Triphone State-tying via Deep Canonical Correlation Analysis

Weiran Wang, Hao Tang, Karen Livescu

Toyota Technological Institute at Chicago, USA
{weiranwang,haotang,klivescu}@ttic.edu

Abstract

Context-dependent phone models are used in modern speech recognition systems to account for co-articulation effects. Due to the vast number of possible context-dependent phones, state-tying is typically used to reduce the number of target classes for acoustic modeling. We propose a novel approach for state-tying which is completely data dependent and requires no domain knowledge. Our method first learns low-dimensional embeddings of context-dependent phones using deep canonical correlation analysis. The learned embeddings capture similarity between triphones and are highly predictable from the acoustics. We then cluster the embeddings and use cluster IDs as tied states. The bottleneck features of a DNN predicting the tied states achieve competitive recognition accuracy on TIMIT.
Index Terms: context-dependent phone embeddings, deep canonical correlation analysis, state-tying

1. Introduction

In most typical speech recognition systems, the states used in recognition correspond to clusters of tied context-dependent sub-phonetic triphone states. Each such clustered state is associated with a target of a deep network classifer (DNN, CNN, RNN, etc.) or a Gaussian mixture model (GMM) observation density. The state clustering is done in order to handle the very large number of possible triphone states, some of which occur too infrequently for robust training.

Decision tree state tying [1] is one approach for state clustering that has stood the test of time, and is still currently being used in popular speech recognition systems [2]. Every sub-phonetic state is assigned a cluster according to the trained decision tree, and every state within a cluster shares the same DNN target or GMM. The tree is typically trained by maximizing the likelihood of the data under a model where each leaf's density is Gaussian, subject to a stopping condition on likelihood increase and/or cluster occupancy. At each step of decision tree construction, a tree leaf is split by asking a question about a property of the leaf, such as the identity of the left or right context phones, their phonetic features, or their membership in ad hoc phone subsets. The set of questions might themselves be constructed by a clustering procedure, as in the Kaldi toolkit [2].

In this paper, we revisit the state tying problem from a new perspective. The idea relies on learning low-dimensional embeddings of the triphone labels that capture the semantics of the label space, and then cluster those embeddings with a simple vanilla clustering such as K-means. The algorithm we use to learn the embedding is deep canonical correlation analysis [3, 4], which projects acoustic inputs and triphone labels into a common subspace using deep neural networks (DNNs), such that the projections are maximally correlated, i.e. are predictive from each other. As a result, the triphones that are acoustically similar (according to the learned deep neural network) are mapped to similar locations in the embedding space. We then simply perform K-means clustering in the label embedding space and use each resulting cluster of triphone labels as a tied state. Our approach is completely data dependent and requires no domain knowledge. We demonstrate our state tying approach on the TIMIT dataset [5], and find that bottleneck features trained to predict our tied states achieve comparable phone error rates to a standard DNN/HMM system, while using a much smaller number of tied triphone states.

2. Label embedding via deep CCA

We first review deep canonical correlation analysis (deep CCA), which has been one of the most successful methods for unsupervised learning of representations (features) from multi-view data for a variety of tasks [3, 6, 7, 8].

In the multi-view representation learning setting, we have access to different types of measurements of the same underlying signal during representation learning. If the views provide complementary information, learning compact representations from the multi-view data can capture useful information provided about each view by the other view ("soft supervision") or remove uncorrelated noise from the original input measurements. This approach has been applied to multi-view data such as audio+articulation [9, 4], audio+video [10, 11], images+text [12, 13, 8], and multilingual text [14, 15, 16, 17, 7].

Unlike previous work that uses deep CCA for unsupervised feature learning [4], here we work in a supervised setting and learn low-dimensional representation of the *labels*. That is, we consider the two views to be the acoustic inputs and the triphone state labels. Formally, the training data consist of pairs of observations $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^{D_x}$ and $\mathbf{y}_i \in \mathbb{R}^{D_y}$ represent acoustic features computed over one frame and the properly encoded triphone label for that frame. We also denote $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$.

