Recent Improvements in Neural Network Acoustic Modeling for LVCSR in Low Resource Languages

Jia Cui[1], *Bhuvana Ramabhadran*[1], *Xiaodong Cui*[1],
Andrew Rosenberg[2], *Brian Kingsbury*[1], *Abhinav Sethy* [1]

[1]IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
[2]Department of Computer Science, Queens College, City University of New York, NY, USA
{jiacui, bhuvana, cuix, bedk, asethy}@us.ibm.com, andrew@cs.qc.cuny.edu

10.21437/Interspeech.2014-213

Abstract

In this paper we focus on several techniques that improve deep neural network (DNN) acoustic modeling for low-resource languages. We explore the use of different features such as, fundamental-frequency variation (FFV), tonal features, and normalization of these features for deep neural network training. Specifically we study the impact of these features in conjunction with a tonal lexicon and several neural network architectures including hybrid and bottleneck feature-based configurations. We also explore the use of un-transcribed data and ways to balance it with transcribed data, to enhance the performance of the best performing LVCSR system. Results are presented in the context of the IARPA Babel program on development languages from Babel option period as well as on the surprise language from the base period of the program. We show that these improved methods can provide up to 15% relative reduction in WER and improvements in keyword search, in the languages explored under the BABEL program.

Index Terms: Neural Network architectures, Fundamental Frequency Variation (FFV), Un-transcribed data, I-vectors, Tonal modeling

1. Introduction

In recent years, Deep Neural Networks (DNNs) have become a popular acoustic modeling technique over the last few years, showing significant gains over Gaussian Mixture Model/Hidden Markov Model (GMM/HMM) systems on small, medium and large vocabulary speech recognition tasks.

Speaker-adapted (SA) and discriminatively trained-features (DT), learned via objective functions such as feature-space maximum likelihood linear regression (fMLLR), and feature-space Minimum Phone Error (fMPE). are now the de facto standard in most LVCSR systems. These features are often augmented with tonal features such as pitch, and other vocal tract related features to capture the characteristics of the language. Deep Neural Networks have been shown to do feature extraction jointly with classification, and have thus been used as feature extractors. Bottleneck features derived from the hidden layers of a neural network [1, 2, 3, 4] have been proposed in the literature to obtain better input representations with deep and stacked neural network architectures. In this work, we present a detailed study of several input features used to train a DNN, as well the deep neural network architectures used for extracting bottleneck features in the context of the IARPA BABEL program.

The IARPA Babel Program simulates a low resource language by collecting data in a limited time from a new language, comprising of scripted and conversational speech. The Limited Language Pack (LLP) scenario is one where only one tenth of the total data (referred to as Full Language Pack (FLP)) is used for training. However, the rest of the audio can be used as un-transcribed data, thus lending itself to the use of several semi-supervised learning methods. The three main questions we attempt to address in this paper include:

- Input Features: What is the input feature representation for both GMMs and DNNs that will yield the lowest Word Error Rate (WER)?
- Modeling: What is the DNN architecture that can take advantage of these input features and result in the lowest overall WER at the end of the training pipeline? While each of the input features and DNN structure explored may provide gains at various stages, these may not hold to the very end of the training process. What is a good training recipe that will yield the best performing ASR system?
- Generalization: Do the gains from these features and model architectures hold across other languages in the BABEL program?

We begin by answering these questions on Vietnamese, the surprise language from the BABEL program's base period. In Section 2, the impact of features including speaker-adapted, discriminatively trained features, tonal features such as pitch and Fundamental Frequency Variation (FFV) features and I-vectors are presented. Section 3 explores DNN architectures and concludes with the training methodology that yields the lowest WER on Vietnamese. The generalizability of this recipe is demonstrated in Section 4 with languages such as, Assamese, Bengali, Haitian Creole and Zulu. We summarize and conclude with the main messages in Section 5.

2. Feature Representations

The IBM LVCSR GMM/HMM system is trained using the IBM Attila speech recognition toolkit and follows the recipe described in [5]. This system bootstraps the DNNs [6, 7].The Speaker-Adaptively trained (SAT) model uses the bootstrap and restructuring procedure [8] to produce stable models from small training corpora. DT models, built on top of this SAT model include feature space and model space discriminative training using the Minimum Phone Error (MPE) rate criterion. The feature processing pipeline computes 13-dimensional PLP features with speaker-based mean and variance normalization and Vocal Tract Length Normalization (VTLN), a context of 9 frames projected down to a 40-dimensional feature space using linear

discriminant analysis (LDA), and further de-correlated with a global, semi-tied covariance (STC) transform. These features comprise our *baseline feature set* for both GMM and DNN acoustic models in any language.

