SparseSpeech: Unsupervised Acoustic Unit Discovery with Memory-Augmented Sequence Autoencoders

Benjamin Milde, Chris Biemann

Language Technology Group, Dept. of Informatics, Universität Hamburg

{milde,biemann}@informatik.uni-hamburg.de

Abstract

We propose a sparse sequence autoencoder model for unsupervised acoustic unit discovery, based on bidirectional LSTM encoders/decoders with a sparsity-inducing bottleneck. The sparsity layer is based on memory-augmented neural networks, with a differentiable embedding memory bank addressed from the encoder. The decoder reconstructs the encoded input feature sequence from an utterance-level context embedding and the bottleneck representation. At some time steps, the input to the decoder is randomly omitted by applying sequence dropout, forcing the decoder to learn about the temporal structure of the sequence. We propose a bootstrapping training procedure, after which the network can be trained end-to-end with standard back-propagation. Sparsity of the generated representation can be controlled with a parameter in the proposed loss function. We evaluate the units with the ABX discriminability on minimal triphone pairs and also on entire words. Forcing the network to favor highly sparse memory adressings in the memory component yields symbolic-like representations of speech that are very compact and still offer better ABX discriminability than MFCC.

Index Terms: unsupervised learning, sparse autoencoders, acoustic unit discovery

1. Introduction

Unsupervised or zero resource speech processing is a relatively new and growing field that deals with speech processing setups where usually no transcriptions or labels are known. In many tasks, only raw audio data is available. One of the first successful applications is voice search by example, where a large collection of audio data can be queried by a voice query [1].

Transcribed and labeled speech data is needed to train supervised speech recognition systems, but is usually costly to obtain. Unlabeled speech data on the other hand is much easier to obtain in larger quantities, even for languages for which much less resources are available as compared to e.g. English. In this paper, we consider the problem of inducing an acoustic unit inventory [2, 3] and perform self-labeling of untranscribed speech data. Such a system can be a building block of systems that learn lexical inventories [4] from speech data alone, or could possibly aid in augmenting or replacing linguistically motivated phonemes in supervised automatic speech recognition (ASR). It might also allow speech processing in languages where no transcribed resources are available.

Variance and variability in recordings of speech and its representations (e.g. FBANK, MFCC) are a common problem in automatic speech processing tasks, whether supervised or unsupervised. Speaker and environment characteristics, distance to, and the type of microphone used will cause large differences in acoustic speech representations, making (direct) similarity comparisons difficult.

We propose a novel neural architecture based on a sequence autoencoder. We force the encoder and decoder to develop a symbolic-like representation, with the goal of reconstructing input speech representation with limited information. We aim to solve the problem of speech variability that the automatic units should be able to capture by using an utterance-level context embedding. The decoder can use this embedding to infer the "style" of the utterance in addition to its (sparse) input representation generated by the encoder.

2. Related Work

Several approaches to unsupervised acoustic modeling and acoustic unit discovery have been proposed:

Segmental approaches: These approaches separate segmentation and clustering of acoustic units. Garcia and Gish [5] developed one of the first of such systems. Segments are found based on spectral discontinuities in the signal and the clustering algorithm uses the polynomial trajectory of the cepstral features to compare speech units of varying length.

Autoencoder approaches: Badino et al. [6] proposed k-means on framewise binarized autoencoder representations, where temporal smoothing of the frame level representations can be achieved with Hidden Markov Models (HMMs). Several autoencoder architectures are compared in [7], standard bottleneck autoencoders do not seems to produce representation with better minimal pair discriminability than MFCC features.

