Improving Semi-supervised Deep Neural Network for Keyword Search in Low Resource Languages

Roger Hsiao, Tim Ng, Le Zhang, Shivesh Ranjan,
Stavros Tsakalidis, Long Nguyen and Rich Schwartz

Raytheon BBN Technologies
10 Moulton Street, Cambridge, MA 02138, USA
whsiao@bbn.com

Abstract

In this work, we investigate how to improve semi-supervised DNN for low resource languages where the initial systems may have high error rate. We propose using semi-supervised MLP features for DNN training, and we also explore using confidence to improve semi-supervised cross entropy and sequence training. The work conducted in this paper was evaluated under the IARPA Babel program for the keyword spotting tasks. We focus on the limited condition where there are around 10 hours of supervised data for training.

Index Terms: semi-supervised training, deep neural network, keyword search

1. Introduction

Deep neural network (DNN) has shown to be successful in acoustic modeling [1, 2]. As shown in previous studies [1, 3], DNN's modeling capability can scale up as the amount of data increases. Advances in hardware and optimization algorithms also allow DNN to handle large data [4, 5]. While DNN is becoming the mainstream of large scale speech recognition systems [3], recent studies are also exploring the application of DNN in low resource languages [6, 7].

In some low resource languages, only a small amount of supervised data is available, say one to ten hours of audio data. This limited amount of audio data may not be enough to build an accurate acoustic model. One approach to tackle this issue is semi-supervised training [3, 8, 9], which uses a bootstrap system to automatically transcribe some unsupervised data. The unsupervised data is then pooled with the supervised data for the final training. Semi-supervised techniques has been widely used in Gaussian mixture model (GMM) based systems [8, 9]. In this paper, we investigate semi-supervised techniques that may improve DNN performance in low resource languages.

Semi-supervised training of DNN is relatively new and techniques based on confidence methods are showing positive results [6, 7, 10]. In [6], a confidence measure is used to select a subset of unsupervised data for training. The confidence model is frame based and the confidence score is the state posterior probability in the training lattices. In [7], DNN is used as a feature extraction front-end. A confidence model is used for data selection before semi-supervised training. Using multiple systems for confidence measure is proposed in [10]. Confidence scores from different systems are re-calibrated and the final scores are used for data selection.

In this paper, we propose methods to integrate confidence scores in semi-supervised DNN training. The confidence scores are used not only in the cross entropy training, but also in sequence training [11, 12]. We also investigate techniques that may help semi-supervised sequence training. We show that even though some methods may not have much impact in reducing the word error rate (WER), they improve the quality of lattices and also the performance of keyword search. This work is evaluated under the IARPA Babel program for the keyword spotting tasks. We focus on the limited condition, for which there are around 10 hours of transcribed audio data and 90 hours of untranscribed audio for a low resource language.

This paper is organized as follows: In section 2, we describe our confidence model and the training procedure of our semi-supervised system. In section 3, we propose our semi-supervised DNN training and discuss issues in DNN sequence training. Section 4 has experimental results and we conclude our work in section 5.

2. Semi-supervised training using confidence based methods

Our confidence model is a generalized linear model (GLM) [13] which computes a confidence score for each utterance. Using a bootstrap system, we decode some held out data and extract features such as acoustic score per word, pronunciation score per word, duration, signal to noise ratio, language model perplexity, nbest posterior, word-level confidence, etc. Using these features, we can train a GLM,

$$c_i = x_i\beta + \epsilon_i \quad (1)$$

where c_i is the confidence of utterance i or the label during training; x_i is the feature vector containing the features we mentioned; β is a vector to be estimated using linear regression and ϵ_i are zero mean stochastic disturbances assuming to be normal distributed with a constant variance. In our confidence model, some features are word-level confidence scores so they are used in another GLM trained with similar features.