In deep CCA, we use two deep neural networks (DNNs) $\mathbf{f} : \mathbb{R}^{D_x} \to \mathbb{R}^{d_x}$ and $\mathbf{g} : \mathbb{R}^{D_y} \to \mathbb{R}^{d_y}$ to transform the view 1 and view 2 inputs respectively, such that the canonical correlation between the outputs of the DNNs, measured by CCA, is maximized. The objective of CCA is to find $L \leq \min(d_x, d_y)$ pairs of linear projection vectors $\mathbf{U} \in \mathbb{R}^{d_x \times L}$ and $\mathbf{V} \in \mathbb{R}^{d_y \times L}$ such that the projections of each view are maximally correlated with their counterparts in the other view, constrained such that the dimensions in each view are uncorrelated with each other. Therefore, the objective of deep CCA can be written as

$$\max_{\mathbf{W_f}, \mathbf{W_g}, \mathbf{U}, \mathbf{V}} \frac{1}{N} \operatorname{tr}\left(\mathbf{U}^\top \mathbf{f}(\mathbf{X}) \mathbf{g}(\mathbf{Y})^\top \mathbf{V}\right) \quad (1)$$

$$\text{s.t. } \frac{1}{N}\mathbf{U}^\top(\mathbf{f}(\mathbf{X})\mathbf{f}(\mathbf{X})^\top)\mathbf{U} = \frac{1}{N}\mathbf{V}^\top(\mathbf{g}(\mathbf{Y})\mathbf{g}(\mathbf{Y})^\top)\mathbf{V} = \mathbf{I},$$

where $\mathbf{f}(\mathbf{X}) = [\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_N)] \in \mathbb{R}^{d_x \times N}$, $\mathbf{g}(\mathbf{Y}) = [\mathbf{g}(\mathbf{y}_1), \dots, \mathbf{g}(\mathbf{y}_N)] \in \mathbb{R}^{d_y \times N}$. (We assume that $\mathbf{f}(\mathbf{X})$ and

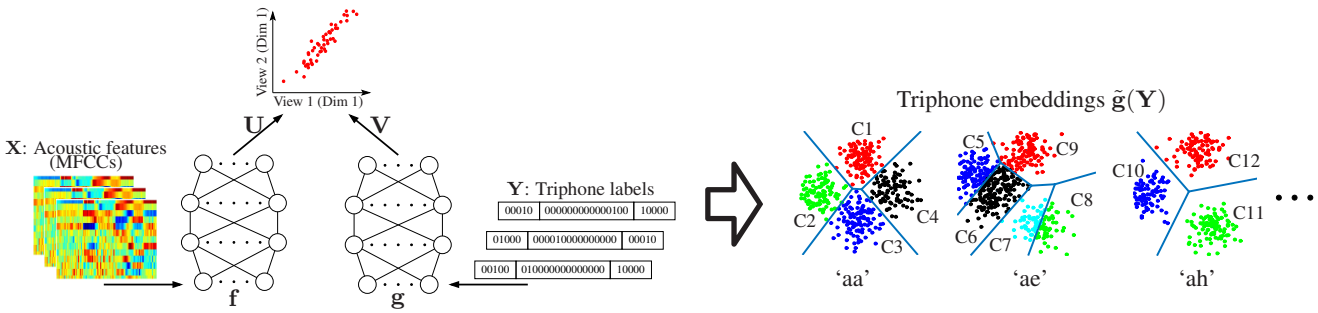


Figure 1: Schematic diagram of state tying with deep CCA-based label embeddings.

$\mathbf{g}(\mathbf{Y})$ are centered at the origin for notational simplicity; in practice, we perform a centering operation for computing the objective.) The parametric form of deep CCA makes it faster to train and test for data sets of reasonable size for speech tasks than the kernel extension of CCA [18]. The final CCA features (projections) are $\tilde{\mathbf{f}}(\mathbf{x}) = \mathbf{U}^\top \mathbf{f}(\mathbf{x})$ for view 1 and $\tilde{\mathbf{g}}(\mathbf{y}) = \mathbf{V}^\top \mathbf{g}(\mathbf{y})$ for view 2.