2.1. Vietnamese Baseline GMM Models

The Vietnamese Limited Language Pack contains 22 hours of conversational speech and 2 hours of scripted speech, however, more than half of this data is silence, leaving approximately 7 hours of real speech for acoustic model training. The transcribed data contains 11204 sentences with 122K word tokens and results in a 3.1K lexicon which contains words in the training data only. The development data contains includes 10 hours of speech, 9466 sentences with 114K words. The Out-of-Vocabulary (OOV) rate is 1.5% and a modified Kneser-Ney smoothed tri-gram language model results in a perplexity of 170 on this test set. The WER of the GMM models with the baseline feature set are presented in Table 1. In roder to obtain better quality alignments, a speaker independent (SI) DT model is built using the SI features, referred to as fMPE.SI in Table 1 (row 2). Each set of features over the baseline set provides gains resulting in the final baseline GMM system, the SAT-based DT model with a WER of 70.7%.

Model	WER
PLP+LDA+STC	81.0
fMPE.SI	76.3
VTLN	79.1
SAT	76.8
fMPE.SA	70.7

Table 1: *Performance of the Vietnamese Baseline GMMs*

2.2. Vietnamese Baseline DNN Models

The baseline DNN contains 5 hidden layers with 1K sigmoid units per layer, and has a final softmax output with 1,500 targets. Training occurs in three phases: (1) layer- wise discriminative pre-training using the cross-entropy (CE) criterion, (2) stochastic gradient descent optimization using the cross-entropy criterion, and (3) distributed Hessian-free training using the state-level minimum Bayes risk criterion (sMBR) [2].

Three different features from the baseline feature set were used to train DNNs. These are: (1) 40 dimensional, SI LDA features (PLP+STC+LDA), (2) speaker adapted features (SAT), and (3) DT trained SAT features from the baseline feature set concatenated with log-mel features (fMPE.SA+logmel). All these features include 9 frames of context and were concatenated with delta and double-delta features but with no context expansion. The target labels for the SI features is the SI GMM states while the target labels for the remaining two sets are from the VTLN-based GMM. The target labels correspond to 1500 context-dependent states. Table 2 illustrates the performance of DNNs with these varied input features for both, CE and sMBR training criteria. The column titled *Alignment Model* indicates the type of GMM model that was used to obtain targets for the DNNs. Regardless of the input feature type, sequence training yields an additional $3-5\%$ relative reduction in WER over DNNs trained with the CE objective function. The last column in the table indicates that in each case, further improvements of $2-3\%$ relative can be obtained with a second pass of training, if the first pass DNN models are used to re-align the training data to regenerate the frame-wise labels. The discriminatively trained, speaker dependent features produce the best improvement, a 7% absolute WER reduction over the baseline. While the speaker dependent DNN improves significantly over the SAT model, its performance is very similar to the baseline fMPE.SA GMM model in Table 1. The last row in Tabletb:basicDNN uses a feature space obtained by the concatenation of these DT trained features with log-mel features that are obtained by passing an FFT through a mel-filterbank, followed by a non-linear log operation. DNNs with sufficiently large number of parameters can capture variance in the signal. This is demonstrated by these features yielding the best baseline DNN WER of 69.3%.

Features	Alignment Model	Training Objective	WER	WER w/ re-alignment
PLP+LDA+STC	SI	CE	79.0	79.8
PLP+LDA+STC	SI	sMBR	75.3	74.0
SAT	VTLN	CE	74.5	72.9
SAT	VTLN	sMBR	71.5	70.0
fMPE.SA+logmel	VTLN	CE	73.9	71.5
fMPE.SA+logmel	VTLN	sMBR	71.6	69.3

Table 2: *Vietnamese Baseline DNN*

2.3. Feature Space Improvements

2.3.1. Channel-aware features

Channel-aware features were originally proposed in [9] to serve as a means of robustness to channel variation seen in wireless telephone channels. Table 3 illustrates the impact of these features when used in conjunction with the DT-transformed, baseline features (fMPE.SA) and pitch estimated using the YIN algorithm [7]. The channel-aware features are represented by the mean of the PLP features and provide gains of up to 0.8% absolute when DNNs are trained with the CE objective function.

