A Comparison of Neural Network Methods for Unsupervised Representation Learning on the Zero Resource Speech Challenge

Daniel Renshaw[*], *Herman Kamper*[†], *Aren Jansen*[‡], *Sharon Goldwater*[§]

[*,†,§]ILCC and [†]CSTR, School of Informatics, University of Edinburgh, UK
[‡]HLTCOE and CLSP, Johns Hopkins University, USA

daniel.renshaw@ed.ac.uk, h.kamper@sms.ed.ac.uk, aren@jhu.edu, sgwater@inf.ed.ac.uk

Abstract

The success of supervised deep neural networks (DNNs) in speech recognition cannot be transferred to zero-resource languages where the requisite transcriptions are unavailable. We investigate unsupervised neural network based methods for learning frame-level representations. Good frame representations eliminate differences in accent, gender, channel characteristics, and other factors to model subword units for within- and across-speaker phonetic discrimination. We enhance the correspondence autoencoder (cAE) and show that it can transform Mel Frequency Cepstral Coefficients (MFCCs) into more effective frame representations given a set of matched word pairs from an unsupervised term discovery (UTD) system. The cAE combines the feature extraction power of autoencoders with the weak supervision signal from UTD pairs to better approximate the extrinsic task's objective during training. We use the Zero Resource Speech Challenge's minimal triphone pair ABX discrimination task to evaluate our methods. Optimizing a cAE architecture on English and applying it to a zero-resource language, Xitsonga, we obtain a relative error rate reduction of 35% compared to the original MFCCs. We also show that Xitsonga frame representations extracted from the bottleneck layer of a supervised DNN trained on English can be further enhanced by the cAE, yielding a relative error rate reduction of 39%.

Index Terms: unsupervised speech processing, representation learning, zero-resources, neural networks, autoencoders

1. Introduction

Automatic speech recognition systems are typically trained using tens or hundreds of hours of hand-transcribed speech data and often still have difficulty dealing with differences in accent, gender, channel characteristics, and other factors. Yet months-old human infants begin to solve the basic problems of identifying phones and words with no comparable supervision. Recent work on *zero-resource speech technology* asks: how can we build artificial systems that might approach the unsupervised learning abilities of human infants? Solving this problem would provide more universally available speech technology in under-resourced languages, and could lead to novel methods that also improve supervised speech recognition.

To promote work in this area, the INTERSPEECH 2015 Zero Resource Speech Challenge (ZRSC) defines two shared tasks. We tackle Track 1, subword modeling, by using several neural network based methods to learn frame-level representations that yield better phonetic discriminability than standard Mel Frequency Cepstral Coefficients (MFCCs) [1]. We evaluate our representations using the ZRSC's Track 1 minimal triphone pair *ABX* discrimination task.

We enhance the correspondence autoencoder (cAE) method of Kamper et al. [2], which learns a nonlinear mapping from MFCCs to latent distributed feature representations. Kamper et al. trained their system on data from the Switchboard corpus and evaluated it using the *same-different* discriminability task [3], showing that the learned representations performed substantially better than the original MFCCs, and also better than representations learned by a standard autoencoder (AE) [4]. Here, we show that Kamper et al.'s model, with no additional tuning, generalizes well to other data sets, languages, and tasks by evaluating it using the *ABX* task with the two ZRSC datasets: Buckeye [5] (English, but with different channel characteristics than Switchboard) and the NCHLT Xitsonga[1] Speech corpus [6].

While the cAE is a weakly supervised model, we train it with correspondence pairs sourced from an unsupervised term discovery (UTD) system [7] making the approach unsupervised as a whole. Alternatives to the weakly supervised regularization implicit in the cAE were not considered in [2]; we show here that a standard unsupervised form of regularization, denoising autoencoders [8], learns better representations than AEs, but still not as good as cAEs. We also introduce an improved cAE architecture and training method that reduces the number of hyperparameters to be tuned, and show that narrow architectures work better, with reduced error rates on a zero-resource language after tuning on English. Unlike similar previous work [9, 10, 11] our cAE-based systems are fully unsupervised, train on individual frames without context, and use a loss function in the input vector space instead of the representation vector space.

