Progress in the BBN Keyword Search System for the DARPA RATS Program

Tim Ng[1], *Roger Hsiao*[1], *Le Zhang*[1], *Damianos Karakos*[1], *Sri Harish Mallidi*[2], *Martin Karafiát*[3], *Karel Veselý*[3], *Igor Szoke*[3], *Bing Zhang*[1], *Long Nguyen*[1] *and Richard Schwartz*[1]

[1] Raytheon BBN Technologies, Cambridge, MA 02138
{tng,whsiao,lzhang,dkarakos,bzhang,ln,schwartz}@bbn.com
[2] The Johns Hopkins University, Baltimore, MD, USA
mallidi.harish@gmail.com
[3] Brno University of Technology, Brno, Czech Republic
{karafiat,iveselyk,szoke}@fit.vutbr.cz

Abstract

This paper presents a set of techniques that we used to improve our keyword search system for the third phase of the DARPA RATS (Robust Automatic Transcription of Speech) program, which seeks to advance state of the art detection capabilities on audio from highly degraded radio communication channels. The results for both Levantine and Farsi, which are the two target languages for the keyword search (KWS) task, are reported. About 13% absolute reduction in word error rate (from 70.2% to 57.6%) is achieved by using acoustic features derived from stacked Multi-Layer Perceptrons (MLP) and Deep Neural Network (DNN) acoustic models. In addition to score normalization and score/system combination for keyword search, we showed that the false alarm rate at the target false reject rate (15%) was reduced by about 1% (from 5.39% to 4.45%) by reducing the deletion errors of the speech-to-text system.
Index Terms: speech recognition, KWS, MLP, DNN

1. Introduction

Based on our previous work [1], this paper presents a set of techniques that we used to improve our Keyword Search (KWS) system for the DARPA RATS (Robust Automatic Transcription of Speech) program. This program seeks to advance state of the art detection capabilities on audio from highly degraded radio communication channels. We showed in [1] that the KWS performance highly correlates with the quality of the undelying Speech-to-Text (STT) system. During the past years, STT performance has been greatly improved by using Deep Neural Network (DNN). There are two usages of DNN: (i) use Multi-Layer Perceptrons (MLP) for acoustic feature extraction [2, 3]; (ii) replace Guassian Mixture Model (GMM) by DNN in likelihood estimation for acoustic modeling using Hidden Markov Model (HMM) [4, 5, 6]. In this work, we showed that our STT systems were significanly improved by both techniques; we obtained about 13% absolute in Word Error Rate (WER) reduction (from 70.2% to 57.6%). Extra reduction in both word error rates and KWS errors can be achieved through the combinations at different levels of the two techniques.

Although KWS performance highly correlates with the quality of the underlying STT system, it may not be optimal for KWS if the STT system is tuned for the lowest WER when WER is high (at around 60% in this case). To obtain the lowest WER, one would prefer deletion to insetion as it becomes riskier to hypothesize a word when WER is high. Intuitively, a high STT deletion rate may result in a low recall for a KWS system, and degrade the KWS performance. In this paper, we propose a method to optimize the STT system for the KWS task using a weighted WER. By using the proposed method, we observed about 2% absolute improvement in recall and 17.4% to 43.5% relative reduction in false alarm rate (pFA) at the target miss rate (pMiss). The target pMiss is 15% for the Phase 3 DARPA RATS program. In this paper, all of the KWS systems are compared in pFA at this target pMiss.

The rest of the paper is organized as follows. In section 2, we describe the train and development data. In section 3, we report our progress in MLP feature extraction. The development of our DNN-HMM STT systems is described in section 4. The proposed method of optimizing a STT system for KWS is presented in section 5. In section 6, we report the system combination results for our evaluation system for the Phase 3 RATS program. The paper is concluded in section 7.

2. Train and Development Data

The Levantine systems used in this paper were trained from the official training corpora released by Linguistic Data Consortium (LDC) for the RATS program: LDC2011E94, LDC2011E114 and LDC2011E114. The training set consists of 484 hours of audio data. The official Dev2 corpus, which consists of 14 hours of audio data and 200 keywords, is used as the development set for Levantine in this work. The Farsi systems were trained from the official training corpora: LDC2011E94, LDC2012E133 and LDC2013E04. The training set consists of 347 hours of audio data. The Farsi development set consists of the audio data from the development portion of the official corpora: LDC2011E94, LDC2012E133 and LDC2013E04. The audio data for the RATS program was created by retransmitting an existing corpus through eight different communication channels. Instead of using all of the development audio data for Farsi, we randomly selected one of the eight audio signals for each of the source utterance. The resulting development set consists of 13 hours of audio data and 215 keywords.

The rank-normalization technique described in [7] is applied on all of the KWS systems in this work. The development sets for both Levantine and Farsi are divided into DEV and TEST partition for score normalization. The DEV partition consists of two third of the development set while the TEST partition consists of one third of the development set.

