speech emotion recognition, the reduction in model capacity and dimensionality has performance within the standard deviation of the larger embedding (variance calculated over 5-fold data splits). In the personalized tasks the quality of the distilled model is the same as the larger model.

6. Analysis

As can be seen in Table 2, fine-tuning the final embedding gives a clear boost on most tasks and sets a new state-of-the-art in 3 out of the 6 datasets. This approach of learning a strong representation on a large benchmark and then fine-tuning it for the downstream task has been proven very effective in other domains [1], and in this paper we demonstrate the same is true in the speech domain.

An important observation of our research is that the effective representation learned by both YAMNet and TRILL is not at their final layer, but in fact in one of their intermediate layers. This intermediate layer must capture information which is later discarded in the final layers. The reasoning might be that when learning the triplet loss, the network might learn to discard properties which vary temporally such as tone of voice or semantic information, but this embedding is still learned in the intermediate layer. Another evidence for that can be seen in the results on the Speech Commands dataset, where the performance of the intermediate layers of all of the learned representations is much better than the performance of their respective top layer.

Table 4 shows that fine-tuning per speaker allows to further personalize to each speaker, and generally improves accuracy. When breaking down the performance impact on each speaker, we can see per-speaker fine-tuning improves accuracy for 31 speakers, is mostly unchanged for 49 speakers, and decreases for 12 speakers.

7. Conclusions

In this work, we explore the importance of clearly defining benchmarks when comparing representations of speech. We propose NOSS (Section 3) to help fairly compare non-semantic speech representations, and we introduce a sub-category of personalization tasks to help measure progress in the age of on-device computation. We also demonstrate that TRILL (Section 4.1), based on a self-supervised training criteria, simultaneously performs well on all benchmark tasks. We show that finetuning TRILL on a small amount of data outperforms or is competitive with almost all previously reported numbers for the NOSS tasks, and that TRILL is significantly better than other representations. Finally, we distill TRILL to be on-device with very little or no performance loss. The NOSS benchmark, the evaluation code, TRILL, and TRILL-distilled are all publicly available.
8. References

[1] X. Zhai, J. Puigcerver, A. Kolesnikov, P. Ruyssen, C. Riquelme, M. Lucic, J. Djolonga, A. S. Pinto, M. Neumann, A. Dosovitskiy, L. Boyer, O. Bachem, M. Tschantz, M. Michalski, O. Bousquet, S. Gelly, and N. Houlsby, “The visual task adaptation benchmark,” 2019.

[2] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” CoRR, vol. abs/1804.07461, 2018. [Online]. Available: http://arxiv.org/abs/1804.07461

[3] S. Latif, R. Sana, S. Khalifa, R. Juridak, J. Qadir, and B. W. Schuller, “Deep representation learning in speech processing: Challenges, recent advances, and future trends,” 2020.

[4] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 776–780.

[5] A. Jansen, M. Plakal, R. Pandya, D. P. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, “Unsupervised learning of semantic audio representations,” CoRR, vol. abs/1711.02209, 2017. [Online]. Available: http://arxiv.org/abs/1711.02209

[6] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: A large-scale speaker identification dataset,” Interspeech 2017, Aug 2017. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-950

[7] K. MacLean, “Voxforge,” Ken MacLean.[Online]. Available: http://www.voxforge.org/home [Acedido em 2012], 2018.

[8] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” 2018.

[9] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, “CREMA-D: Crowd-sourced emotional multimodal actors dataset,” IEEE transactions on affective computing, vol. 5, no. 4, pp. 377–390, 2014.

[10] S. Haq, P. J. Jackson, and J. Edge, “Speaker-dependent audio-visual emotion recognition.” in AVSP, 2009, pp. 53–58.

[11] F. Boller and J. Becker, “Dementiabank database guide,” University of Pittsburgh, 2005.

[12] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[13] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in International conference on artificial neural networks. Springer, 2018, pp. 270–279.

[14] V. Cheplygina, M. de Bruijne, and J. P. Pluim, “Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis,” Medical image analysis, vol. 54, pp. 280–296, 2019.

[15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3320–3328.

[16] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, “Panns: Large-scale pretrained audio neural networks for audio pattern recognition,” arXiv preprint arXiv:1912.10211, 2019.

[17] S. Pascual, M. Ravanelli, J. Serr, A. Bonafonte, and Y. Bengio, “Learning Problem-Agnostic Speech Representations from Multiple Self-Supervised Tasks,” in Proc. Interspeech 2019, 2019, pp. 161–165. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2605

[18] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. W. Wilson, “CNN architectures for large-scale audio classification,” CoRR, vol. abs/1609.09430, 2016.

[19] S. Parthasarathy and C. Busso, “Ladder networks for emotion recognition: Using unsupervised auxiliary tasks to improve predictions of emotional attributes,” Interspeech 2018, Sep 2018. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1391

[20] R. Arandjelovic and A. Zisserman, “Look, listen and learn,” CoRR, vol. abs/1705.08168, 2017. [Online]. Available: http://arxiv.org/abs/1705.08168

[21] M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. W. Schuller, “audaeep: Unsupervised learning of representations from audio with deep recurrent neural networks,” CoRR, vol. abs/1712.04382, 2017. [Online]. Available: http://arxiv.org/abs/1712.04382

[22] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, “An unsupervised auto-regressive model for speech representation learning,” Interspeech 2019, Sep 2019. [Online]. Available: http://dx.doi.org/10.21437/interspeech.2019-1473

[23] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” CoRR, vol. abs/1807.03748, 2018. [Online]. Available: http://arxiv.org/abs/1807.03748

[24] S. Latif, R. Sana, J. Qadir, and J. Epps, “Variational autoencoders for learning latent representations of speech emotion: a preliminary study,” Interspeech 2018: Proceedings, pp. 3107–3111, 2018.

[25] S. Revay and M. Teschke, “Multiclass language identification using deep learning on spectral images of audio signals,” CoRR, vol. abs/1905.04348, 2019. [Online]. Available: http://arxiv.org/abs/1905.04348

[26] Kaggle, “Tensorflow speech recognition challenge,” https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/leaderboard accessed: 2020-01-28.

[27] E. Ghaleb, M. Popa, and S. Asteriadis, “Multimodal and temporal perception of audio-visual cues for emotion recognition,” in 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII). IEEE, 2019, pp. 552–558.

[28] Z. Noorrian, C. Pou-Prom, and F. Rudzicz, “On the importance of normative data in speech-based assessment,” CoRR, vol. abs/1712.00069, 2017.

[29] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[30] F. Eyben, M. Wöllmer, and B. Schuller, “openSMILE: The Munich versatile and fast open-source audio feature extractor,” in Proceedings of the 18th ACM international conference on Multimedia ACM, 2010, pp. 1459–1462.

[31] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanimi, “The INTERSPEECH 2016 computational paralinguistics challenge: Deception, sincerity and native language,” in Proc. Interspeech 2016, 09 2016, pp. 2001–2005.

[32] M. Plakal and D. Ellis, “Yamnet,” Jan 2020. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/audioset/yamnet

[33] Y. Tian, C. Xu, and D. Li, “Deep audio prior,” 2019.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, M. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[35] R. Arandjelović, P. Gronat, A. Torii, T. Papaja, and J. Sivic, “Netvlad: CNN architecture for weakly supervised place recognition,” 2015.

[36] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, F. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” CoRR, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861