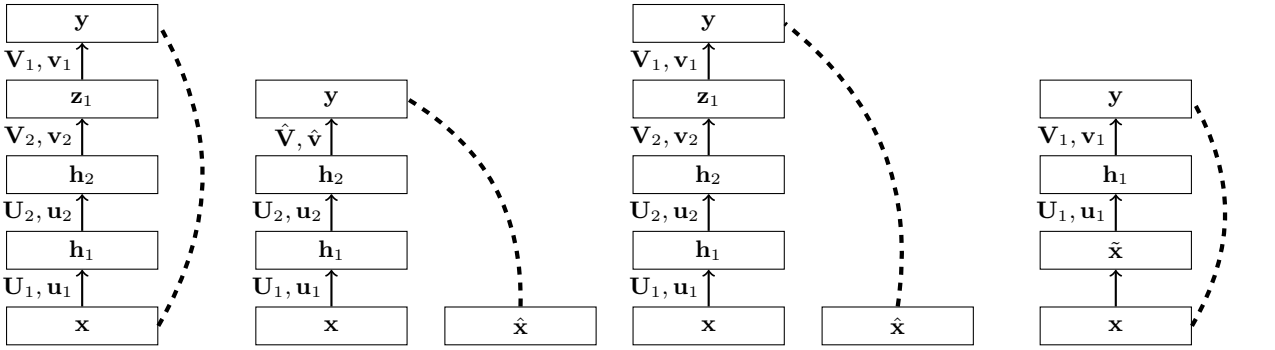
Finally, we explore whether *ABX* performance can be improved on a zero-resource language by using representations trained on large amounts of supervised data in a different language. We use a deep neural network (DNN) trained on a large amount of English data to extract bottleneck features (BNFs) for the Xitsonga test data. We find that these cross-language supervised representations perform comparably with the cAE representations trained with in-language data and that further improvements can be achieved by combining the two approaches.

2. Models

2.1. Autoencoder

A single-layer *autoencoder* (AE) [4] has two components. The *encoder* projects an input, e.g. an MFCC vector, $\mathbf{x} \in \mathbb{R}^{D_0}$ into *hidden representation* $\mathbf{h}_1 \in \mathbb{R}^{D_1}$. The *decoder* projects the hidden representation back into the original vector space $\mathbf{y} \in \mathbb{R}^{D_0}$. We treat $\mathbf{y}$ as a *reconstruction* of $\mathbf{x}$ and train the network to minimize the reconstruction error $\mathcal{L}^{AE} = \sum_{i=1}^{D_0} (y_i - x_i)^2$.

[1] Xitsonga is a southern African Bantu language.



(a) Deep autoencoder. (b) Correspondence autoencoder with shallow decoder. (c) Correspondence autoencoder with deep decoder. (d) Denoising autoencoder. $\tilde{\mathbf{x}}$ is an artificially noisy version of $\mathbf{x}$.

Figure 1: Autoencoder model types. Directed arrows depict feedforward relations. Dashed lines indicate loss functions. Parameters in use at each layer are given to the left of each directed arrow. Every layer uses a nonlinear activation function with the exception of the top decoder, which is linear. In Figure 1b and Figure 1c, $\mathbf{x}$ and $\hat{\mathbf{x}}$ are examples drawn from the same class.

The encoder is implemented as a conventional feedforward neural network layer, $\mathbf{h}_1 = \sigma_1(\mathbf{U}_1\mathbf{x} + \mathbf{u}_1)$, with weights $\mathbf{U}_1 \in \mathbb{R}^{D_1 \times D_0}$, biases $\mathbf{u}_1 \in \mathbb{R}^{D_1}$, and nonlinear activation function σ_1 (we use the hyperbolic tangent). The decoder has the same form as the encoder, $\mathbf{y} = \psi_1(\mathbf{V}_1\mathbf{h}_1 + \mathbf{v}_1)$, with additional weights $\mathbf{V}_1 \in \mathbb{R}^{D_0 \times D_1}$ and biases $\mathbf{v}_1 \in \mathbb{R}^{D_0}$. To reconstruct unbounded data, such as MFCCs, the decoder's activation function ψ_1 must have an unbounded range; we use the identity function, $\psi_1(\mathbf{a}) = \mathbf{a}$, yielding a linear decoder.