To get some intuition for this approach, note that the CCA objective is equivalent to a constrained regression problem. By switching $\max(\cdot)$ with $\min -(\cdot)$, and adding $1/2$ times the constraints, we observe that (1) is equivalent to the following objective:

$$\min_{\mathbf{U},\mathbf{V}} \quad \frac{1}{2N} \sum_{i=1}^{N} \left\| \tilde{\mathbf{f}}(\mathbf{x}_i) - \tilde{\mathbf{g}}(\mathbf{y}_i) \right\|^2$$

$$\text{s.t.} \quad \frac{1}{N} \sum_{i=1}^{N} \tilde{\mathbf{f}}(\mathbf{x}_i) \tilde{\mathbf{f}}(\mathbf{x}_i)^\top = \frac{1}{N} \sum_{i=1}^{N} \tilde{\mathbf{g}}(\mathbf{y}_i) \tilde{\mathbf{g}}(\mathbf{y}_i)^\top = \mathbf{I}.$$

That is, CCA minimizes the distance between the projections of the two views, subject to the whitening constraints. This is our first motivation for learning triphone embeddings with deep CCA: The deep CCA objective ensures that the triphone embeddings are highly predictable from the acoustic inputs transformed by DNNs, while the constraints encourage each learned dimension to add new information. It has been validated empirically that the uncorrelatedness constraints are crucial to the good performance of CCA-based feature learning [6].

Another motivation is the successful application of CCA-based methods in the supervised learning setting in prior work. First, it is known that for multi-class classification problems where the two views are inputs and target labels, CCA is equivalent to linear discriminant analysis when the labels are encoded as one-hot vectors [19]. Second, for multi-label classification problems where the label for each input contains a subset of relevant classes, the CCA projections for labels can capture the label correlations [20]. For our application, the triphone labels have a clear structure: there are really 3 labels for each input frame, namely, the phone (state) for the current frame and the previous and next phones in the utterance. Therefore, learning triphone label embeddings with deep CCA should allow us to exploit the correlation between each phone and its context.

3. State tying

Figure 1 provides a schematic diagram of our approach to label embedding and state tying.

3.1. Learning triphone embeddings

We learn the triphone embeddings on the TIMIT dataset [5], using the standard training set with the typical 61 phone labels.

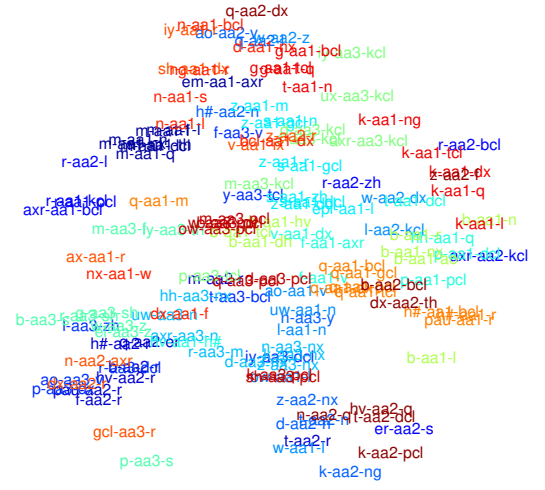
For deep CCA, the acoustic view inputs are based on 13-dimensional mel-frequency cepstral coefficients (MFCCs) computed every 10ms over a 25ms window, along with their first and second derivatives, resulting in 39-dimensional frames. Per-utterance cepstral mean normalization (CMN) is performed when extracting MFCCs. To incorporate context information for the acoustics inputs, we further concatenate the MFCCs over a 15-frame window around each frame. No speaker information is used in our experiments.

The second view inputs for deep CCA are triphone labels obtained as follows. We first divide each phone segment in the manually labeled training data into three sub-segments, whose lengths are 30%/40%/30% of the original segment. We then assign a label of the form '<phone>-begin', or '<phone>-middle', or '<phone>-end' to this frame. This gives $61 \times 3 = 183$ possible (central) labels for each frame, whose purpose is to mimic the 3 states used by typical ASR systems. (An alternative would be to use an alignment produced by a baseline system. Our approach avoids dependence on any particular system.) We then find for each frame its previous and next phone in the utterance. The final input to deep CCA for each frame is the combined 'previous label' (61 classes) + 'central label' (183 classes) + 'next label' (61 classes). We represent each component with a one-hot vector, so the second view inputs are 305-dimensional binary vectors where precisely 3 dimensions are 1.