2.3.2. I-vector featue representations

I-vectors [10] are a popular technique for speaker verification and speaker recognition as they encapsulate all the relevant information about a speakers identity in a low-dimensional fixed-length representation [11]. As described in [12], we train two 2048 40-dimensional diagonal covariance GMMs for the SI and SAT feature sets to derive the statistics needed for I-vector estimation. I-vectors provided a gain of 0.9% absolute when used with the baseline feature set and DNNs were trained using the CE objective function. However, these gains disappeared after sMBR training in Vietnamese, therefore, we did not explore the use of these features further.

2.3.3. Fundamental Frequency Variation

One limitation to extracting and representing pitch information from speech is that pitch is not always present in the acoustic signal. In speech, the vocal folds only vibrate, and thereby generate pitch, during voiced phones. When producing unvoiced phones, and during silence, pitch is undefined. Thus pitch representations are forced to adopt one of two strategies: (1) interpolate over unvoiced regions, essentially hallucinating unobserved data, or (2) representing pitch in two orthogonal dimensions, one to represent the value of fundamental frequency, and another dimension to represent whether pitch is present or not, sometimes denoted as p(voicing).

Fundamental Frequency Variation (FFV) is a technique proposed to measure changes in fundamental frequency[13]. FFV

is (1) instantaneous, not relying on adjacent frames, (2) continuous, defined at all time instances, and (3) potentially sparse. This is accomplished by calculating the normalized dot product between two FFT spectra across a short region at different dilations of one or both spectra. The intuition is that if pitch is increasing the spectrum will dilate over time, i.e. the space between harmonics will increase. On the other hand, if pitch is decreasing the harmonics will contract. The FFV spectrum is a measure of this dot product (the magnitude of similarity) at a range values of a dilation parameter, τ. A visualization of this parameterized dilation can be found in Figure 1.

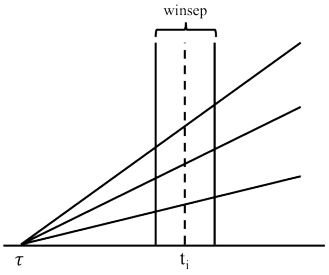

Figure 1: FFV dilation visualization

There are three windowing parameters in the FFV calculation: an external window, internal window, and window separation (winsep). The two FFT spectra are calculated from two asymmetric windows before and after a time, t_i. The window functions are defined by an external (away from t_i) half Hamming window and internal (toward t_i) half Hanning window with its peak at $t_i \pm winsep/2$.

Identifying the maximum value of the FFV spectrum provides a measure of instantaneous fundamental frequency variation. However, it is more common to apply a filter bank of five or seven filters to the FFV spectrum to yield a frame based measure of fundamental frequency variation. These filters correspond to stable pitch (the middle filter), increasing degrees of change in pitch as you move away from the middle filter. Recently a number of groups have demonstrated the utility of FFV as an input feature for speech recognition [14]. In order to align FFV frames with PLP frames for acoustic modeling, we use a 10ms frame size, an external window of 15ms, and winsep of 10 ms.

We explored the addition of FFV features to our best baseline input feature representations. Table 3 clearly demonstrates that the reliable estimation of changes in fundamental frequency is correlated well with improvements in the performance of the DNN systems (WER reduction of 1.4% absolute). The FFV features serve as a good replacement for the pitch feature estimated with the YIN algorithmr̃efbedk. In this table, the improved baseline WER of 69.4% corresponds to same baseline feature set as the last row in Table 4, but with an improved lexicon as described in Section 3.

Feature Space	WER
fMPE.SA+logmel	69.4
fMPE.SA+pitch+plp-mean	68.2
fMPE.SA+logmel+**FFV**+plp-mean	66.8

Table 3: *Impact of FFV features*

One explanation for FFV to improve ASR performance is that FFV represents a more reliable measure of change in pitch and by operating in a more parsimonious space, can be more reliably modeled. We also notice another property of the FFV spectrum which may inform its efficacy. Figure 2 includes the FFV spectrum obtained from a synthesized voice (Praat's parametric synthesizer) producing the phrase "This is". This is a low quality synthesizer, but has the desirable properties of having very simple harmonic structure, and no prosodic variation. The pitch in this utterance is nearly constant at 95±5Hz. If the FFV spectrum were only capturing change in fundamental frequency, there would be no differentiation across the utterance. However, we see that across all synthesized phones, the FFV spectrum is nearly flat within the middle five bands, not showing a peak at the middle band. Moreover, within each phone, the overall magnitude of the spectrum within these middle bands varies significantly. The brighter regions during the two /ɪ/ phones indicate greater FFV magnitude during these phones, while the darker regions occur during the fricatives /ð/, /s/ and /z/. We hypothesize that the FFV spectrum is capturing harmonic stability.