Given this GLM, we can process the unsupervised data and assign a confidence score to each utterance. Based on the scores, we can perform data selection and weighted training. For weighted training, the confidence scores are first scaled and shifted by,

$$w_i = s \times c_i + b \quad (2)$$

where w_i is the weight for utterance i; s is the slope; c_i is confidence score for utterance i and b is the bias. In this work, s is 2.0 and the average of the utterance-level weights is constrained to 1.0. Hence, $b = 1 - \frac{\sum_{i=1}^{N} s \times c_i}{N}$ with N being the total number of utterances.

Scaling and shifting the confidence scores is important for semi-supervised GMM systems. As mentioned in [8], it is to ensure the thresholds commonly used in GMM training are still meaningful. One example is the minimum number of frames required to build a codebook (a set of Gaussians shared by several states). If the confidence scores are not adjusted, the number of frames would be underestimated which may not be ideal for acoustic model training.

3. Semi-supervised DNN training

Similar to the work in [8], we use confidence weights in DNN training. These weights are used to scale the learning rates of stochastic gradient descent in both cross entropy and sequence training [11, 12]. The purpose is to discourage excessive changes to the DNN if the confidence of some unsupervised data is low, which implies the data may contain more errors. The weights are scaled and shifted to ensure that the expected utterance weights of the unsupervised data is one. If the weights are not scaled, it effectively decreases the learning rate and the DNN training may require more epochs. For the supervised data, we apply a fixed weight which needs to be tuned. This fixed weight is often larger than one so the DNN training can focus on the supervised data. In this work, we use the same weights for both cross entropy and sequence training.

Our semi-supervised DNN training is similar to regular DNN training. Our DNN initialization uses discriminative pretraining which starts with a shallow network. We randomly select around 20% of the training data and train the network for one epoch. Then, we add a layer with randomized weights to the DNN. This procedure repeats until the target number of layers is reached. Once the DNN is initialized, we perform cross entropy training with a mini-batch size of 256 frames. The base learning rate starts with 0.002 and if the improvement of frame accuracy on an held out set falls below 0.25% absolute, the base learning rate is reduced by half for each epoch. This scheme has shown to be effective in DNN training [14]. The actual learning rate used to update the DNN is derived by scaling the current base learning rate with the frame weight. The frame weight simply assumes all the frames in an utterance share the same utterance weight.

After cross entropy training, we perform sequence training on the semi-supervised data. Similar to the semi-supervised discriminative training described in [8], the numerator statistics of the unsupervised data are collected from the 1-best instead of the reference. Instead of using mini-batches, we perform one stochastic update for each utterance with a fixed base learning rate of 0.00001. Similarly, the actual learning rate is computed by scaling the base learning rate with the utterance weight.

3.1. Issues in semi-supervised sequence training

In practice, sequence training often requires a few heuristics to prevent some unwanted behaviors such as "run-away" silence. As discussed in [15], sequence training may continue to boost the likelihood of silence frames and create high deletion error. This corrupts the model even though the training continues to improve the objective function. This problem is partially due to the sparseness of the training lattices which fail to cover some possible competitors. For "run-away" silence, since silence arcs seldom appear in the training lattices as competitors, they do not contribute to the denominator statistics. As a result, the likelihood of silence is boosted as a side effect of the optimization trying to suppress other speech targets that occur in the lattices.

To handle this problem, [15] suggests adding artificial silence arcs to the training lattices. However, this does not completely solve the problem as other speech outputs may also suffer from similar issues. Hence, [15] proposes frame smoothing (f-smoothing). The idea is to fix the sparseness of the training lattices by incorporating the cross entropy function in the optimization. Instead of optimizing the discriminative objective function alone, the objective function becomes,

$$F = \lambda F^D + (1-\lambda) F^{CE} \,, \tag{3}$$

where F^D is the discriminative objective function; F^{CE} is the cross entropy function and λ is a tunable parameter to control the weight.