Bayesian non-parametric models: Lee and Glass [8] proposed a Dirichlet process mixture model, where each mixture is a HMM representing an acoustic unit. Chen et al. [9] proposed Dirichlet process Gaussian mixture model (DPGMM) clustering for acoustic frame-level unit modeling in their winning entry for unsupervised acoustic unit modeling of the ZeroSpeech 2015 challenge on acoustic sub-word modeling [10]. Heck et al. [11] proposed DPGMM clustering on PLP features followed by obtaining unsupervised feature transformations by retraining with a speaker-independent GMM-HMM on the obtained labels. This is also the winning entry for automatic unit discovery of the ZeroSpeech challenge 2017 [12]. Noteworthy is that while the re-training using a GMM-HMM with speaker adaptations improves on the DPGMM results, the DPGMM clustering on its own on PLP features on a frame-level basis already provides strong results.

The 2015 and 2017 ZeroSpeech challenges also targeted acoustic unit discovery [10, 12] as part of the evaluation and have established the use of the ABX discriminability [13, 14] to intrinsically compare how well semantically relevant sounds are mapped by a discovered representation.

Lightly supervised: Lightly supervised systems consider some other form of available information besides raw speech data. The systems in [15] and [16] use weak top-down constraints in the form of same-type annotations.

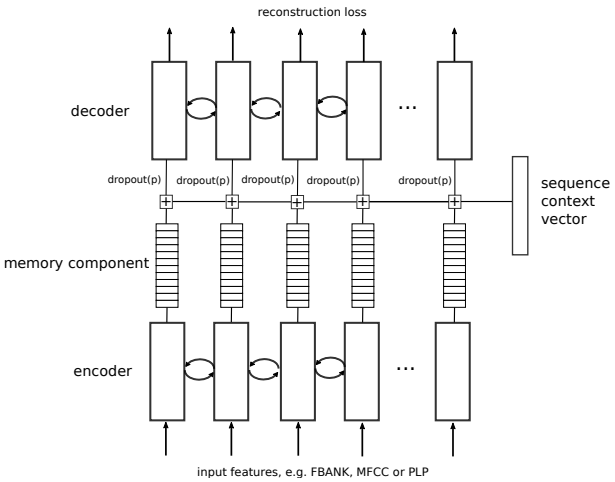

Figure 1: *Sequence encoder / decoder with a memory component as sparsity bottleneck.*

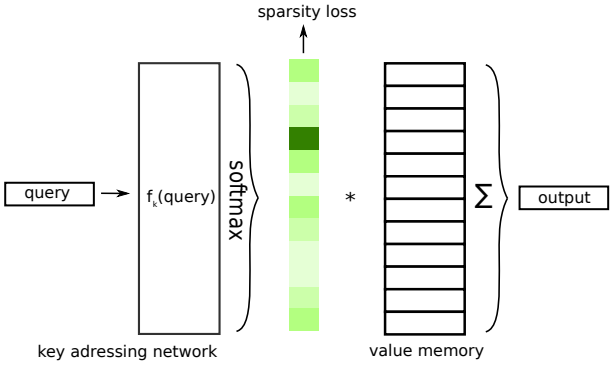

Figure 2: *Memory module consisting of an input query key adressing network and a value memory bank containing fixed-sized embeddings.*

3. Proposed Model

The proposed network in this paper jointly infers a segmentation and a classification on an utterance level with an unsupervised training objective. Although the network outputs a label for each frame individually, in contrast to frame-level (or window-level) autoencoders, the network has access to the temporal structure of a full utterance. Figure 1 illustrates the proposed network. Encoder and decoder are bidirectional long short-term memories (LSTMs) [17, 18]. Between encoder and decoder, we apply dropout at the sequence level, i.e. we randomly omit vectors in the sequence between them while training the network. Metaphorically speaking, this achieves that the encoder is never sure that its outputs get passed on to the decoder and must transmit the most salient information about the current and surrounding frames. The decoder must also learn to predict missing frames and corrects accordingly when new information is available from the encoder. Note that without this, the decoder could simply rely only on the inputs from the encoder at each time step and ignore its state, ultimately ignoring the temporal structure entirely and the network structure would be more similar to a regular autoencoder. We use mean squared error (MSE) for the reconstruction loss.