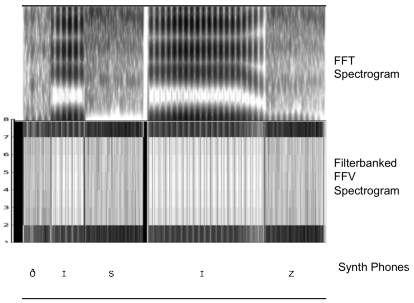

Figure 2: Synthesized speech FFT spectrogram and FFV spectrogram

3. Modeling

3.1. Dictionary and Tonal phones

On an average, the lexicon provided as part of the data, includes 1.5 different pronunciations for each Vietnamese word, with 74 different non-tonal phones. Given that Vietnamese is a tonal language, we subsequently switched to a different tonal dictionary graciously provided by one of the members of the BABEL program, BBN [15], which retains only one pronunciation for each word in the dictionary and appropriately maps phones to reduce the non-tonal phone set to 53.

The dictionary and tonal phones were used in conjunction with the best input feature representations from Table 2. The results are presented in Table 4. With just a reduced and carefully crafted lexicon, we see that an additional improvement of $0.7 - 0.9\%$ absolute is obtained with the baseline DNN system, and 2.7% absolute with the baseline GMM system (Compare columns 2 and 3 in Table 4). This lexicon in combination with additional tonal phones (resulting in a phone set of size 145), produces a WER of 66.9%.

3.2. Network Architectures

In the section, we explore two different DNN architectures. The first is the same as the baseline DNN, presented in Section 2.2, and the second is the cascaded stacked architecture presented in [3]. Similar to [3], we stacked two DNNs. The first DNN has 4 hidden layers of 1024 sigmoid units , except for the third layer, which has 80 sigmoid units and serves as the bottleneck layer.

Features	Baseline	New Lexicon	New Lexicon + tonal phones
SAT	76.8	74.1	73.4
fMPE.SA+logmel (CE)	71.5	70.6	69.4
fMPE.SA+logmel (sMBR)	69.3	68.6	66.9

Table 4: *WER with BBN dictionary and Tonal phones*

The input to this network are the best features as determined in Section 2, namely, the baseline feature set concatenated with log-mel and FFV features. The 80-dimensional bottleneck features were spliced together with a 10 frame context on each side and down-sampled to a 2 frame context on either side, resulting in a 400-dimensional input vector which is mean-normalized to train the second DNN. Both the DNNs are trained with CE and sMBR criteria. The second DNN is the exact configuration of the first DNN with the exception of 40 hidden units in the bottleneck layer. This 40-dimensional feature is subsequently used to train GMMs. Table 5 presents the improvements with the stacked architecture. We can see that the bottleneck configuration does indeed provide a 0.3% absolute gain over the hybrid stacked configuration. The hybrid stacked configuration is in itself 1% better than the hybrid baseline DNN configuration trained with the sMBR criterion at 66.9% WER.

Model	Configuration	WER
CE,80-dim	Hybrid	68.5
sMBR,80-dim	Hybrid	66.2
CE,40-dim	Hybrid	67.1
sMBR,40-dim	Hybrid	65.9

Table 5: *Stacked Architecture*

3.3. Putting it all together

This section presents the overall gains we can obtained when using the best feature space representations learnt thus far, and rebuilding the baseline GMMs with the same features. In addition, we use the new lexicon and tonal phones. One additional process involved elimination of silence frames from the beginning and end of the utterance in the training data which occupy approximately 40% of the training data. The entire training pipeline is repeated with the improved GMMs used to re-align the training data and produce target labels for the DNNs. Table 3.3 shows the WER starting from the baseline GMMs. Note that speaker-dependent GMM baseline has a WER of 72.9% now compared to a WER of 73.4% presented in Table 4. Each of the DNNs improve in the hybrid configuration over the previous WERs presented in Table 5. The 40-dimensional feature extracted from this improved DNN is used to build a GMM after applying fMPE on these bottleneck features. This improves the performance even further, resulting in a final WER of 62.9%.

We used the the best performing system at 62.9% WER to generate confusion networks on the training data provided as part of the Full Language Pack (200 hours of audio). After discarding utterances that contain no speech, the remaining data is folded into the training of DNNs. Table 3.3 illustrates the impact of adding semi-supervised training in Vietnamese. We see a reduction of 1% absolute with the bottleneck configuration.