In this combined function, although the training lattices may not cover all the competitors, the cross entropy function covers all states and discourages any increase in likelihood except for states that correspond to the references. As a result, issues like "run-away" silence are prevented since the cross entropy function would discourage boosting the likelihood of the silence frames. This characteristic could help semi-supervised training when the auto transcript is wrong. Although the discriminative objective function may direct the model to a wrong direction, the cross entropy function would try to deactivate most of the state targets. As a result, it serves as a regularization mechanism for semi-supervised DNN training.

4. Experimental Results

4.1. System description and evaluation method

We evaluate our proposed semi-supervised DNN training in the Babel year two evaluation on development languages. The IARPA Babel program is a research program for rapid development of keyword spotting systems for low resource languages. In the second year of the program, Bengali, Assamese, Haitian Creole, Lao and Zulu are used as the development languages. The evaluation has different conditions and one of them is the limited condition, in which the training consists of 10 hours of transcribed audio and roughly 90 hours of unsupervised data. The audio data is mainly conversational speech between two persons over a telephone channel, but each language pack also comes with a small amount of read speech. The telephone channels can be landlines, different kinds of cellphones, or phones embedded in vehicles, and the sampling rate is 8000 Hz. The development set for each language consists of roughly 10 hours of conversational telephone speech. The evaluation set, given by IARPA, contains 15 hours of speech for each language. In this paper, we evaluate our approaches on the IARPA Babel Program Bengali language collection release (babel103b-v0.4b), and we report our results on the development set.

For keyword spotting, each language has two set of keywords: a development keyword list and an evaluation keyword list. The development keyword list contains around 2000 keywords which were selected by the performers for development. The evaluation keyword lists consists of 3000 to 4000 keywords, and they were given during the evaluation. Each keyword may contain several words and it may or may not be in the training vocabulary. The performance of a keyword spotting system is measured by the Actual Term Weighted Value (ATWV) and WER is also measured for the underlying STT system. ATWV is computed by,

$$ATWV = 1 - \frac{1}{K}\sum_{w=1}^{K}\left(\frac{\#miss(w)}{\#ref(w)} + \beta\frac{\#fa(w)}{T - \#ref(w)}\right) \tag{4}$$

where K is the number of keywords; $\#miss(w)$ is the number of true keyword tokens that are not detected; $\#fa(w)$ is the number of false alarms; $\#ref(w)$ is the number of words in reference; T is the number of trials (e.g., seconds in the audio), and β is a constant set at 999.9. The details and the design of this metric are available in [16].

The BBN keyword spotting system is divided into several components. At a high level, the speech recognition system [8, 17] is run to produce a detailed lattice of word hypotheses. This lattice is used to extract keyword hits with nominal posterior probability scores produced by various methods. Different extraction methods are necessary because, for example, we can use whole-word extraction methods for the known keywords but we must use phonetic extraction for the keywords that were not known when the recognizer was run. Also, multiple extraction methods help the system to be more robust for different languages. The scores are normalized so that they are consistent across keywords and so that they are good estimates of posteriors. Details of score normalization are available in [18].

For the semi-supervised system, we first built a system using the MLP features trained on the 10-hr supervised data. This system, trained solely on supervised data, is used as the baseline system and used to transcribe the unsupervised data. A confidence model is then built to select data with over 50% confidence. The selected data is then pooled with the supervised data for final training. A GMM based system is then built on the semi-supervised data set according to our previous work as described in [8]. Figure 1 is an overview of our data selection and confidence weighted training.

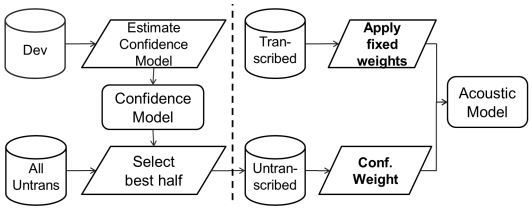

Figure 1: Overview of the confidence weighted training

The BBN GMM system is trained on the semi-supervised MLP features provided by Brno University of Technology [19]. We use region dependent feature transformation to combine with regular PLP features and reduce the feature dimension to 46 [17]. These final features are also used to train our semi-supervised DNN systems, and the forced alignment for DNN training is computed by the GMM system.