3.1. Memory Component

Figure 2 illustrates the memory component. It is similar to the one used in [19]. For an input x we use $softmax(Wx + b)$, a standard single-layer network without an activation function for the key addressing. We set the dimensions of W such that the output of Wx has as many elements as there are value embeddings. After the softmax operation, we multiply each of the outputs with a corresponding value memory embedding and sum all vectors. This is the output vector of the memory module and it is a weighted sum over the value memory embeddings.

Note that the memory component that we used can also be understood as a form of key-value separated attention [20, 21, 22], where we compare a query by attending to a key memory (instead of attending to RNN states) and output separate vectors from a value memory. The key memory, in our case, is the matrix W. The memory module is smooth and can be used and trained with back-propagation. As part of a bigger neural network, it can be placed between any layers, the input query dimension can be the same as the output vector dimension.

The goal of the memory network, within the context of the sparse sequence autoencoder, is to sparsify the outputs of encoder network. However, without additional constraints, the output of the softmax layer will not necessarily be sparse and performs a soft-clustering over the output. We use the following constraint described in the next section on the softmax layer of the addressing network to achieve varying degrees of sparseness of this clustering, up to a quasi-hard clustering.

3.2. Enforcing Sparsity

We propose the following sparsity-inducing constraint for the softmax layer, based on L_∞ regularization. Let $\sigma_1, ..., \sigma_n$ be the outputs of the softmax layer in the memory component, then:

$$\text{Softmax-}L_\infty = 1 - \sup_n \sigma_i \tag{1}$$

This regularization loss is zero, iff one of the softmax outputs is one and all other outputs are zero (perfect sparsity, one-hot encoding), because the softmax layer enforces the linear constraint that the sum of all outputs is one and all elements are positive. The regularization term is multiplied by a sparsity weight and is added to the loss of the full network.

3.3. Context Vector

We concatenate a context vector at each time step to the outputs of the encoder and memory component and use this combined vector as input to the decoder. We compare separately trained *Unspeech* context embeddings [23] as static additional input to the network and propose a simple integrated alternative inspired by sequence summary networks [24] that does not require extra training: we sum all encoder states to a single vector with the same size as the encoding states.

3.4. Enforcing Diversity

A degenerate solution to minimize the sparsity constraint is to place all inputs into a single memory component, i.e. clustering all inputs into a single cluster. To discourage this degenerate solution, we also propose a diversity constraint calculated over m time steps of an utterance with n softmax outputs per timestep:

$$\text{Diversity-}L_\infty = \sup_n \frac{\sum_{j=0}^{m} \sigma_{ji}}{m} \tag{2}$$

3.5. Training Procedure

With a random initialization, we found it difficult to train the complete network together with the memory component directly from scratch, even with the diversity constraint. Most of the time, the training will be stuck in a degenerate solution of using either one memory component for all queries or two alternating memory value components for speech and silence.

We propose the following iterative training procedure, based on pre-training a network without the sparsity-inducing memory component first:

1. Train the network without the sparsity layer until the reconstruction loss converges.
2. Run a cluster algorithm, e.g. k-means (n = number of memory banks) on a subset of the randomly select selected bottleneck features obtained by running the encoder of the network on a couple of input sequences.
3. Initialize the value memory bank weights of the memory component to the cluster centers of step (2). Then train the key addressing subnetwork on the cluster labels.
4. Connect the memory sub-module to the network by placing it between the encoder and decoder of the network trained in (1) and continue training the full network with added sparsity loss term to the loss function.

The proposed training procedure above fixes n (the number of units) for simplicity to a manually set value. By swapping the k-means initialization with a different cluster algorithm, e.g. a density based one, n could also be inferred from the data.

