DEEP NEURAL NETWORK FEATURES AND SEMI-SUPERVISED TRAINING FOR LOW RESOURCE SPEECH RECOGNITION

Samuel Thomas[1], Michael L. Seltzer[2], Kenneth Church[3] and Hynek Hermansky[1]

[1] The Johns Hopkins University, Baltimore, USA
[2] Microsoft Research, Redmond, USA
[3] IBM Research, Yorktown Heights, USA
{samuel,hynek}@jhu.edu,
mseltzer@microsoft.com, kwchurch@us.ibm.com

ABSTRACT

We propose a new technique for training deep neural networks (DNNs) as data-driven feature front-ends for large vocabulary continuous speech recognition (LVCSR) in low resource settings. To circumvent the lack of sufficient training data for acoustic modeling in these scenarios, we use transcribed multilingual data and semi-supervised training to build the proposed feature front-ends. In our experiments, the proposed features provide an absolute improvement of 16% in a low-resource LVCSR setting with only one hour of in-domain training data. While close to three-fourths of these gains come from DNN-based features, the remaining are from semi-supervised training.

Index Terms— Low resource, speech recognition, deep neural networks, semi-supervised training, bottleneck features.

1. INTRODUCTION

Acoustic models for state-of-the-art speech recognition systems are typically trained on several hundred hours of task specific training data, but in low resource scenarios, one often has to make do with much less training data. Annotated training data can be especially hard to come by. In these settings, it is possible to take advantage of transcribed data from other languages to build multilingual acoustic models [1, 2]. Multilingual training with Subspace Gaussian Mixture Models [3] have also been proposed to train acoustic models [4, 5].

An alternative approach moves the focus to data-driven feature front-ends. The key element in this data-driven approach is a multilayer perceptron (MLP) trained on large amounts of task independent data, i.e. multilingual data or data from the same language but collected under different settings [6, 7]. Features corresponding to limited task specific data are then derived using the trained MLP for ASR [8, 9, 10, 11, 12]. We build on this front-end-based approach since features produced using these front-ends can further improve performance in low-resource settings when combined with other ASR modeling techniques.

While this work is related to several recent approaches [8, 9, 10, 11, 12], we use two different techniques to derive better features and improve acoustic model training in low resource settings: data driven features extracted using deep neural networks (DNN) and semi-supervised training. We show that both these techniques significantly improve the performance of ASR systems in these scenarios. In Section 2 we discuss the use of multilingual data to build a DNN-based front-end. The use of semi-supervised training to improve both the DNN and GMM-HMM-based acoustic models is discussed in Section 3. The paper concludes with Section 4.

2. DEEP NEURAL NETWORK-BASED FEATURES

A deep neural network (DNN) is an MLP with several more layers than traditionally used networks. The parameters of a DNN are often initialized using a pre-training algorithm before the network is trained to completion using error back-propagation [13, 14]. In this section we discuss the development of a DNN for low-resource scenarios.

2.1. DNN pre-training and initialization

The purpose of pre-training is to initialize the parameters of a DNN network to a better starting point than random initialization prior to back-propagation. Networks trained from pre-trained weights are observed to be well regularized and converge to a better local optimum than randomly initialized networks [15, 16]. As with traditional neural networks, DNNs have been used both as acoustic models that directly model context-dependent states of HMMs [17] and also as data-driven feature extractors [18, 19]. In both cases, DNNs have outperformed traditional shallow networks [17, 20].

DNNs can be pre-trained using either a generative or discriminative approach. In generative pre-training, the network is trained in a layer-by-layer manner, by treating each successive pair of layers as a restricted Boltzmann machine (RBM). The weights that connect a pair of layers are trained in an unsupervised fashion using an approximate maximum likelihood criterion known as contrastive divergence [21]. Alternatively, the network can be initialized using discriminative pre-training [13, 22]. This procedure starts by training an MLP with 1 hidden layer using back-propagation. These weights are fixed and a new randomly initialized hidden layer and output layer are introduced to replace the output layer of the initial network. The deeper network is then trained again using back propagation. This procedure is repeated until the desired number of hidden layers are in place.

