Phonetics embedding learning with side information

Gabriel Synnaeve[1] **Thomas Schatz**[1,2]**, Emmanuel Dupoux**[1]
[1] LSCP, IEC ENS/EHESS/CNRS, Paris, France
[2] SIERRA Project-team INRIA/ENS/CNRS, Paris, France
gabriel.synnaeve@gmail.com, thomas.schatz@laposte.net, emmanuel.dupoux@gmail.com

Abstract

We show that it is possible to learn an efficient acoustic model using only a small amount of easily available word-level similarity annotations. In contrast to the detailed phonetic labeling required by classical speech recognition technologies, the only information our method requires are pairs of speech excerpts which are known to be similar (same word) and pairs of speech excerpts which are known to be different (different words). An acoustic model is obtained by training shallow and deep neural networks, using an architecture and a cost function well-adapted to the nature of the provided information. The resulting model is evaluated on an ABX minimal-pair discrimination task and is shown to perform much better (11.8% ABX error rate) than raw speech features (19.6%), not far from a fully supervised baseline (best neural network: 9.2%, HMM-GMM: 11%).

1 Introduction

State-of-the art speech recognition systems rely on the availability of large quantities of human-annotated signals. However, it is also of interest, both for theoretical and practical reasons to explore the possibility of constructing speech technologies in settings where such a resource is not available. Theoretically, algorithms performing unsupervised or weakly supervised discovery of linguistic structure represent plausible models of language acquisition in the human infant [1]. Practically, such algorithms can be put to use in situations of low resources [2, 3].

Previous works have shown that same-different side information can benefit metric learning [4]. Siamese networks [5] describe a very similar model architecture as ours, used to authentify handwritten signatures. Hadsell et al. [6] also used an asymmetric loss for same-different pairs to learn invariant-properties manifolds (on images). However, to our knowledge, such an approach has not been attempted in the area of speech recognition. In the cognitive science literature, previous work have shown that top-down information derived from a lexicon learned in an unsupervised fashion can help refine phoneme categories [7, 8, 9, 10]. These studies, however, did not use raw speech signals as input, but used either a fine grained allophonic transcription [7, 8, 10] or distributions based on phonetic measurements [11].

Here, we examine whether speech features can be derived from a Neural Net (NN) architecture using side information. We assume that the learner has managed to segment some words out of the continuous speech stream, presumably using some version of spoken term discovery [3]. We use the resulting lexicon of word forms to learn an embedding of speech sounds which is more invariant than the original speech representation to variations in the acoustic realization of phonemes. The idea is to establish a loss function whereby two instances of the same word are pulled together, and two instances of different words are differentiated. The loss function is actually computed at the frame level, after an alignment through Dynamic Time Warping. In this paper, we focus on testing whether this kind of side-information about the lexicon is sufficient to train an acoustic model, *in principle*. To this end, we use the "gold" human-annotated lexicon to derive the side-information. Future work will investigate whether our results generalize to the fully unsupervised case, whereby a lexicon derived from spoken term discovery and/or word segmentation [12] is used.

In the next section, we describe how to prepare the dataset, how to build an "ABnet" and discuss loss functions. In section 3., we present our results, detailing our training setup and evaluation procedure. We conclude by laying out the possible extensions of this work.

2 Methods

2.1 Data preparation

For the speech signal, frames are taken every 10ms and each is encoded by a 40 log-energy Mel-scale filter bank representing 25ms of speech (Hamming windowed), without deltas or delta-delta coefficients. This encoding was used by several state-of-the-art supervised deep-learning phone-recognition systems [13, 14].

Separately for each of the tree sets (train, validation, test) of the standard split of the TIMIT corpus (without sentences from the *sa* set), we searched for all the words of more than 5 characters that are repeated (by the same talker or another) and applied dynamic time warping (DTW, [15]) to their speech signal (see Figure 1). That gives us train/validation/test sets of aligned speech frames that represent, in principle, the same phonemes. Here, we ignored heteronyms like (to) record and (the) record, which were considered identical. We sampled the same number of different word pairs in order to generate negative examples. This way, we have about the same statistics of same speakers for both of the words in a pair, and statistically the same phonetics. We did not DTW align these pairs of different words, we just aligned them linearly on the shorter word. It can happen (by chance) that some of these negative examples are indeed positive pairs (of same things, even though we labeled them as "different"), but that is not the most likely case (there are about 39 core English phonemes, even if these are distributed following a power law).