4. Implementation

We implemented the model in Tensorflow [25] and make the code publicly available[1]. Encoder and decoder are either 2-layer respectively 4-layer stacked LSTMs with a hidden size of 256. The bottleneck layer has a size of 32. We use ADAM [26] for training the network with a learning rate of 0.001. For initializing the memory subnetwork we use scikit-learn's [27] implementations for k-means++ [28] and DPGMM. The memory values are either initialized with the cluster centers from k-means, or the mean vectors of the DPGMM components. They are initialized from a subset of all training data, we randomly sample 1000 utterances and subsample them further to obtain 2 million vectors for the initial clustering. We use *Unspeech* vectors [23] trained on the same data with a size of 256 dimensions. Feature vectors (MFCC and PLP) are created with Kaldi [29]. We also created our own phone and word alignments on Librispeech's 360 hour clean subset [30] using force-alignment with a strong speaker-independent GMM-HMM trained with Kaldi (7% WER on clean read speech).

5. Evaluation

Visualization of different sparsity values: Figure 3 shows an example input and reconstruction, along with pseudo-posteriors of training runs with different sparsity weights in the sparsity term of the loss function. The pseudo-posteriors are generated by running a forward pass on the encoder and memory component of the network and taking the output after the internal softmax. It also shows that the reconstruction is similar to the input, even though we zero out the connection between decoder and encoder with a probability of 66.6% (sequence dropout).

[1] See https://gitlab.com/milde/sparsespeech

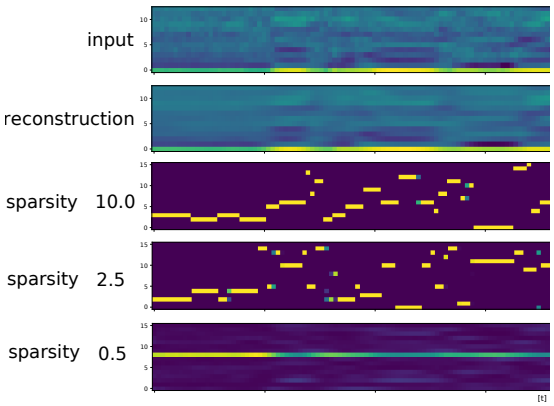

Figure 3: *Training runs with varying sparsity weights (n=16).*

Table 1: *Baseline ABX values on English (Librispeech).*

	ABX within		ABX across	
	word	triphone	word	triphone
MFCC	18.2	30.3	28.1	41.4
PLP	17.8	29.6	27.6	39.7
PLP-DPGMM	7.5	19.0	11.0	26.9

The decoder seems to be able to guess missing feature vectors fairly well. Higher values for the sparsity weight yield sparser representations; at a sparsity weight of 10.0 the representations are mostly one-hot encoded.

ABX evaluation: In an ABX task, we test if stimulus A or B is closer to X. The minimal pair ABX discriminability [13, 14] is an error measure that applies this scheme to speech sounds. We use two different forms of ABX tasks in our evaluation. In a word ABX task, A and B are different words, e.g. A=dog B=cat[1] X=cat[2] where B is closer to X. In a triphone minimal pair (MP) task, e.g. A=beg B=bag[1] X=bag[2], A and B are differentiated only by a center phone. The task includes all triphone phoneme sequences that appear in the data, not only the ones that form words. The ABX error measure discriminates between within- and across-speaker tasks, in the former B and X are from the same speaker and in the latter they are uttered by different speakers. The distance metric is representation agnostic and uses Dynamic Time Warping (DTW) [32] to compare if A or B is closer (smaller distance) to X. For all posteriogram-like outputs, we use the Kullback-Leibler (KL) divergence as local comparison function, cosine distance otherwise.

The ZeroSpeech challenges in 2015 and 2017 used comparatively little speech data, 5 hours respectively 45 hours of English speech. We assume that neural models need more unlabeled data to learn effective representations and base our evaluation on training on the 360 hour subset of the Librispeech corpus [30] for English read speech. While our model can work with any feature representation, we follow the winning systems of the ZeroSpeech 2017 challenge [12] and train on perceptual linear predictive (PLP) features [33].