Although pre-training algorithms are effective in initializing DNNs, they require sufficient training data to perform properly. In low resource settings, the amount of data available is often insufficient. We show that in these scenarios, pre-training can be

This work was supported in parts by the DARPA RATS project D10PC0015, IARPA BABEL project W911NF12-C-0013, and by the Johns Hopkins Center of Excellence in Human Language Technologies.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

performed using multilingual data before the DNN is finally adapted with a limited amount of task specific monolingual data.

We propose to train a DNN with L wide layers and a bottleneck layer - $d, h_1, \ldots, h_L, h_B, p$. The network has a linear input layer with a size d corresponding to the dimension of the input feature vector, followed by several nonlinear layers $h_1, \ldots, h_L, h_B$ and a final soft-max output layer of size p corresponding to the phoneset of the multilingual data the DNN is being trained on. Both posterior and bottleneck features [23, 24] can be derived from the DNN. We use the following steps to pre-train a DNN:

1. *Initialize the network:* We perform discriminative pre-training, starting with a network with 1 hidden layer - d, h_1, p. The weights are initialized randomly and trained using a single epoch of back propagation, similar to [13].
2. *Grow the network:* The d, h_1, p network is now grown by inserting a new layer h_2 with randomly initialized weights connecting h_1-h_2 and h_2-p. The weights in the first layer are kept fixed and a single epoch of back propagation is performed to pre-train the weights in the second layer. This process is repeated for subsequent layers until the L hidden layers have been added. The final network $d, h_1, \ldots, h_L, h_B, p$ is created by adding a bottleneck layer h_B. While weights connecting the wide layers till L-1 are copied from the previous step, new random weights are used to connect h_L-h_B and h_B-p.
3. *Train the full network:* After all the layers of the network have been discriminatively pre-trained, the complete network is trained to convergence.

Once the DNN has been trained using multilingual data, it is updated using limited amounts of monolingual data from the low-resource setting.

2.2. Updating the DNN with monolingual data

Incompatible phoneme sets can be a challenge for adapting networks across languages. In previous work, we proposed the use of a modified neural network in which the final phoneme-dependent soft-max layer is replaced [11]. We use this technique in this work for adapting the DNN as well.

1. *Initialize the network for the low resource language:* To train a DNN for a new language with a different phoneme set q, we use the multilingual DNN described in Sec. 2.1 but replace the parameters of the output layer with randomly initialized weights. These weights between h_B and q are then discriminatively trained using monolingual data, keeping the parameters of the lower layers fixed.
2. *Update the network on the low resource data:* Once the new DNN $d, h_1, \ldots, h_L, h_B, q$ has been initialized, we update all the parameters using the low-resource language.

Features for ASR are then derived from the bottleneck layer of the final DNN.

2.3. Experiments and Evaluations

We use the English, German and Spanish parts of the Callhome corpora collected by LDC for our experiments [25, 26, 27]. The English database consists of 120 spontaneous telephone conversations between native English speakers. The complete training set consists of 80 conversations, corresponding to about 15 hours of speech [25]. We use 1 hour of randomly chosen speech from the training set for our experiments as an example of data from a low-resource language. The English DNNs and subsequent HMM-GMM systems use this one hour of data. Two sets of 20 conversations, roughly containing 1.8 hours of speech each, form the test and development sets. The German and Spanish databases contain 100 and 120 spontaneous telephone conversations, respectively, between native speakers. We use 15 hours of German and 16 hours of Spanish as data from out-of-domain high resource languages for training the DNNs. Each of these three languages have a different phoneme set: 47 phonemes for English, 46 for German and 28 for Spanish.

Speech recognition experiments are performed using HTK. We train an acoustic model with 600 tied states and 4 Gaussians per state on the 1 hour of data from the low resource language. We use fewer states and components per state mixture since the amount of training data is low. The recognizer uses a 62K trigram language model with an OOV rate of 0.4%, built using the SRILM tools. The language model is interpolated from individual models created using the English Callhome corpus, the Switchboard corpus [28], the Gigaword corpus [29] and some web data. The web data is obtained by crawling the web for sentences containing high frequency bigrams and trigrams occurring in the training text of the Callhome corpus. The 90K PRONLEX dictionary with 47 phones is used as the pronunciation dictionary for the system. The test data is decoded using the HDecode decoder from HTK, and scored with NIST scoring scripts.

We build a multilingual DNN front-end by combining data from Spanish, German and English. Separate DNNs are trained on two different feature representations, short-term spectral PLP features [30] and long-term FDLP-based modulation features [31]. Bottleneck features from these front-ends are then combined and used for ASR experiments.