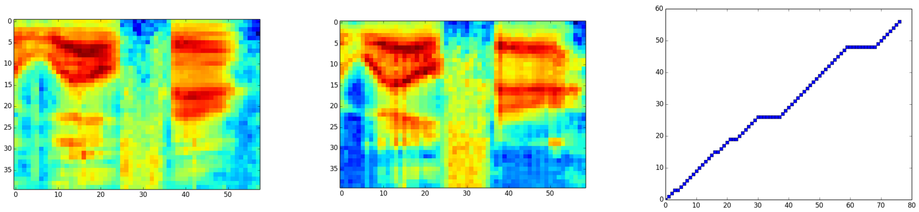

Figure 1: 2 left plots: filterbanks (y-axis) along frames (x-axis) for the word "welfare". Right: dynamic time warping of the left-most one to the other.

2.2 ABnet Architecture

We designed a neural network that takes two chunks of speech as input and projects them in the vector space formed by the last layers. To train this neural network, we optimize a distance or similarity in this embedding space, between the two inputs of the network: minimize this distance if they are the "same" or maximize it if they are "different". To give an example in practice: for the results section, our "small" ABnet takes 2 DTW-aligned chunks of 70ms (7 frames) of speech and percolates them up to the distance- or similarity-based loss function (see Figure 2). At first, we used as the loss function the normalized Euclidean distance in the last layer (in 100 dimensions) if the two inputs were representing the "same" thing, or minus this distance is they were "different". We quickly found out that it was better to minimize two distances, one for the case "same", and one for "different".

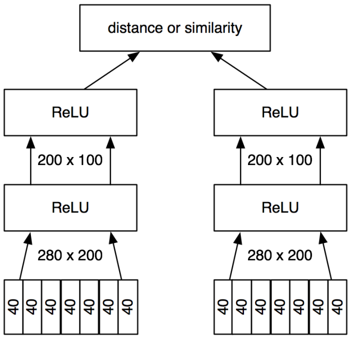

Figure 2: AB "neural net". We feed to two copies of the same network the aligned stacked frames of a pair of words (A and B). The outputs are compared using a dissimilarity function. During training, the loss function tries to minimize the dissimilarity for "same" pairs and maximize it for "different" pairs. The loss is backpropagated in both sides of the network equivalently.

For the nonlinearity, we used rectified linear units because they are used widely in state of the art deep neural networks nowadays (speech, vision) and they performed as well as sigmoid units for our task, while being faster to train. Rectified linear units activation function corresponds to:

$$h(v) = \max(0, v)$$

For the cost function, we tried several metrics, and present the following in the results: the Normalized Euclideaan Distance (Loss_{NED}), the squared similarity ($\text{Loss}_{\text{COS}^2}$), and cosine squared cosine ($\text{Loss}_{\text{COSCOS}^2}$) as explained below. Consider Y_A and Y_B being the output representations for input words A and B:

$$\text{Loss}_{\text{NED}}(A, B) = \begin{cases} \text{NED}(Y_A, Y_B) & if \text{ same} \\ 1 - \text{NED}(Y_A, Y_B) & if \text{ different} \end{cases}$$

with

$$\text{NED}(x, y) = \frac{\|x - y\|}{\|x\| \|y\|}$$

This normalized Euclidean distance makes sense but does not give conclusive results, as it seems to be very hard to train our architecture with it. Particularly the "different" case is much easier to solve than the "same" one, and without "different", the null function is an obvious solution (mapping everything to $0_{d=100}$).

$$\text{Loss}_{\text{COS}^2}(A, B) = \begin{cases} 1 - \cos^2(Y_A, Y_B) & if \text{ same} \\ \cos^2(Y_A, Y_B) & if \text{ different} \end{cases}$$

with

$$\cos(x, y) = \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

Using the cosine similarity (not squared) as the loss function can train our ABnet and gives interesting results. Note however that cosine is maximal when the angle is 0 (collinear) while it is minimal when the angle is π (anti-collinear). That forces two things:

- "different" examples embedding representations are drawn towards anti-collinearity, which is harder to achieve than orthogonality in a high dimensional space[1].