Deep narrow AEs typically achieve lower reconstruction error than shallow wide AEs with the same number of parameters. As shown in Figure 1a, each of a deep AE's L encoders project the output of the previous encoder into a new hidden representation, i.e. $\mathbf{h}_i = \sigma_i(\mathbf{U}_i\mathbf{h}_{i-1} + \mathbf{u}_i) \in D_i$ for all $i \in [1, L]$, with $\mathbf{h}_0 = \mathbf{x}$. Each of the L decoders reconstruct their respective hidden representations in turn, finally reconstructing the input, i.e. $\mathbf{z}_{i-1} = \psi_i(\mathbf{V}_i\mathbf{z}_i + \mathbf{v}_i) \in D_{i-1}$ for all $i \in [1, L]$, with $\mathbf{z}_L = \mathbf{h}_L$ and $\mathbf{y} = \mathbf{z}_0$. We tie weights and use nonlinear activation functions in all internal decoders, i.e. $\sigma_i = \psi_i$ and $\mathbf{V}_i = (\mathbf{U}_i)^T$ for all $i \in [2, L]$. Our autoencoders use a consistent hidden layer size, i.e. $D_i = D_{i-1}$ for all $i \in [1, L]$.

Training all layers in a deep AE concurrently often yields poor results due to the vanishing gradient problem [12, 13]. We use the standard mitigation of pre-training the deep AE layerwise, then fine-tuning the entire network [14].

2.2. Correspondence autoencoder

An AE is trained with an unsupervised objective. We can learn better quality representations if we use an objective that better approximates the extrinsic task: learning features that discriminate between different subword units. To this end, Kamper et al. [2] introduced a weakly supervised AE variant: the *correspondence autoencoder* (cAE). Instead of reconstructing its input, a cAE is trained to 'reconstruct' an unseen example that is known to be similar to the input. Four steps go into training a cAE for unsupervised speech representation learning: (1) unsupervised discovery of word pairs, (2) align word-pair frames, (3) pre-train AE layerwise, (4) fine-tune whole network as a cAE.

We use a UTD system [7] to identify pairs of segments in the corpus that are likely to be instances of the same word (uttered by the same or different speakers). We obtain a frame-level alignment for each segment pair using dynamic time warping (DTW) with cosine distance, yielding a set of frame pairs $\{(\mathbf{x}, \hat{\mathbf{x}})\}$. A deep AE is trained layerwise on the entire corpus to initialize the network's parameters prior to cAE fine-tuning. The cAE is then trained to minimize the reconstruction error of $\mathbf{y}$ relative to $\hat{\mathbf{x}}$ for each frame pair $(\mathbf{x}, \hat{\mathbf{x}})$, yielding the cAE loss $\mathcal{L}^{cAE} = \sum_{i=1}^{D_0}(y_i - \hat{x}_i)^2$. We obtain better results by using every pair twice, once where the first item is input to the network and a second time where the second item is input to the network.

In Kamper et al.'s previous work [2] the layerwise pre-trained decoders were discarded and a single, randomly initialized, decoder trained during cAE fine-tuning, as in Figure 1b. This approach has the advantage that the total number of layers is reduced, mitigating the vanishing gradient problem, but a single linear decoder is unable to undo the work of many nonlinear encoders forcing some of the top encoder layers to be implicitly retrained as decoders during fine-tuning. The layer to use for representation must then be determined using a held-out validation set. In this work we use a deep nonlinear decoder during cAE fine-tuning, as in Figure 1c, allowing the top encoding layer to always be used as the input's representation.