2.3.1. *DNN pre-training with multilingual data*

A multilingual speech corpus consisting of 16 hours of Spanish, 15 hours of German and 1 hour of English is used to train a 5 layer multilingual DNN network following the procedure described in Sec. 2.1. Training is performed using a combined phoneme set size of 52 derived from a count-based mapping [8].

Two DNNs are trained on different feature representations. The first network is trained on a 9-frame context window of 39 dimensional PLP features (13 cepstral + Δ + $\Delta\Delta$ features). The network has 2 wide hidden layers and a bottleneck layer, resulting in an architecture of 351 x 1000 x 1000 x 25 x 52. The second system is trained on modulation features derived using FDLP. These features (FDLPM) correspond to 28 static and dynamic modulation frequency components extracted from 17 bark spaced bands. A reduced feature set from only 9 alternate odd bands is used to train a system with an architecture of 252 x 1000 x 1000 x 25 x 52. Both the systems are trained with the standard back propagation algorithm using a cross entropy error criterion. The learning rate and stopping criterion are controlled by the frame classification error on a cross validation data set.

2.3.2. *DNN adaptation to low-resource settings*

Each of the multilingual DNN networks is then adapted to the low-resource setting using 1 hour of English data. This is done by replacing the multilingual output layer of the DNNs with an output layer corresponding to the English phoneset (Sec. 2.2).

After initialization, PLP and FDLPM features derived from 1 hour of English are used to train the new low-resource networks. These networks have the same architecture as before, except that

Table 1. Word Recognition Accuracies (%) with different monolingual training configurations.

Amount of English training data	PLP features	DNN features
1 hour	28.8	31.2
15 hours	46.5	49.7

Table 2. Word Recognition Accuracies (%) with different multilingual training configurations

Network Configuration	DNN features
MLP with random initialization followed by multilingual training + update with 1h of English	37.2
DNN with random initialization followed by multilingual training + update with 1h of English	40.7
DNN with discriminative pre-training followed by multilingual training + update with 1h of English	41.0

they now have a 47-dimensional English-specific output layer. These networks are then used to derive bottleneck features. The 2 sets of 25-dimensional bottleneck features from each of the networks are concatenated before applying a dimensionality reduction to form the final 25-dimensional bottleneck feature vector for speech recognition [11].

2.3.3. Experiments using DNN features

Table 1 shows the recognition accuracies using conventional PLP features and the DNN based features derived using monolingual data. Separate baseline systems are trained with 1 hour of English (similar to a low-resource setting) and with all the available 15 hours of transcribed data. Although DNN based features perform better, their performance is still poor in low-resource settings.

Experiments in Table 2 show that the performance gaps observed in Table 1 can significantly be reduced by utilizing multilingual training data. Additional improvements are obtained by increasing the number of hidden layers from two in [11] to three layers in the current approach and by pre-training the DNN in the multilingual training stage. In separate experiments, placing the bottleneck layer in the center as used in [18] did not provide any gains.

These experiments demonstrate how the performance of the DNN-based front-end can be improved by augmenting the one hour of English data with data from other languages. However, the recognizer was still trained on just the one hour of available transcribed English. In the next section, we show how semi-supervised training can be used to generate additional transcribed training data for both the acoustic model and front end.

3. SEMI-SUPERVISED TRAINING

Semi-supervised training has been effectively used to train acoustic models in several languages and conditions [32, 33, 34, 35, 36]. This section discusses the application of these approaches to low-resource settings. We start by using a baseline decoder (the best front-end and acoustic model we have so far) to generate recognition hypotheses for any available untranscribed training data. The most reliable of these estimated transcriptions are then combined with the limited existing transcribed training data to train both of the DNN front-end and GMM-HMM acoustic models in a semi-supervised fashion.

3.1. Selecting reliable data

In low-resource settings, it is important to select reliable outputs from the baseline decoder, since the quality of the outputs vary considerably from quite good (should be used) to poor (should be excluded). We use a selection that uses a hybrid combination of two confidence scores.