[1] collinearity is hard to achieve too, but at least the inputs are supposed to be "similar" sounds

- this loss function entails that we will have negative units, which may be a hindrance in our case of using rectified linear units (using the biases).

So instead, we used the squared cosine similarity ($\cos^2$) as noted above, in which "different" examples should be orthogonal, while "same" examples should be collinear or anti-collinear.

$$\text{Loss}_{\text{coscos}^2}(A, B) = \begin{cases} (1 - \cos(Y_A, Y_B))/2 & if \text{ same} \\ \cos^2(Y_A, Y_B) & if \text{ different} \end{cases}$$

With the above observations in mind, we devised a loss function that makes the "same" examples embedding representations collinear, while making the "different" ones orthogonal. Note that both parts of this loss function prohibit vectors to be anti-correlated. This loss function also pulls the output to be positive (with angles in the $[0; \pi/2]$ quadrant).

The whole feedforward computation of the smaller ABnet that we present in the results is thus:

$$dist(\max(0, H^{(2)} \cdot \max(0, H^{(1)} \cdot X_A)), \max(0, H^{(2)} \cdot \max(0, H^{(1)} \cdot X_B)))$$

with $H^{(1)}$ of dimensions 200×100 and $H^{(2)}$ of dimensions 280×200. The bigger (so called "deep") ABnet is comprised of 4 rectified linear units, noted: 280 × 1000 × ReLU × 1000 × ReLU × 1000 × ReLU × 100 × ReLU.

3 Results

3.1 Setup

All the code to reproduce these results is free (BSD 3-clause) and available on Github[2]. To compare our models, we also implemented two supervised (with phones annotations) architectures that we trained on TIMIT in a standard fashion (on forced aligned phone's states from HTK, e.g. as in [16]). The first supervised architecture is a supervised version of our embedding with the last ReLU layer replaced by a logistic regression (softmax), with output in the 186 phone states: 60 phones + start/end silences, 3 states for each phone. It is called "supervised 7" as it takes 7 consecutive frames as input. This supervised architecture gives 49.9% of phone's state frame accuracy on the TIMIT test set. The second consists of 4 1000-dimensional-ReLU hidden layers followed by the same logistic regression that has competitive phone error rates (24.6% on the TIMIT test set).

Finally, we also included a supervised HMM-GMM baseline trained on the train set with HTK. As speech features, we used 13 MFC coeficients plus the first and second temporal derivatives. We used a three-state monophone 17 mixtures architecture and talker-specific MLLR adaptation (both during training and during test). We also trained a bigram language model for the decoding. This baseline achieved 25.5% phone error rate on the test set.

3.2 Training and visualization

All the models (weakly supervised embedding and supervised ones) were trained using Adadelta [17], an adaptive learning rate method correcting the magnitude of the updates using an accumulation of past gradients (on a sliding window) and a local approximation of the Hessian. While the same results should be obtainable with plain stochastic gradient descent, Adadelta is faster to converge and requires (almost) no tuning: we used a rho (hyper-parameter on the momentum) value of 0.9 and an epsilon (precision of the updates) of 10^{-6}.

For the DTW-aligned dataset, the training set consisted of 62,625 paired same words, yielding 5.66M frames for "same" and 4.49M for "different" (both for same and different, we see

[2] `https://github.com/SnippyHolloW/abnet`

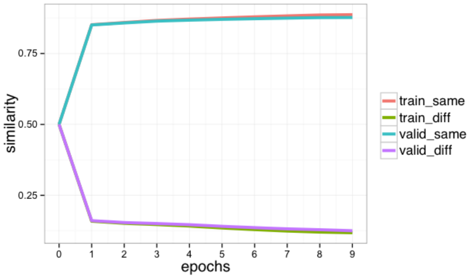

Figure 3: Similarities (coscos^2) for the "same" and "different" subsets of the datasets on the training set (used for learning) and on the validation set (used for early stopping).