2.3. Denoising autoencoder

Although Kamper et al. showed that cAEs perform better than AEs on the *same-different* task, AEs are not a strong baseline for representation learning. Denoising autoencoders (dAEs) usually perform better because they implicitly regularize the parameters avoiding degenerate transformations, such as the identity function, being learned [15]. Regularization is especially important when training overcomplete AE architectures, i.e. where $D_i \geq D_0$ for all $i \in [1, L]$, as was done by Kamper et al.

A dAE is trained to reconstruct the clean versions of artificially noisy inputs. Different types of noise may be applied. In our case the input features, once normalized to zero mean and unit variance, are approximately Gaussian distributed so additive zero mean Gaussian noise is appropriate. A dAE, such as that shown in Figure 1d, is identical to a conventional AE except the input $\tilde{\mathbf{x}} = \mathbf{x} + \mathcal{N}(\mathbf{0}, \gamma\mathbf{I})$ is a noisy version of $\mathbf{x}$. γ is a hyperparameter defining the standard deviation of the noise.

We can view the cAE as a version of a dAE where, instead of artificial noise, the network is presented with input pairs that differ only in nonlinguistic sources of variation, e.g. speaker or channel. Denoising the true sources of extrinsic variability is a more optimal method than introducing artificial sources, though we are limited to what can be discovered with the UTD system.

3. Experiments

3.1. Data

We use two datasets. The first is a 5 hour portion of the Buckeye corpus [5] distributed as part of the ZRSC. The second is a 2.5 hour portion of the NCHLT Xitsonga Speech corpus [6] consisting of 16 kHz, close-talking microphone, prompted speech.

Using HTK [16], we extract MFCCs using 25 ms windows with 10 ms step size, which are augmented with first and second order derivatives to yield 39-dimensional feature vectors. The MFCCs falling entirely within the speaker segments of interest–the ZRSC's evaluation intervals–are extracted and cepstral mean and variance normalization is applied to those segments per source file. All of the resulting frames are used during pre-training of our networks.

In addition to raw acoustic features as input to the various learning algorithms, we also evaluate the utility of data-driven features that exploit out-of-domain and/or out-of-language supervision. Specifically, we extract BNFs using the Kaldi speech recognition toolkit [17]. Our DNN architecture takes a 9-frame context window of MFCCs as input to 5 hidden layers of 5,000 units (2-norm maxout nonlinearity) followed by a linear bottleneck layer of 60 units (see [18] for details). The softmax output layer consists of 7600 clustered context-dependent HMM state targets. The DNN is trained using the Switchboard and Fisher English corpora, which amount to approximately 1,500 hours of English conversational telephone speech drawn from over 12,000 speakers. The resulting network thus encodes a detailed knowledge of the speaker-independent acoustic-phonetic structure of English, which we expect to produce good BNFs when applied to Buckeye data, despite the channel mismatch. We can also apply the network to Xitsonga data to produce Xitsonga BNFs. Here, any demonstrated improvement over the raw acoustic features would be derived from cross-lingual generalization of the encoded English knowledge.

Correspondence pairs for the Buckeye and Xitsonga corpora are extracted by the UTD system described in [7]. We use the graph clustering method of [19] to group individual discovered repetitions into term clusters from which we can derive more extensive transitive matches. In this way, we recover 11,041 token pairs for the ZRSC Buckeye portion (57% across-speaker) and 6,982 token pairs for Xitsonga (61% across-speaker).

3.2. Training

We have four sources of training data: pairing each language (English and Xitsonga) with each input encoding (MFCC and BNF). In the spirit of using zero-resources, we demonstrate the effect of applying unsupervised models to domains/languages that differ from those used to optimize the architecture.

Our neural networks, implemented in Theano [20, 21], are trained via minibatch stochastic gradient descent backpropagation [22]. Weights and biases are random and zero initialized respectively. The training data is shuffled prior to each epoch of training. MFCC-based models include delta and double-delta features unless otherwise stated.