3.1.1. ASR-based word confidence scores

ASR lattice outputs can be treated as directed graphs with arcs representing hypothesized words. Each arc spans a duration of time (t_s, t_f), that the word is hypothesized to be present in the speech signal and is associated with acoustic and language model scores. Using these scores, word posterior probabilities can be computed using the standard forward-backward algorithm [37]. For any given hypothesized word w_i, at a given time frame t, several instances of the word can be present on different lattice arcs simultaneously. A frame-based word posterior of w_i can be computed as

$$p(w_i|t) = \sum_j p(w_i^j|t) \tag{1}$$

where j corresponds to all the different instances of w_i that are present at time frame t [38]. In our proposed selection technique we use a word confidence measure C_{max} based on these frame level word posteriors [38], given as the maximum word confidence of the word in its hypothesized time interval (t_s, t_f)

$$C_{max}(w_i, t_s, t_f) = \max_{t \epsilon (t_s, t_f)} p(w_i|t) \tag{2}$$

3.1.2. MLP posteriogram-based phoneme occurrence confidence

In addition to the word confidence scores from the speech recognizer, we also derive confidences scores from phoneme posterior outputs of a neural network classifier. This confidence measure uses a posteriogram representation of an utterance, derived by passing the acoustic features corresponding to the utterance through the trained DNN front-end classifier. For each hypothesized word w_i in the ASR transcripts, we first look up its set of constituent phonemes $\{p_1, p_2 \dots p_n\}$ from a pronunciation lexicon. Posteriors corresponding to each phoneme are then selected for the utterance's posteriogram representation and binarized to indicate the phoneme's presence or absence using a set threshold. The average number of times the constituent phonemes appear in the hypothesized time span (t_s, t_f) along a Viterbi search path is then used as confidence measure. The selected path is designed to produced an occurrence count while visiting all constituent phonemes in sequence. The rationale behind this measure is that if a word is hypothesized correctly, it is likely that all its constituent phonemes will be present in the posteriogram, hence resulting in a high average occurrence count. The proposed count-based measure is computed as

$$C_{occ}(w_i, t_s, t_f) = \frac{c}{N} \tag{3}$$

where c is the total number of times phoneme occurrences and N is the total number of frames in the hypothesized interval (t_s, t_f).

A logistic regression is used to combine the two confidence measures into a single hybrid confidence score. The regressor is trained to predict a combined confidence using word confidence and phoneme occurrence confidence scores using a held out data set.

Table 3. Word Recognition Accuracies (%) at different word confidence thresholds on a held-out set

Threshold	Acc (%)	Threshold	Acc (%)
None	38.75	+ 0.2	44.0
- 0.1	39.5	+ 0.3	45.5
+ 0.0	41.7	+ 0.4	45.4
+ 0.1	42.7	+ 0.5	44.6

3.2. Experiments and results

For our experiments in low-resource settings, we use a randomly selected 1 hour of transcribed data from the complete 15 hour Callhome English data set covering all speakers. In our semi-supervised training experiments we consider the remaining 14 hours as untranscribed data and attempt to use it.

3.2.1. Data selection

Using the ASR system trained with features from the multilingual DNN front-end, the 14 hour set of untranscribed data is decoded. Word lattices are generated during the decoding process and used to generate confidence scores for each hypothesized word, as described above. The multilingual DNN front-end is also used to produce phoneme posterior outputs from which phoneme occurrence-based confidence scores are derived. Combination weights for these confidence scores are then estimated by training a logistic regressor on a 45 minute held-out data set with the set's ground truth transcriptions.

After every hypothesized word in the decoded output has been given a score using the trained logistic regression module, each utterance is assigned an utterance-level score. This utterance level score is the average of all word-level scores in the utterance.

To evaluate the usefulness of the proposed confidence selection scheme we generate utterance level scores for the held out data. The recognition accuracy is then evaluated on selected sentences at different threshold levels. Table 3 shows the word recognition accuracies at different thresholds on the held out set. As the threshold increases, fewer reliable sentences get selected.

3.2.2. Selective semi-supervised training of DNNs

The initial multilingual DNN training experiments described earlier were based on only 1 hour of transcribed data. For semi-supervised training of DNNs we include additional data with noisy transcripts. These utterances are selected from the untranscribed data based on their utterance level confidences. To avoid detrimental effects from noisy semi-supervised data during discriminative training of neural networks, we make the following design choices -

(a) During back-propagation training, the semi-supervised data is de-weighted. This is done by multiplying the cross-entropy error with a small multiplicative factor during training.