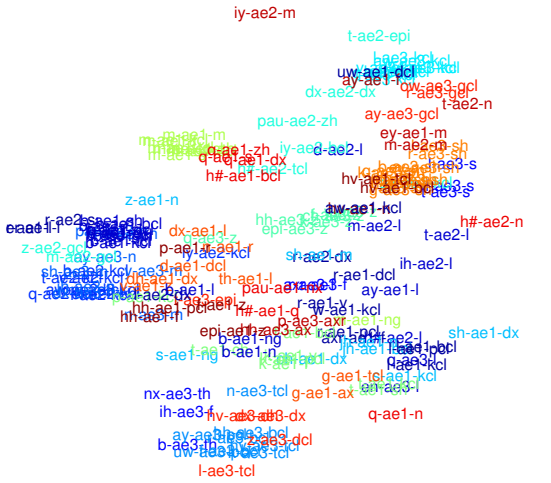
This coding scheme has certain advantages over potential alternatives. First, there are 67554 unique triphone labels in the training set. Our 305-dimensional coding scheme is much more compact than a 67554-dimensional one-hot representation, with a correspondingly smaller number of parameters in the DNN $\mathbf{g}$. Second, unlike a one-hot coding, our approach addresses the problem of unseen triphone labels; as long as all individual phones are seen in training, all possible triphones will have a parametrically defined embedding. Finally, in our structured coding scheme, each phone has different representations depending on its location in the triphone (previous phone, center state, or next phone). The final triphone embeddings are compositions of such location-dependent representations, which are sensitive to the different effects of the same phone in different roles.

Our view 1 network $\mathbf{f}$ has three hidden layers of 2048 ReLUs each [22]. The view 2 network $\mathbf{g}$'s architecture (depth and hidden layer width) is tuned over a coarse grid; the best-performing architecture has three hidden layers of 1024 ReLUs

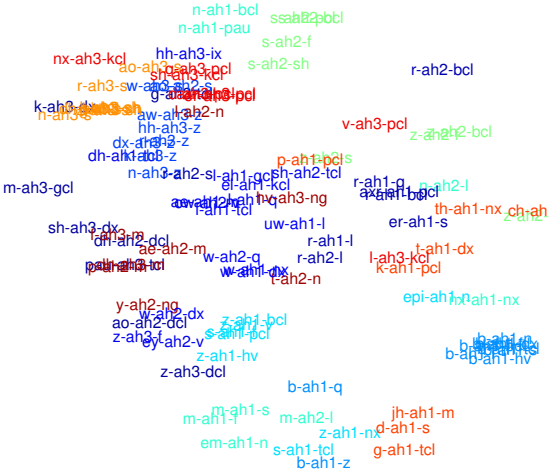
’aa’, # training frames=27566, # clusters=18



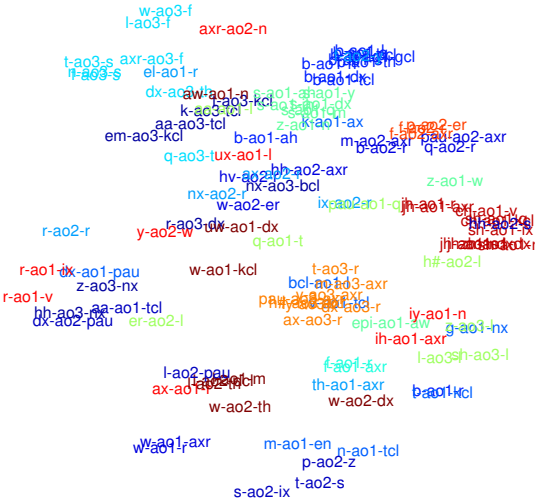
’ae’, # training frames=31003, # clusters=20