The "original" model architecture is identical to Kamper et al.'s 9×100-layer model [2] (i.e. 9 encoder layers each of size 100) and is optimized for English Switchboard MFCCs on the *same-different* task and then trained with the data from one of the four sources. We initially layerwise AE pre-train over 30 epochs per layer at a learning rate of 0.00025 then cAE fine-tune over 120 epochs at a learning rate of 0.008. We use minibatches of size 256. A single randomly initialized linear decoder is used during fine-tuning so we must select a representation layer; we report results from using the Switchboard optimal layer, the 6th, and the Buckeye optimal layer, the 9th (found by testing each layer in the network).

The "optimal" model architecture is optimized for the ZRSC Buckeye portion MFCCs on the *ABX* task and then trained with the data from one of the remaining three sources. The optimal network structure, 5 layers each of size 13, was found by grid searching over the number of layers (1, 3, 5, 9) and layer sizes (13, 39, 100). Unlike the "original" architecture, the "optimal" architecture uses tied weights and a deep nonlinear decoder to avoid the layer selection problem. We also use a different training regime for the "optimal" architecture which was found to improve the results. We initially layerwise AE pre-train over 4 epochs per layer at a learning rate of 0.1 then cAE fine-tune over 320 epochs at a learning rate of 0.1. We use AdaGrad [23], minibatches of size 2048, and the correspondence pairs are presented in both directions. The use of AdaGrad allows the learning rate to be set to a single large value eliminating much of the advantage/cost of optimizing this hyperparameter.

Informal experience from our past uses of the cAE suggests the optimal ratio between input size and hidden layer size is similar across datasets and models. This ratio is $\frac{100}{39}$ and $\frac{13}{39}$, respectively, for the "original" and "optimal" approaches using 39-dimensional MFCC inputs, so we use hidden layer widths of 154 and 20 for models taking 60-dimensional BNF inputs.

The AE-only and dAE-only models are trained using the same training regime as the "optimal" models and their architectures are optimized via grid-search over the same layer widths and counts using the ZRSC Buckeye portion MFCCs. The optimal sizes are 1×13 and 1×200 for the AE and dAE respectively. The dAE was trained with $\gamma = 0.2$.

3.3. Evaluation

Following the ZRSC's protocol, we evaluate our frame representations using an *ABX* task [24, 25] which measures the discriminability of frame representations by asking whether triphone x is most like triphone a or triphone b, where a and x are distinct examples of the same triphone sequence and b is a triphone sequence differing from a and x in only the middle phone. We consider two variants: in the *within-speaker* case a, b, and x belong to the same speaker and in the *across-speaker* case a and b belong to one speaker and x belongs to a different speaker.

Triphones are compared by aligning their frame representations using DTW with cosine distance. If the minimum alignment cost between a and x is greater than that between b and x then the model has made an error. The error rate is the mean over all possible (a, b, x) triples in the test set.

3.4. Results

Our results are alphabetically labeled and presented in Table 1a (English) and Table 1b (Xitsonga); we focus our discussion on the more challenging across-speaker case. Comparisons are made to the ZRSC official baselines (MFCCs, `a` and `l`) and supervised toplines (Kaldi posteriorgrams with HMM-GMM, `b` and `m`) [26]. An alternate baseline, using 39-dimensional MFCCs enriched with delta and double delta features, performed similarly to the official plain 13-dimensional MFCC baselines.

AE/dAE: As in previous work, we find that plain AEs (`c` and `n`) barely outperform MFCCs (`a` and `l`). dAEs (`d` and `o`) provide a bigger benefit, supporting previous work showing the importance of regularization in unsupervised representation learning, e.g. [15, 27]. Nevertheless, even dAEs do not match the

English models	Within	Across
a Official baseline (13-dim MFCCs)	15.6	28.1
b Official topline (HMM-GMM)	12.1	16.0
c Optimal AE	16.9	28.6
d Optimal dAE	15.8	25.3
e Original cAE (Switchboard layer: 6th)	15.8	24.7
f Original cAE (Buckeye layer: 9th)	15.1	23.2
g Optimal cAE	**13.5**	**21.1**
h BNFs from English DNN	**12.8**	**18.1**
i Original cAE (Switchboard layer: 6th)	14.1	19.2
j Original cAE (Buckeye layer: 8th)	13.7	18.8
k Optimal cAE	14.0	19.3

(a) English results. Bold indicates per-section best results.