(b) The semi-supervised data is used only in the final pre-training stage after all the layers of the DNN have been created.

(c) Only a limited amount of selected semi-supervised data is added.

For our experiments we select about 4.5 hours of data using utterances with a score of 0.3 and greater. This data is then combined with the multilingual pre-training data set of 15 hours of German, 16 hours of Spanish and 1 hour of English. During the DNN training, we use a multiplicative factor of 0.3 to de-weight the cross-entropy error from the semi-supervised data.

Table 4. Word Recognition Accuracies (%) with semi-supervised pre-training. ASR models are trained on *1hr-Eng-all-spks* in both cases.

System	Acc (%)
Multilingual pre-training	41.0
Multilingual pre-training with selected semi-supervised data	42.7

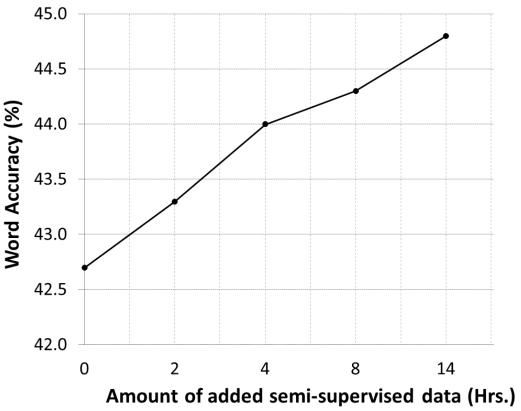

Fig. 1. Word Accuracy (%) improves with more and more semi-supervised data

The semi-supervised data is used in the final pre-training stage (Sec. 2.3.1, step 3) to train both the DNN networks using PLP and FDLPM features (Sec. 2.3). After pre-training, both the networks are adapted with 1 hour of English as before. Bottleneck features from both the networks are combined and used to train the low-resource ASR system with 1 hour of data. Table 4 shows the performance of the system after using semi-supervised data.

3.2.3. Semi-supervised training of acoustic models

Features from the DNN front-end with semi-supervised data are used to extract data-driven features for semi-supervised training of the ASR system. Similar to the weighing of semi-supervised data during the DNN training, we also use a simple corpus weighing while training the ASR systems. This is done by adding the 1 hour of fully supervised data with accurate transcripts twice.

To understand the effect of the semi-supervised data, we evaluate the recognition performance using different amounts of semi-supervised data. We observe that as we double the amount of semi-supervised data, there is close to a 0.5% increase in performance (Fig. 1). With semi-supervised training, the performance (44.8%) becomes comparable with the performance using conventional features (46.5%) on all the transcribed training data (Table 1).

4. CONCLUSIONS

This paper describes how complex neural networks classifiers can be built in low resource settings, using multilingual data and semi-supervised training. Semi-supervised training is used for training both neural network front-ends as well as acoustic models. We observe an absolute improvement of 16% in a low resource setting with only 1 hour of transcribed training data. Close to three-fourths of this gain come from DNN-based discriminative features and the remaining gains come from semi-supervised training.

5. REFERENCES

[1] H. Lin, L. Deng, D. Yu, Y. Gong, A. Acero, and C.H. Lee, “A study on multilingual acoustic modeling for large vocabulary ASR,” in *IEEE ICASSP*, 2009.

[2] D. Imseng, J. Dines, P. Motlicek, P.N. Garner, and H. Bourlard, “Comparing different acoustic modeling techniques for multilingual boosting,” in *ISCA Interspeech*, 2012.

[3] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, et al., “Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models,” in *IEEE ICASSP*, 2010.

[4] L. Lu, A. Ghoshal, and S. Renals, “Regularized subspace Gaussian mixture models for cross-lingual speech recognition,” in *IEEE ASRU*, 2011.

[5] Y. Qian, D. Povey, and J. Liu, “State-level data borrowing for low-resource speech recognition based on subspace GMMs,” in *ISCA Interspeech*, 2011.

[6] S. Sivadas and H. Hermansky, “On use of task independent training data in tandem feature extraction,” in *IEEE ICASSP*, 2004.

[7] A. Stolcke, F. Grézl, M.Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, “Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons,” in *IEEE ICASSP*, 2006.

[8] S. Thomas, S. Ganapathy, and H. Hermansky, “Cross-lingual and multistream posterior features for low resource LVCSR systems,” in *ISCA Interspeech*, 2010.