’ah’, # training frames=20053, # clusters=13



’ao’, # training frames=22735, # clusters=15



’t’, # training frames=19145, # clusters=12

’m’, # training frames=22339, # clusters=14

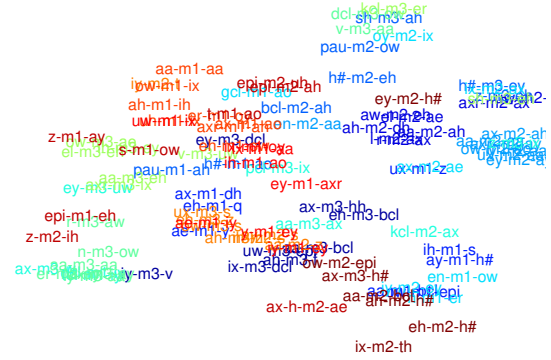


Figure 2: 2D t-SNE [21] visualization of triphone embeddings (subsampled to avoid clutter). Each color represents a cluster.

Table 1: Test phone error rates (PER, %) obtained with MFCC inputs and with bottleneck features of DNNs predicting deep CCA tied states and decision tree tied states.

Features	mono PER	tri1 PER	dnn PER
MFCCs (2500 states)	35.9	30.4	22.3
Bottleneck, decision tree tying (2500 states)	23.9	20.9	19.3
Bottleneck, Deep CCA tying (735 states)	23.6	20.2	19.2

each. The output layer width for both networks, or the dimensionality of triphone embeddings, is tuned over $\{64, 128\}$; 128 tends to perform better and we use this dimensionally in all remaining experiments. Our tuning criterion is the recognition performance of a tandem system (described in Section 3.3) on the TIMIT held-out dev set. We use the stochastic training algorithm of deep CCA [4] with a minibatch size of 8000, learning rate of 0.01, and momentum of 0.99. Training typically converges fairly quickly, after roughly 10 epochs.

3.2. Clustering

After training the deep CCA networks, we obtain a 128-dimensional embedding of all 67554 triphones. We now need to categorize the triphone states into a manageable number of groups, such that each group has sufficient training frames for acoustic modeling.

This is achieved by the following clustering process. For each of the 61 phones, we collect embeddings of triphones for which the phone appears as the 'central label'. Then we replicate each embedding as many times as the corresponding triphone appears in the training set; this ensures that the relative frequency of each triphone is taken into account during clustering. Finally, we perform K-means clustering on the replicated set of embeddings for each central phone. The triphones in each cluster therefore share the same central phone. This mimics the typical practice of learning separate clustering trees for each phone, but in principle it need not be the case. The number of clusters K is simply set to

$$K \leftarrow \lceil \frac{\#\text{training frames with the same central phone}}{p} \rceil$$

p is tuned over $\{400, 800, 1000, 1600\}$, resulting in a total number of clusters of 2786, 1434, 1156, and 735 respectively. In the recognition experiments below, we find that 1156 and 735 clustered states achieve the best performance.

Figure 2 provides 2D visualizations (via t-SNE [21]) of the label embeddings for triphones associated with two of the central phones, with colors indicating cluster identity. These visualizations demonstrate that the learned embeddings and clusters group the labels in intuitive ways, with contexts that we might expect (from linguistic considerations) to have similar effects typically being clustered together.

3.3. Recognition

Here we use a simple protocol for evaluating the learned embeddings and state tying. We train a frame classifier DNN with a bottleneck layer to predict the tied states (clusterID) from the acoustic inputs, and then use the bottleneck features for recognition. The DNN has 3 hidden layers of 3000 ReLUs each, followed by a linear bottleneck layer of 128 units, and a final softmax output layer of 735 units. It has been shown that having a linear bottleneck prior to the softmax layer causes little degradation in classification and recognition performance [23, 24]. The DNN is trained with the cross-entropy objective using stochastic gradient descent, with a minibatch size of 256, an initial learning rate of 0.01 which is halved every time the dev set frame error rate increases, for a maximum of 30 epochs. Dropout training [25] with a rate of 0.5 is used at all ReLU hidden layers.