the frames by AB pairs), with ratios of same speaker and over different speakers (in the pairs) of 0.0014 and 0.0022 respectively. Note that a lot of these frames are duplicates and not all the frames of the TIMIT train set are seen in our DTW-aligned word pairs dataset. 10% of the training set were held-out as a validation set for early-stopping. The evolution of the similarities for the coscos^2 is shown in Figure 3. Unlike some other loss functions, we do not see any unbalance between the "same" and "different" conditions. For the supervised learning (with phonetic annotation), we use the standard TIMIT split: the train set is comprised of 1.1M frames, the validation set (for early stopping) of 122K frames, and the test set (frame accuracy results) of 57K frames, all three had different speakers. Each frame is always seen with its surrounding +/-3 frames (or +/-5 in the case of an 11 stack) but it still means that we have 1 training example per frame.

In Figure 4, we compare the representation learnt with our best ABnet (deep 7 coscos^2 architecture) to the initial speech features (*fbanks1*) in their coding of the 39 core English phones from the TIMIT annotations. Specifically, for each representation, we performed bi-clustering on the matrix containing on line i, column j, the average activation of dimension i of the representation when the sound being represented is an acoustic realisation of phone j. The dimensions of the representation correspond to frequency channels in the *fbanks1* representations. They have no clear *a priori* interpretation for the ABnet representation. Interestingly, the clusters of phones obtained through bi-clustering are not very different for the two representations. However, there are at least two clear differences. First, the learnt representation, when averaged over each phone, is much sparser than the original representation: each channel only respond to a few phones. Second, when averaged over each phone, the learnt representation is almost categorial, i.e. it is possible to find roughly a one-to-one mapping between clusters of channels and clusters of phones such that the channels are only activated (on average) when presented with a phone from the associated phone cluster. In contrast, *fbanks1* representation is not categorial at all. Overall, this shows that, beyond the quantitative improvement in phoneme discriminability, tics embedding learnt by our ABnet is qualitatively different from the original spectro-temporal representation.

3.3 Evaluation ABX

We evaluate the resulting models in a minimal-pair ABX phone discrimination task [18]. This evaluation method relies on the simple idea that, in a good model, representations of different occurrences of a same phone should be "close" to each other and "far away" from occurrences of other phones. It does not require choosing or training a particular classifier, the only thing needed being a notion of distance on the space of the representations of speech sounds by the model. Here, we use as a distance the sum of the frame-to-frame cosine or the symmetric Kullback-Leibler (KL) divergence along the optimal DTW path. The use of DTW can be understood as providing some invariance to changes in the time course of the speech sounds. The choice of the frame-to-frame metric is based on the kind of representation considered. The use of a cosine distance can be understood as providing some invariance to changes in the loudness of the speech sounds and is suitable for representations where scaling is best ignored, such as filterbank representations for example. The use of a Kullback-Leibler