We use a randomized and unweighted version of the ABX measure (the random seed is fixed between all comparisons) to evaluate word and minimal-pair ABX. We sample 400 pairs of speakers from the corpus and select either all common words or triphone sequences for ABX tasks. We show baseline scores for common feature representations and a PLP-DPGMM in Table 1. Note that our ABX seems to be more difficult than the English subset in Task 1 of the ZeroSpeech 2017 challenge, as

Table 2: *Word and triphone minimal pair ABX error rates of the proposed model on 360 hours of English read speech (Librispeech).*

context vector	stacked layers	memory init	sparsity	diversity	n	training epochs	ABX within speakers word	ABX within speakers triphn-MP	ABX across speakers word	ABX across speakers triphn-MP
Encoder sum	2	k-means++	10.0	0	10	10	7.8	18.4	9.2	22.4
Encoder sum	2	k-means++	10.0	0	16	10	6.9	17.2	8.5	22.0
Encoder sum	2	k-means++	10.0	0	20	10	6.7	16.8	9.2	22.5
Encoder sum	2	k-means++	10.0	0	20	1	8.0	18.4	11.1	25.2
Encoder sum	2	k-means++	10.0	0	42	10	7.8	17.9	12.6	26.5
Encoder sum	2	k-means++	10.0	0	80	10	9.4	19.3	15.2	29.0
Unspeech	2	k-means++	10.0	0	20	10	7.3	17.5	9.9	23.8
Unspeech	2	k-means++	10.0	0	42	10	8.5	18.8	13.4	27.7
Encoder sum	2	k-means++	0	0	16	10	8.8	18.0	16.2	29.5
Encoder sum	2	k-means++	0.5	0	16	10	9.2	18.5	17.1	30.6
Encoder sum	2	k-means++	5.0	0	16	10	6.7	16.4	8.2	20.1
Encoder sum	2	DPGMM	8.0	0	(max 80)	10	8.7	18.3	13.6	26.8
Encoder sum	2	DPGMM	10.0	0	(max 80)	10	8.5	18.1	13.4	26.6
Encoder sum	2	k-means++	10.0	10.0	16	10	6.9	16.9	8.6	21.5
Encoder sum	2	k-means++	2.0	10.0	16	10	5.5	14.5	6.7	18.2
Encoder sum	4	k-means++	2.0	10.0	16	10	5.5	**14.1**	**6.4**	17.5
Encoder sum	4	k-means++	2.0	100.0	16	10	**5.3**	**14.1**	**6.4**	**17.1**

Table 3: *Triphone minimal pair ABX error rates on the ZeroSpeech 2017 test set (English), lower is better.*

	MP-ABX within			MP-ABX across		
Features	1s	10s	120s	1s	10s	120s
(A) MFCC [12]	12.0	12.1	12.1	23.4	23.4	23.4
(B) Heck et al. [11]	6.9	6.2	6.0	10.1	8.7	8.5
(C) Pellegrini et al. [31]	9.8	8.1	8.2	17.6	16.2	16.3
(1) Bottleneck (dense)	9.7	9.7	9.7	23.4	23.4	23.4
(2) Bottleneck 360h (dense)	9.6	9.7	9.7	22.2	22.2	22.2
(3) n=20, KL	13.4	13.3	13.3	22.2	22.3	22.3
(4) n=20, 360h, KL	11.1	10.7	10.6	17.1	16.8	16.7
(5) n=42, 360h, KL	12.2	12.0	12.0	21.7	21.5	21.4
(6) n=16,+div, 360h, KL	9.4	8.9	9.0	14.1	13.3	13.2
(7) + 4-layer LSTM, KL	9.5	9.0	8.9	14.0	12.8	12.4
(B) + VQ argmax	22.6	11.5	11.8	30.1	16.2	16.7
(7) + VQ argmax	11.1	10.5	10.3	15.9	14.7	14.2

the baseline scores are higher on the triphone minimal-pair task. We computed the PLP-DPGMM with $1/10$ of the training data and restricted the maximum number of components to 80, as we found it difficult to scale the DPGMM training to 360 hours.