For comparison, we train a DNN of the same architecture to predict the triphone state alignments produced by a standard Kaldi [2] `tri1` HMM-GMM system, which has 2500 leaves in the decision trees. Both sets of bottleneck features are further reduced to a lower dimensionality by PCA (with dimensionality tuned on the dev set) and fed to a modified Kaldi TIMIT recipe with the pipeline `mono` $\rightarrow$ `tri1` $\rightarrow$ `dnn`, again not using any speaker information. These initial experiments were designed to allow quick tuning and experimentation, but many alternatives for experimental comparison are possible and are the subject of future work.

Phone error rates (PERs, %) obtained by both bottleneck features and MFCCs on the test set are reported in Table 1. We observe that both types of bottleneck features significantly outperform the MFCC features. While the two types of bottleneck features achieve about the same accuracy, our method uses a much smaller number of states.

4. Conclusions

We have proposed a completely data-dependent approach for triphone state tying, based on low-dimensional triphone label embeddings learned by deep canonical correlation analysis. We have shown that our triphone clusters are suitable targets for acoustic modeling, and lead to competitive phone recognition performance while using a smaller number of clusters.

There are many directions for going beyond these initial experiments. To fully exploit the power of our state tying, we will use the tied states in a hybrid approach, i.e., instead of using the bottleneck features of DNN classifiers predicting the cluster ID, we could directly use the predicted posterior probabilities, together with suitable transition probabilities between the clusters or states, for decoding.

There are also many potential variants of the approach that can be explored. For example, any clustering can be applied instead of simple K-means. A hierarchical clustering may be appropriate, and may also hold independent interesting as a way of learning the structure in a speech data set. We can also consider different alternatives for the initial structured coding of the labels, such as including binary phonetic feature encodings of the central and context phones; this would bring back some of the flavor of typical decision tree questions, while being learned in a more discriminative way governed by the CCA objective.

Acknowledgements This research was supported by NSF grant IIS-1321015. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funding agency. The Tesla K40 GPUs used for this research were donated by NVIDIA Corporation.

5. References

[1] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of the Workshop on Human Language Technology*, 1994.

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[3] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *ICML*, 2013.

[4] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "Unsupervised learning of acoustic features via deep canonical correlation analysis," in *ICASSP*, 2015.

[5] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "TIMIT acoustic-phonetic continuous speech corpus," 1993, lDC93S1.

[6] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *ICML*, 2015.

[7] A. Lu, W. Wang, M. Bansal, K. Gimpel, and K. Livescu, "Deep multilingual correlation for improved word embeddings," in *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT 2015)*, 2015.

[8] F. Yan and K. Mikolajczyk, "Deep correlation for matching images and text," in *CVPR*, 2015.

[9] R. Arora and K. Livescu, "Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains," in *ICASSP*, 2013.

[10] E. Kidron, Y. Y. Schechner, and M. Elad, "Pixels that sound," in *CVPR*, 2005.

[11] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng, "Multimodal deep learning," in *ICML*, 2011.

[12] R. Socher and F.-F. Li, "Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora," in *CVPR*, 2010.

[13] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013.

[14] A. Vinokourov, N. Cristianini, and J. Shawe-Taylor, "Inferring a semantic representation of text via cross-language correlation analysis," in *NIPS*, 2003.

[15] A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein, "Learning bilingual lexicons from monolingual corpora," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, 2008.

[16] S. Chandar, S. Lauly, H. Larochelle, M. M. Khapra, B. Ravindran, V. Raykar, and A. Saha, "An autoencoder approach to learning bilingual word representations," in *NIPS*, 2014.

[17] M. Faruqui and C. Dyer, "An information theoretic approach to bilingual word clustering," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[18] R. Arora and K. Livescu, "Kernel CCA for multi-view learning of acoustic features using articulatory measurements," in *Symposium on Machine Learning in Speech and Language Processing (MLSLP)*, 2012.

[19] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," Dept. of Statistics, University of California, Berkeley, Tech. Rep. 688, 2005.

[20] Y. Zhang and J. Schneider, "Multi-label output codes using canonical correlation analysis," in *AISTATS*, 2011.

[21] L. J. P. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, 2008.

[22] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton, "On rectified linear units for speech processing," in *ICASSP*, 2013.

[23] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *ICASSP*, 2013.

[24] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *ICASSP*, 2014.

[25] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, 2014.