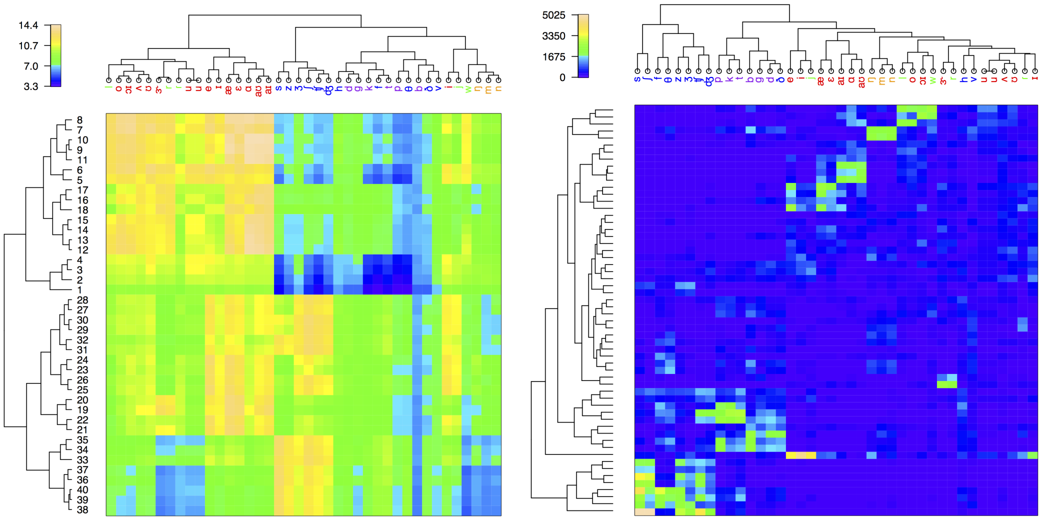

Figure 4: Bi-clustering of the mean activations of (top plot) the 40 filterbank features (y-axis) and (bottom plot) the most activated 58 embedding features (y-axis), with the phonetic input (x-axis). The phones are clustered as fricatives: blue, stops: purple, liquids (and semi-vowels and flaps): green, nasals: orange, vowels: red.

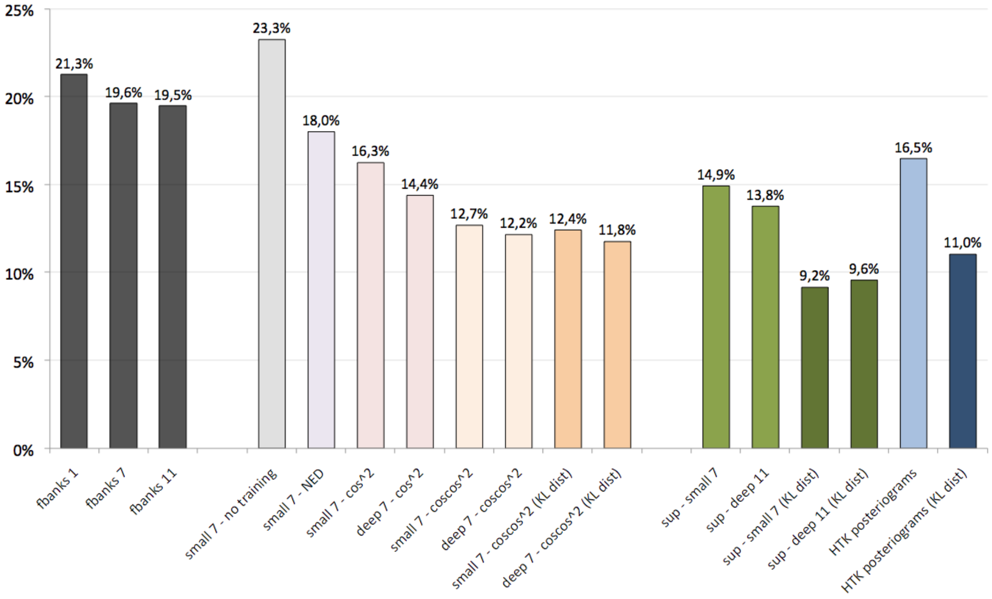

Figure 5: Average phone discrimination error-rate (ABX task, averaged over talkers and triphone contexts). In dark grey (black) are the filterbanks and stacked filterbanks that are used as input features for subsequent models. The middle cluster (pastels of orange) represents our ABnets. In green are the supervised neural networks, and in blue are the supervised HMM-GMM trained with HTK. “small” indicates a network with one hidden layer of 200 units, while “deep” indicates a network with 3 hidden layers, plus the last one whose size depends on the task (100 for the embedding, 186 for the supervised phone states). When not noted in parenthesis, the “distance” used for the ABX task was the cosine similarity.

distance is suited to representations that can be interpreted as probability distributions, such as the (rescaled) posteriorgrams over phonemes obtained with the supervised baselines.