In Table 2, we compare (sampled) word and triphone-MP ABX error rates on different sparsity and cluster settings on the 360 hour Librispeech set. DPGMM-based memory value initialization (effectively running DPGMM clustering on the encoder states) does not seem to be more effective than the much faster and simpler k-means-based initialization. Using the encoder-sum vector as context vector for the sequence is slightly better than using a fixed and pretrained Unspeech context embedding. With k-means, the best ABX scores across speakers are obtained with k=16, while k=20 yields slightly better within-speaker ABX scores. Training longer than 1 epoch after the initial cluster initialization improves ABX errors by a large margin. Disabling the sparsity constraint by setting it to zero yielded very high ABX error rates, reflecting its important role in our setup. The best ABX scores are obtained by additionally using the diversity constraint. Using 4-layer LSTM encoders and decoders is slightly more effective than 2 layers.

In Table 3 we use the evaluation scripts and English test data of the ZeroSpeech 2017 challenge. (1) and (3) are trained using the 45 hours train set of the challenge. The bottleneck features are dense and from the pretrained network without the memory component; they seem to provide improvements to within-speaker ABX scores over the baseline (A). Pre-training on more data does not seem to improve this significantly (2). System (3), trained with a sparsity weight of 10.0, outputs a quasi-symbolic representation. While this representation results in higher within-ABX error rates than MFCCs (A), it performs better across speakers. Systems (4) and (5), trained on an order of magnitude more data with 360 hours but the same parameters otherwise, drastically reduce within and across ABX error rates over System (3). Reference system (B) provides lower ABX error rates than our system (7), but the generated posteriors are 1144-dimensional and they are not as sparse as the ones generated by our system; for 10s the mean max. value is 0.386 vs. 0.986. In the last two rows, we compare (7) and (B) under the constraint that the representation must be completely sparse (symbolic) at each time step, i.e. we take the argmax and use indices in the comparison. In this case, our system provides significantly better ABX scores than the reference system (B).

6. Conclusion

We presented a novel neural approach to unsupervised acoustic unit discovery, based on a sequence autoencoder with a sparsity inducing memory component. The proposed sparsity constraint restricts the model to develop a quasi-symbolic representation. We propose an iterative training procedure, where the network is pre-trained without the sparsity memory component first. The architecture can be trained with standard backpropagation. This makes it easily possible to scale the training to larger training sizes. We were able to train the model on up to 360 hours of speech in two days on a single GPU. Extending this to much larger training sizes should be easily possible. Our generated representations are very compact with few sub-word units and are mostly one-hot encoded symbolic-like representations, with better ABX discriminability than MFCC.

We can also confirm that a PLP-DPGMM model is a strong baseline for the ABX task, even though the training objective operates on the frame level. It often produces many hundreds of sub word units when not restricted, which may be impractical in certain tasks (see [34]). Also, we found it difficult to scale DPGMM to larger training sizes. Scaling our model to 360 hours of data provides representations with better ABX discriminability than PLP-DPGMM with speaker-independent transformations [11] under the constraint that the output representation needs to be completely sparse and symbolic.

7. References

[1] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, Merano/Meran, Italy, 2009, pp. 398–403.

[2] M.-H. Siu, H. Gish, S. Lowe, and A. Chan, "Unsupervised audio patterns discovery using HMM-based self-organized units," in *Proc. Interspeech*, Florence, Italy, 2011, pp. 2333–2336.

[3] L. Badino, A. Mereta, and L. Rosasco, "Discovering discrete subword units with binarized autoencoders and hidden-markov-model encoders," in *Proc. Interspeech*, Dresden, Germany, 2015, pp. 3174–3178.