4.2. Confidence weighted cross entropy training

Similar to the work in [8], the utterance weights of the unsupervised data are computed by the methods in section 2. For supervised data, we assign a fixed weight larger than 1.0. Table 1 shows the performance of the baseline DNN using only the supervised data and different semi-supervised DNN systems after cross entropy training. The keyword search results are based on the development keyword list. It is important to note that even though the baseline DNN is only trained on the supervised data, it is still a semi-supervised system since the MLP features are semi-supervised.

From the results in table 1, using confidence weights in cross entropy training slightly reduces the WER. This observation is the similar to the experiments reported in [8], where using confidence weights would give small improvement in WER.

System	sup. : unsup.	WER(%)	ATWV(%)
10-hour baseline	-	66.30	-
semi-supervised	unweighted	62.59	35.58
semi-supervised	2 : x	62.88	36.09
semi-supervised	3 : x	62.20	36.76
semi-supervised	4 : x	62.13	36.35

Table 1: WER and ATWV of the Bengali DNN systems using confidence weighted cross entropy training

However, even though the improvement in WER is small, the improvement in ATWV is over 1.0% absolute. Therefore, this technique is still valuable to a keyword search system.

4.3. F-smoothing in semi-supervised sequence training

Based on the best DNN from cross entropy training, we perform sequence training and study the effect of f-smoothing on semi-supervised training. Table 2 shows the results of performing sequence training on either supervised data only or semi-supervised data, and with or without f-smoothing. When f-smoothing is enabled, we choose $\lambda = 0.9$ in equation 3 which is the suggested setting in [15].

System	train set	F-smth.	WER(%)	ATWV(%)
Xent.	-	-	62.20	36.76
seq. trn.	sup.	no	61.67	37.36
seq. trn.	sup.	yes	62.18	37.18
seq. trn.	semi	no	63.16	34.37
seq. trn.	semi	yes	61.51	37.97

Table 2: WER and ATWV of the Bengali DNN systems using semi-supervised sequence training

The results show that sequence training could help when it is trained on the supervised data alone. This observation is also mentioned in [12]. When sequence training is applied to semi-supervised data, f-smoothing becomes crucial. This supports our argument in section 3.1 that f-smoothing would help semi-supervised training. Using semi-supervised sequence training with f-smoothing also gives better performance compared to using sequence training on supervised data alone.

4.4. Confidence weighted sequence training

Using the best DNN from cross entropy training, we investigate whether sequence training can also benefit from confidence weights. In this experiment, we used the same weights as in the cross entropy experiments, and f-smoothing was enabled. Table 3 shows the performance of confidence weighted sequence training.

System	sup. : unsup.	WER(%)	ATWV(%)
Xent.	-	62.20	36.76
seq. trn.	unweighted	61.51	37.97
seq. trn.	3 : x	61.88	38.53

Table 3: WER and ATWV of the Bengali DNN systems using confidence weighted sequence training

The results show that although using confidence weights in

semi-supervised sequence training may not reduce WER, they can still improve the keyword search performance. In our work, the relative improvement on WER is small for sequence training but it is due to two factors. The first reason is that we are using semi-supervised MLP features, so the baseline system is already benefited by neural network training, and the second reason is that our sequence training is semi-supervised. Nonetheless, the improvement on keyword search is encouraging in that we improved from 35.58% to 38.53% ATWV.

Table 4 summarizes the improvement of our Bengali DNN system using the techniques covered in this paper. Compared to the semi-supervised GMM system developed using the procedure in [8], our semi-supervised DNN system is competitive.