[9] Y. Qian and J. Liu, “Cross-lingual and ensemble MLPs - Strategies for low-resource speech recognition,” in *ISCA Interspeech*, 2012.

[10] N. Thang, B. Wojtek, F. Metze, and T. Schultz, “Initialization schemes for multilayer perceptron training and their impact on ASR performance using multilingual data,” in *ISCA Interspeech*, 2012.

[11] S. Thomas, S. Ganapathy, and H. Hermansky, “Multilingual MLP features for low-resource LVCSR systems,” in *IEEE ICASSP*, 2012.

[12] S. Thomas, S. Ganapathy, A. Jansen, and H. Hermansky, “Data-driven posterior features for low resource speech recognition applications,” in *ISCA Interspeech*, 2012.

[13] F. Seide, G. Li, X. Chen, and D. Yu, “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” in *IEEE ASRU*, 2011.

[14] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” in *IEEE Signal Processing Magazine*, 2012.

[15] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent, and S. Bengio, “Why does unsupervised pre-training help deep learning?,” *JMLR*, 2010.

[16] D. Yu, L. Deng, and G. Dahl, “Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition,” in *NIPS Workshop*, 2010.

[17] G.E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE TASLP*, 2012.

[18] D. Yu and M.L. Seltzer, “Improved bottleneck features using pretrained deep neural networks,” *ISCA Interspeech*, 2011.

[19] T.N. Sainath, B. Kingsbury, and B. Ramabhadran, “Autoencoder bottleneck features using deep belief networks,” in *IEEE ICASSP*, 2012.

[20] T.N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A.R. Mohamed, “Making deep belief networks effective for large vocabulary continuous speech recognition,” in *IEEE ASRU*, 2011.

[21] G.E. Hinton, S. Osindero, and Y.W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, 2006.

[22] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” *Advances in neural information processing systems*, 2007.

[23] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocky, “Probabilistic and bottle-neck features for lvcsr of meetings,” in *IEEE ICASSP*, 2007.

[24] B. Chen, Q. Zhu, and N. Morgan, “Learning long-term temporal features in lvcsr using neural networks,” in *ISCA ICSLP*, 2004.

[25] A. Canavan, D. Graff, and G. Zipperlen, “Callhome american english speech,” *LDC*, 1997.

[26] A. Canavan, D. Graff, and G. Zipperlen, “Callhome german speech,” *LDC*, 1997.

[27] A. Canavan and G. Zipperlen, “Callhome spanish speech,” *LDC*, 1997.

[28] J. Godfrey, E. Holliman, and J. McDaniel, “Switchboard: Telephone speech corpus for research and development,” in *IEEE ICASSP*, 1992.

[29] D. Graff, J. Kong, K. Chen, and K. Maeda, “English gigaword,” *LDC*, 2003.

[30] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *The Journal of the Acoustical Society of America*, 1990.

[31] S. Thomas, S. Ganapathy, and H. Hermansky, “Phoneme recognition using spectral envelope and modulation frequency features,” in *IEEE ICASSP*, 2009.

[32] G. Zavaliagkos, M. Siu, T. Colthurst, and J. Billa, “Using untranscribed training data to improve performance,” in *ISCA ICSLP*, 1998.

[33] T. Kemp and A. Waibel, “Unsupervised training of a speech recognizer: Recent experiments,” in *ISCA Eurospeech*, 1999.

[34] L. Lamel, J.L. Gauvain, and G. Adda, “Unsupervised acoustic model training,” in *IEEE ICASSP*, 2002.

[35] J. Ma, S. Matsoukas, O. Kimball, and R. Schwartz, “Unsupervised training on large amounts of broadcast news data,” in *IEEE ICASSP*, 2006.

[36] S. Novotney and R. Schwartz, “Analysis of low-resource acoustic model self-training,” in *ISCA Interspeech*, 2009.

[37] T. Kemp and T. Schaaf, “Estimating confidence using word lattices,” in *ISCA Eurospeech*, 1997.

[38] L. Burget, P. Schwarz, P. Matejka, M. Hannemann, A. Rastrow, C. White, S. Khudanpur, H. Hermansky, and J. Cernocky, “Combination of strongly and weakly constrained recognizers for reliable detection of OOVs,” in *IEEE ICASSP*, 2008.