[4] H. Kamper, A. Jansen, S. King, and S. Goldwater, "Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings," in *Proc. Spoken Language Technology Workshop (SLT)*, South Lake Tahoe, NV, USA, 2014, pp. 100–105.

[5] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources," in *Proc. Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, 2006, pp. 949–952.

[6] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7634–7638.

[7] D. Renshaw, H. Kamper, A. Jansen, and S. Goldwater, "A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge," in *Proc. Interspeech*, Dresden, Germany, 2015, pp. 3199–3203.

[8] C.-Y. Lee and J. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proc. ACL*, Jeju Island, South Korea, 2012, pp. 40–49.

[9] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Parallel inference of Dirichlet process Gaussian mixture models for unsupervised acoustic modeling: A feasibility study," in *Proc. Interspeech*, Dresden, Germany, 2015, pp. 3189–3193.

[10] M. Versteegh, R. Thiolliere, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The Zero Resource Speech Challenge 2015," in *Proc. Interspeech*, Dresden, Germany, 2015, pp. 3169–3173.

[11] M. Heck, S. Sakti, and S. Nakamura, "Feature optimized DPGMM clustering for unsupervised subword modeling: A contribution to ZeroSpeech 2017," in *Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 740–746.

[12] E. Dunbar, X. N. Cao, J. Benjumea, J. Karadayi, M. Bernard, L. Besacier, X. Anguera, and E. Dupoux, "The Zero Resource Speech Challenge 2017," in *Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 323–330.

[13] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux, "Evaluating speech features with the minimal-pair ABX task: Analysis of the classical MFC/PLP pipeline," in *Proc. Interspeech*, Lyon, France, 2013, pp. 1781–1785.

[14] T. Schatz, "ABX-discriminability measures and applications," Ph.D. dissertation, Université Paris 6 (UPMC), 2016.

[15] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Queensland, Australia, 2015, pp. 5818–5822.

[16] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Proc. Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 4950–4954.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[19] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2015, pp. 2440–2448.

[20] M. Daniluk, T. Rocktäschel, J. Welbl, and S. Riedel, "Frustratingly short attention spans in neural language modeling," in *Proc. International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.

[22] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, 2015, pp. 1412–1421.

[23] B. Milde and C. Biemann, "Unspeech: Unsupervised speech context embeddings," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 2693–2697.

[24] K. Veselỳ, S. Watanabe, K. Žmolíková, M. Karafiát, L. Burget, and J. H. Černockỳ, "Sequence summarizing neural network for speaker adaptation," in *Proc. Acoustics, Speech and Signal Processing (ICASSP)*, Lujiazui, Shanghai, China, 2016, pp. 5315–5319.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. USENIX Symposium on Operating Systems Design and Implementation*, Savannah, GA, USA, 2016, pp. 265–283.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, vol. abs/1412.6, 2014.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[28] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. ACM-SIAM symposium on Discrete algorithms*, New Orleans, LA, USA, 2007, pp. 1027–1035.

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. Automatic Speech Recognition and Understanding Workshop (ASRU)*, Atlanta, GA, USA, 2011.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*, Brisbane, Queensland, Australia, 2015, pp. 5206–5210.

[31] T. Pellegrini, C. Manenti, and J. Pinquier, "Technical report: The IRIT-UPS system at ZeroSpeech 2017 track1: Unsupervised subword modeling," 2017.

[32] T. K. Vintsyuk, "Speech discrimination by dynamic programming," *Cybernetics and Systems Analysis*, vol. 4, no. 1, pp. 52–57, 1968.

[33] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[34] B. Wu, S. Sakti, J. Zhang, and S. Nakamura, "Optimizing DPGMM clustering in zero-resource setting based on functional load," in *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, Gurugram, India, 2018, pp. 1–5.