Xitsonga models	Within	Across
l Official baseline (13-dim MFCCs)	19.1	33.8
m Official topline (HMM-GMM)	3.5	4.5
n Buckeye optimized AE	17.4	29.5
o Buckeye optimized dAE	15.8	25.9
p Original cAE (Switchboard layer: 6th)	13.4	22.0
q Original cAE (Buckeye layer: 9th)	12.1	19.6
r Buckeye optimized cAE	**11.9**	**19.3**
s Xitsonga optimized cAE	11.6	18.5
t BNFs from English optimized DNN	14.4	19.3
u Original cAE (Switchboard layer: 6th)	14.1	19.0
v Original cAE (Buckeye layer: 8th)	13.1	17.8
w Buckeye optimized cAE	**13.0**	**18.2**

(b) Xitsonga results. Bold indicates best zero-resource results (architectures optimized on English data).

Table 1: Minimal triphone pairs *ABX* within-/across- speaker error rates. Top sections: official baseline (unsupervised) and topline (supervised). Middle sections: models of MFCCs including delta and double-delta features unless otherwise stated. Bottom sections: models of bottleneck features (BNFs) extracted from English trained DNN.

performance of any cAEs, supporting Kamper et al's claim that guiding the representation learning using UTD pairs provides a major benefit over standard unsupervised methods. With further optimization (e.g. different types and levels of noise) better dAE results may be obtained which could be helpful in situations where correspondence pairs are unavailable.

"Original" cAE: Despite having its architecture optimized in a different domain, the "original" cAE (e) reduces Buckeye error rates compared to MFCCs (a) by 17% relative. When we use oracle layer selection (f), the relative error rate reduction increases to 22%. These results are in line with previous work showing layer selection is important for getting the best results from a cAE that uses a single linear decoder during fine-tuning. The same architecture trained on Xitsonga has the same pattern of results but with larger relative error rate reductions of 26% and 34% (p and q) compared to the baseline MFCCs (l). These latter cross-language results are encouraging evidence that the cAE could be applied productively to other zero-resource settings without fearing the architecture is especially sub-optimal.

"Optimal" cAE: Optimizing the cAE architecture on the Buckeye data increases the error rate improvement from 17% (e) to 29% (g) relative to the MFCC baseline (a). Clearly, the channel and task differences between Switchboard/*same-different* and Buckeye/*ABX* are significant. In the zero-resource case we find that the English improvements transfer to Xitsonga without any further optimization; the Buckeye-"optimal" cAE (r) reduces the Xitsonga error rate from the MFCC baseline by 35% relative. Optimizing the architecture on Xitsonga (s; a 9×13 architecture was best here) yields an improvement but this is not a zero-resource result. The greater reductions achieved by the Buckeye/*ABX*-optimized architecture compared to the Switchboard/*same-different*-optimized architecture may be due to changes in architecture, to changes in training regime, or to optimizing for a different task.

DNN BNFs: Unsurprisingly, for English, the supervised BNFs (h) perform substantially better than the representations found by the unsupervised cAE (g). Furthermore, the cAE is unable to improve the English BNFs (i, j, k) suggesting the two training objectives are not complementary in the same-language setting. Pleasingly, applying the English DNN to Xitsonga (t) produces representations that perform just as well as the Buckeye-"optimal" cAE (r) representations. Our best zero-resource result is obtained by applying the "optimal" cAE to the Xitsonga BNFs (u, v, w) yielding representations of better quality than either approach achieves independently. By optimizing the Xitsonga cAE architecture a little we found that a narrower cAE network, 100 instead of 154, produced better results, reducing the error rate to 16.6% when using an "original" architecture, but this is not a zero-resource result.