4. Generalization

In this section, we apply the best feature spaces, training recipe and the network architecture we learnt from our experiments on the Vietnamese language to additional tonal and non-tonal languages provided in Babel option period 1. The results are presented in Table 7. It can be seen that these feature spaces and modeling techniques generalize well to other languages providing a reduction of 2.2 to 3.2% absolute in WER with the hybrid configuration. The bottleneck configurations provide additional improvements for Haitian Creole and Zulu as seen in Vietnamese. In addition, we have observed in Zulu, that addition of un-transcribed data, reduced the WER further to 69.8%.

Model	WER
SAT GMM	72.9
fMPE.SA GMM	68.8
CE,80-dim, Hybrid	66.2
sMBR,80-dim, Hybrid	64.0
CE,40-dim, Hybrid	65.2
sMBR, 40-dim, Hybrid	63.5
GMM :FMPE.40-dim Bottleneck	62.9
GMM :FMPE.40-dim Bottleneck + untranscribed data	61.9

Table 6: *Final performance of the Vietnamese LP system*

Language	SAT DNN	Hybrid	Bottleneck
Haitian Creole	63.5	60.3	59.3
Zulu	73.9	71.8	71.3
Assamese	66.7	64.8	-
Bengali	67.6	65.4	-

Table 7: *Generalization to Option Period 1 languages*

5. Conclusion

FFV features provide significant gains in tonal and non-tonal languages on top of an already rich feature set that includes discriminately trained SAT and log-mel features. I-vectors do not provide any additional gain over these features. A bottleneck configuration derived from a stacked DNN architecture can provide improvements across several languages. Realignment with a better model is important in training DNN systems. Overall, these various improvements, result in an improvement up to 15% relative over the baseline DNN in Vietnamese, with similar improvements in other languages.

6. Acknowledgements

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. 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. This effort uses the IARPA Babel Program language collection releases IARPA-babel102b-v0.5a, IARPA-babel103bv0.4b, IARPA-babel201b-v0.2b, IARPA-babel203b-v3.1a and IARPA-babel206b-v0.1e limited language packs.

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," vol. vol. 29, no. 6, 2012, pp. 82–97.

[2] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novk, and A. rahman Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *ASRU*, 2011, pp. 30–35.

[3] F. Grzl, M. Karafit, and L. Burget, "Investigation into bottle-neck features for meeting speech recognition," in *Proc. Interspeech 2009*, no. 9. International Speech Communication Association, 2009, pp. 2947–2950.

[4] K. Vesel, M. Karafit, and F. Grzl, "Convolutive bottleneck network features for LVCSR." in *ASRU*, D. Nahamoo and M. Picheny, Eds. IEEE, 2011, pp. 42–47.

[5] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE, 2010, pp. 97–102.

[6] J. Cui, X. Cui, B. Ramabhadran, J. Kim, B. Kingsbury, J. Mamou, L. Mangu, M. Picheny, T. N. Sainath, and A. Sethy, "Developing speech recognition systems for corpus indexing under the IARPA Babel program," in *Proc. ICASSP*, 2013.

[7] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEEWorkshop on Spoken Language Technology*, 2010.

[8] X. Cui, J. Xue, X. Chen, P. A. Olsen, P. L. Dognin, U. V. Chaudhari, J. R. Hershey, and B. Zhou, "Hidden markov acoustic modeling with bootstrap and restructuring for low-resourced languages." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 8, pp. 2252–2264, 2012.

[9] X. Cui, B. Kingsbury, J. Cui, B. Ramabhadran, A. Rosenberg, M. S. Rasooli, O. Rambow, N. Habash, and V. Goel, "Improving deep neural network acoustic modeling for audio corpus indexing under the IARPA Babel program," in *submitted to Interspeech 2014*.

[10] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[11] O. Glembek, L. Burget, P. Matejka, M. Karafit, and P. Kenny, "Simplification and optimization of i-vector extraction." in *ICASSP*. IEEE, 2011, pp. 4516–4519.

[12] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors." in *ASRU*. IEEE, 2013, pp. 55–59.

[13] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," in *FONETIK*, 2008.

[14] F. Metze, Z. A. W. Sheikh, A. Waibel, J. Gehring, K. Kilgour, Q. B. Nguyen, and V. H. Nguyen, "Models of tone for tonal and non-tonal languages," in *ASRU*, 2013.

[15] F. Grezl and M. Karafiát, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 470–475.