System	WER(%)	ATWV(%)
10-hour baseline	66.30	-
semi-supervised baseline	62.59	35.58
+ conf. weighted Xent.	62.20	36.76
+ seq. training w/ f-smoothing	61.51	37.97
+ conf. weighted seq. training	61.88	38.53
semi-supervised GMM w/ MLP	61.64	36.78

Table 4: Performance of our Bengali semi-supervised GMM and DNN systems

5. Conclusions

In this paper, we explore techniques that improve semi-supervised DNN training. We focus on confidence based methods and issues in sequence training. We show that, by using confidence weights in cross entropy training and sequence training, and also with f-smoothing, we can improve the WER from the baseline 66.30% to 61.88%. For keyword search performance, we improve from 35.58% ATWV for the baseline semi-supervised system to 38.53% ATWV. These results also imply that WER may not be a good predictor of keyword search performance. Even though some techniques may give only mild improvement in WER or even a small degradation, they can still improve keyword search performance.

6. Acknowledgements

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

7. References

[1] G. Hinton, L. Deng, D. Yu, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, G. Dahl, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[3] H. Liao, E. McDermott, and A. Senior, "Large Scale Deep Neural Network Acoustic Modeling with Semi-supervised Training Data for YouTube Video Transcription," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.

[4] V. Vanhoucke, A. Senior, and M. Mao, "Improving the Speed of Neural Networks on CPUs," in *Proceedings of the NIPS*, 2011.

[5] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large Scale Distributed Deep Networks," in *Proceedings of the NIPS*, 2012.

[6] K. Veselý, M. Hannemann, and L. Burget, "Semi-supervised Training of Deep Neural Networks," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 267–272.

[7] S. Thomas, M. Seltzer, K. Church, and H. Hermansky, "Deep Neural Network Features and Semi-supervised Training for Low Resource Speech Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 6704–6708.

[8] R. Hsiao, T. Ng, F. Grézl, D. Karakos, S. Tsakalidis, L. Nguyen, and R. Schwartz, "Discriminative Semi-supervised Training for Keyword Search in Low Resource Languages," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.

[9] K. Yu, M. Gales, L. Wang, and P. Woodland, "Unsupervised Training and Directed Manual Transcription for LVCSR," *Speech Communications*, vol. 52, no. 7–8, pp. 652–663, 2010.

[10] Y. Huang, D. Yu, Y. Gong, and C. Liu, "Semi-Supervised GMM and DNN Acoustic Model Training with Multi-system Combination and Confidence Re-calibration," in *Proceedings of the INTERSPEECH*, 2013.

[11] B. Kingsbury, "Lattice-based Optimization of Sequence Classification Criteria for Neural Network Acoustic Modeling," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2009.

[12] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative Training of Deep Neural Networks," in *Proceedings of the INTERSPEECH*, 2013.

[13] P. McCullagh and J.A. Nelder, *Generalized Linear Model*. London: Chapman and Hall, 1989.

[14] A. Senior, G. Heigold, M. Ranzato, and K. Yang, " An Empirical Study of Learning Rates in Deep Neural Networks for Speech Recognition ," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 6724–6728.

[15] H. Su, G. Li, D. Yu, and F. Seide, "Error Back Propagation for Sequence Training of Context-Dependent Deep Networks for Conversational Speech Transcription," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 6664–6668.

[16] "OpenKWS13 Keyword Search Evaluation Plan," http://www.nist.gov/itl/iad/mig/upload/OpenKWS13-EvalPlan.pdf, 2013.

[17] T. Ng, B. Zhang, S. Matsoukas, and L. Nguyen, "Region Dependent Transform on MLP Features for Speech Recognition," in *Proceedings of the INTERSPEECH*, 2011.

[18] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grézl, M. Hannemann, M. Karafiát, I. Szoke, K. Veselý, L. Lamel, and V.-B. Le, "Score Normalization and System Combination for Improved Keyword Spotting in Speech," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.

[19] F. Grézl and M. Karafiát, "Semi-Supervised Bootstrapping Approach for Neural Network Feature Extractor Training," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.