To perform the evaluation, all sequences of three consecutive phones $p_1 - p_2 - p_3$ in TIMIT are extracted. Then triplets of these triphones are formed such that the first triphone (A) and the second triphone (B) differ only by their central phone and are spoken by the same talker, and such that the last triphone (X) is constituted from the exact same phones as either A or B, but spoken by a different talker. All possible triplets verifying this pattern are formed and for each of them the representations a, b, x by the model under evaluation of the three triphones A, B, X are computed. DTW distances $d(a, x)$ and $d(b, x)$ between respectively a and x and b and x are then computed based on the frame-to-frame cosine or KL distance. If X is like A (i.e. constituted from the exact same phones as A), the model correctly classifies the triplet if $d(a, x) < d(b, x)$; if X is like B, the model correctly classifies the triplet if $d(b, x) < d(a, x)$. For each possible pair (p_1, p_2) of phones, an average proportion $s_{p1,p2}$ of correctly classified triplets is computed over all triplets A, B, X where the central phone of A is either p_1 or p_2 and the central phone of B is either p_2 or p_1. The ABX discrimination score is obtained as the average of s_{p_1,p_2} over all possible pairs (p_1, p_2) of phones, weighted by $w_1 w_2$, where w_i is the frequency of occurrence of the phone p_i in TIMIT. The ABX error-rate is obtained as one minus the ABX score.

All the results in this ABX phone discrimination task are summarized in Figure 5. The chance level for the error rate in this task is at 50%. At 11.8% error-rate, our best ABnet (coscos^2 with KL) improves by almost 9% in absolute over the speech features it was trained with as input (*fbanks7*). Note that raw speech features perform better in ABX with the cosine similarity than the symmetric KL divergence. We tried other stacks of speech features: (*fbanks 1*), and an 11 stacked frames (*fbanks 11*) version (the input of the competitive supervised deep neural network). Stacking more and more speech features (more and more temporal / phonotactical context) helps, but with rapidly diminishing returns. We confirmed that the random projections alone were not responsible for our gain over raw features by testing the ABnet with random weights (*small 7 - no training*. This even added noise and yielded a worse score than speech features.

Overall, deeper models perform better. Also, we can clearly see that some loss functions are better than others, with NED being marginally better than raw speech features (18% ABX error rate), while cos^2 gets better (deep 7 cos^2 is as 14.4%) and coscos^2 has the best scores, with the cosine similarity (12.2%), or the KL divergence (11.8%).

As a comparison, we looked at the ABX scores of two supervised neural network, both matching our ABnet architectures ("small" or "deep"), but with an additional Softmax at the end. The first (*sup-small 7*) has an ABX error-rate of 14.9% with the cosine similarity and 9.2% with the KL divergence. The second is a deep (4 hidden layer) neural network (*sup-deep 11*) with 11 frames as input, it gives the best phone error rate, an ABX error-rate of 13.8% with the cosine similarity, and 9.6% with the KL divergence. For evaluating the ABX performance on the HMM-GMM baseline, we first derived a frame-by frame posteriogram from an N-best decoding lattice (with N=62). The results are not very favorable when using the cosine distance, but achieve 11.0% of error rates with the KL divergence. It is remarkable, though, that the supervised neural network achieves a better performance than the HMM-GMM baseline despite the fact that the latter but not the former incorporates a bigram language model.

4 Discussion

We trained an acoustic model using only a small amount of easily available annotations and evaluated it in an ABX minimal-pair discrimination task. The resulting model, with an average ABX error of 11.8%, was much closer in performance to a fully supervised baseline (9.2%) than to raw speech features (19.5%). This work has implications for two distinct fields of research: low-resource speech technologies and studies of early language acquisition [19]. For low-resources speech technology, it provides a practical way to learn an efficient acoustic model. For studies of language acquisition, it lends plausibility to the hypothesis that simple measures of similarity between word-size units of speech signal constitute one of the sources of information that are used by infants when learning the phonetic categories of their language.

In future work, we will investigate further the nature of the learnt phonetic embedding by studying in detail the differences and similarities with the embeddings obtained by fully supervised methods. We can use clustering over the embedding (and use a Mahalanobis distance as the loss function), to see if we learn phone-states, phones, tri-phones, syllables... We can also alternatively re-do the dynamic time warping using our embedding features. We will also try and make the method truly zero-resource by using Spoken Term Discovery technologies [3] to provide the required similarity labels from raw speech. This will require testing the method's robustness to noise in the similarity labels. Finally, we will investigate ways of combining the learnt acoustic model with a language model to build a full speech recognition pipeline in a low-resource setting.