More training data: Training the optimal cAE architecture on the entire Buckeye corpus, with more UTD pairs, did not yield distinctly different results from (g).

4. Conclusions and future work

We have presented a selection of approaches for learning frame representations using unsupervised methods in zero-resource settings. Using a minimal triphone pair ABX discrimination task we showed that correspondence autoencoders (cAEs) outperform denoising autoencoders which outperform plain autoencoders. Although the cAE is a weakly supervised model, we obtain the correspondence data from a fully unsupervised term discovery system making our approach, taken as a whole, fully unsupervised. Compared to the original MFCCs, the cAE architecture optimized on Switchboard English reduces ABX error rates by 17% (relative) on Buckeye English and 26% on Xitsonga. Optimizing on Buckeye English instead yields a Xitsonga error rate reduction of 35%. These results demonstrate that our unsupervised system can be optimized in a high-resource setting and then applied productively in a zero-resource setting.

We also found that applying the unsupervised cAE system to Xitsonga bottleneck features obtained from a supervised DNN trained on English yielded better results on Xitsonga than either system alone: an overall relative error rate reduction of 39% over MFCCs. This result suggests a promising future line of work combining supervised training in a high-resource language with unsupervised language adaptation to the zero-resource language.

Acknowledgments DR is supported by a Google European Doctoral Fellowship and SG by a James S McDonnell Foundation Scholar Award. The authors thank Daniel Garcia-Romero of JHU for providing the trained DNN for BNF extraction, and Krzysztof Jerzy Geras of UoE for useful discussions about AEs.

5. References

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[2] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015.

[3] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *INTERSPEECH 2011: 12th Annual Conference of the International Speech Communication Association*, 2011, pp. 821–824.

[4] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, vol. 59, no. 4-5, pp. 291–294, 1988.

[5] M. A. Pitt, K. Johnson, E. Hume, S. Kiesling, and W. Raymond, "The Buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability," *Speech Communication*, vol. 45, no. 1, pp. 89–95, 2005.

[6] N. J. De Vries, M. H. Davel, J. Badenhorst, W. D. Basson, F. De Wet, E. Barnard, and A. De Waal, "A smartphone-based ASR data collection tool for under-resourced languages," *Speech Communication*, vol. 56, pp. 119–131, 2014.

[7] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 401–406.

[8] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*. ACM, 2008, pp. 1096–1103.

[9] G. Synnaeve, T. Schatz, and E. Dupoux, "Phonetics embedding learning with side information," in *Spoken Language Technology (SLT), 2014 IEEE Workshop on*. IEEE, 2014.

[10] G. Synnaeve and E. Dupoux, "Weakly supervised multi-embeddings learning of acoustic models," *arXiv preprint arXiv:1412.6645*, 2014.

[11] X. Zheng, Z. Wu, H. Meng, and L. Cai, "Contrastive auto-encoder for phoneme recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2529–2533.

[12] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," *Master's thesis, Institut für Informatik, Technische Universitat, Munchen*, 1991.

[13] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[15] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[16] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, "The HTK book (for HTK version 3.4)," 2009.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," 2011.

[18] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 215–219.

[19] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, "NLP on spoken documents without ASR," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 460–470.

[20] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[21] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, oral Presentation.

[22] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, no. 8, 1991.

[23] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[24] 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 *INTERSPEECH 2013: 14th Annual Conference of the International Speech Communication Association*, 2013, pp. 1–5.

[25] T. Schatz, V. Peddinti, X.-N. Cao, F. Bach, H. Hermansky, and E. Dupoux, "Evaluating speech features with the Minimal-Pair ABX task (II): Resistance to noise," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[26] M. Versteegh, R. Thiolliere, T. Schatz, X.-N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," 2015, submitted to *INTERSPEECH 2015: 15th Annual Conference of the International Speech Communication Association*.

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