5 Acknowledgements

This project is funded in part by the European Research Council (ERC-2011-AdG-295810 BOOTPHON), the Agence Nationale pour la Recherche (ANR-10-LABX-0087 IEC, ANR-10-IDEX-0001-02 PSL*), the Fondation de France, the Ecole de Neurosciences de Paris, and the Region Ile de France (DIM cerveau et pensée).

References

[1] Gautam K Vallabha, James L McClelland, Ferran Pons, Janet F Werker, and Shigeaki Amano. Unsupervised learning of vowel categories from infant-directed speech. Proceedings of the National Academy of Sciences, 104(33):13273–13278, 2007.

[2] Alex S. Park and James R. Glass. Unsupervised pattern discovery in speech. IEEE Transactions on Audio, Speech, and Language Processing, 16(1):186–197, January 2008.

[3] Aren Jansen, Kenneth Church, and Hynek Hermansky. Towards spoken term discovery at scale with zero resources. In INTERSPEECH, pages 1676–1679, 2010.

[4] Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. Advances in neural information processing systems, pages 521–528, 2003.

[5] Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence, 7(04):669–688, 1993.

[6] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In Computer vision and pattern recognition, 2006 IEEE computer society conference on, volume 2, pages 1735–1742. IEEE, 2006.

[7] Andrew Martin, Sharon Peperkamp, and Emmanuel Dupoux. Learning phonemes with a proto-lexicon. Cognitive Science, 37(1):103–124, 2013.

[8] Abdellah Fourtassi and Emmanuel Dupoux. A rudimentary lexicon and semantics help bootstrap phoneme acquisition. In ConLL-2014, 2014.

[9] Naomi H Feldman, Thomas L Griffiths, Sharon Goldwater, and James L Morgan. A role for the developing lexicon in phonetic category acquisition. Psychological review, 120(4):751, 2013.

[10] Daniel Swingley. Statistical clustering and the contents of the infant vocabulary. Cognitive psychology, 50(1):86–132, 2005.

[11] N.H. Feldman, T.L. Griffiths, and J.L. Morgan. Learning phonetic categories by learning a lexicon. In Proceedings of the 31st annual conference of the cognitive science society, pages 2208–2213. Cognitive Science Society (CD-ROM) Austin, TX, 2009.

[12] Michael R Brent. Speech segmentation and word discovery: A computational perspective. Trends in Cognitive Sciences, 3(8):294–301, 1999.

[13] MD Zeiler, M Ranzato, R Monga, M Mao, K Yang, QV Le, P Nguyen, A Senior, V Vanhoucke, J Dean, and G.E. Hinton. On rectified linear units for speech processing. In ICASSP, 2013.

[14] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 6645–6649. IEEE, 2013.

[15] TK Vintsyuk. Speech discrimination by dynamic programming. Cybernetics and Systems Analysis, 4(1):52–57, 1968.

[16] A. Mohamed, G. Dahl, and G. Hinton. Deep belief networks for phone recognition. In NIPS Workshop on Deep Learning for Speech Recognition and Related Applications, 2009.

[17] Matthew D Zeiler. Adadelta: An adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012.

[18] Thomas Schatz, Vijayaditya Peddinti, Francis Bach, Aren Jansen, Hermansky Hynek, and Emmanuel Dupoux. Evaluating speech features with the minimal-pair abx task: Analysis of the classical mfc/plp pipeline. In INTERSPEECH-2013, pages 1781–1785, 2013.

[19] Aren Jansen, Emmanuel Dupoux, Sharon Goldwater, Mark Johnson, Sanjeev Khudanpur, Kenneth Church, Naomi Feldman, Hynek Hermansky, Florian Metze, Richard Rose, Mike Seltzer, Pascal Clark, Ian McGraw, Balakrishnan Varadarajan, Erin Bennett, Benjamin Borschinger, Chiu Justin, Ewan Dunbar, Abdellah Fourtassi, David Harwath, Chia-ying Lee, Keith Levin, Atta Norouzian, Vijayaditya Peddinti, Rachael Richardson, Thomas Schatz, and Samuel Thomas. A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition. In ICASSP, pages 8111–8115, 2013.