3. MLP Features

3.1. MLP Feature Extraction

Frequency domain linear prediction (FDLP) is an efficient technique to obtain a smooth parametric model of temporal envelope [8, 9]. Long segments of input speech (of the order of 100 ms $-$ 1000 ms) are transformed into frequency domain using Discrete Cosine Transform (DCT). The DCT samples are decomposed into subband DCT coefficients by applying critical band windowing. The sub-band temporal envelopes are then computed by applying FDLP on the subband DCT samples. The sub-band temporal envelopes are then compressed using a static compression scheme, which is a logarithmic function and dynamic compression scheme [10]. The logarithmic compression is to model the overall non-linear compression in the auditory system. The transitions are enhanced by the dynamic compression. Figure 1 shows the proposed feature extraction technique. The compressed envelopes are divided into 200 ms segments with a shift of 10 ms. DCT is applied both on static and dynamic compressed envelopes to obtain modulation spectrum representation. We use 14 modulation frequency components from each cosine transform, to cover modulation range of 0-35 Hz.

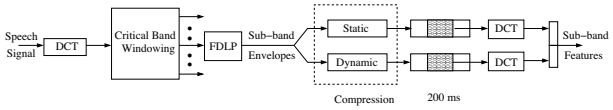

Figure 1: *Blockdiagram for Frequency Domain Linear Prediction feature extraction*

In this work, we use a long-term window of 10 seconds in the FDLP feature extraction. Although it is in 16KHz sampling rate, the RATS audio data is originally from the telephone corpora. Configurations for narrow-band data in signal processing are used. The low and high cut-off frequencies are set to 125Hz and 3800Hz, respectively. Seventeen critical filter banks are derived from this frequency bandwidth. After applying the two versions of compression schemes, there are 476 parameters/dimensions in the resulting FDLP feature.

In addition to the 476-dimensional FDLP features, a pitch value is estimated for each frame using the RAPT algorithm [11], followed by a speaker-based mean and variance normalization. We expand the pitch feature context with an 11-frame concatenation. These 11 pitch values are then appended to the 476 FDLP features and the resulting 487 features are input to the MLP.

For the MLP training, we borrow the stacking strategy described in [2]. Instead of using context-independent HMM phone states as targets, the cross-word State Cluster Tied Mixture (SCTM) [12] codebooks are used in this work. The configuration for the first MLP is $487 \times 1500 \times 1500 \times 80 \times 1500 \times N$, where N is the number of SCTM codebooks. N is 3707 for Levantine while 5306 for Farsi. The 80 bottle-neck outputs from the first MLP are sampled at times t, $t-10$, $t-5$, $t+5$ and $t+10$. Where t is the index of the current frame. The resulting 400-dimensional features are input to the second MLP with a configuration of $400 \times 1500 \times 1500 \times 80 \times 1500 \times N$. Instead of using the 80 outputs from the bottle-neck layer from the second MLP, the 1500 outputs from the hidden layer from the bottle-neck layer are extracted. These 1500 outputs are denoted as BBN MLP features.

The similar approach as described in [13] is employed to integrate the MLP features into our STT system. In the Speaker Independent (SI) training, the 1500-dimensional features are augmented with the 135-dimensional long span features which is 9-frame concatenation of 15-dimensional PLP. Region Dependent Transform (RDT) is applied on the resulting 1635-dimensional features to reduce the dimensionality to 46. The similar procedure is used in the Speaker Adaptive Training (SAT) training. Prior to be combined with the MLP features, a CMLLR transform is applied on the 15-dimensional PLP features. Another CMLLR is applied on the 46-dimensional features output from RDT.

As shown in Table 1, by incorporating the BBN MLP features into the GMM-HMM STT systems, about 11% and 8% absolute reduction in WER are obtained for Levantine and Farsi, respectively.

Levantine	PLP only	72.27
	RDT(PLP + BBN MLP)	61.53
Farsi	PLP only	70.23
	RDT(PLP + BBN MLP)	62.08

Table 1: *Reduction in word error rate for using MLP Features with RDT for GMM-HMM system*

3.2. MLP Feature Combination

In addition to the BBN MLP features, we also have MLP features from Brno University of Technology (BUT), who is one of our collaborators for the RATS program. The BUT MLP features have 69 dimensions. They are derived from the MLPs trained from the Mel-filter bank energies [3]. Since RDT directly reduces the speech recognition errors and is flexible for input feature dimensionality, the two sets of MLP features can be integrated into a single STT system.

As shown in Table 2, the BBN Byblos GMM-HMM STT system is used in all of the experiments. The BUT MLP features are first combined with our PLP features using the same procedure as described in Section 3.1. The STT system with BUT MLP features is about 2.3% and 1.0% better in terms of STT WER as compared to the system with BBN MLP features. By combining the two sets of MLP features, we obtained 1.6% and 2.3% reduction in WER for Levantine and Farsi, respectively.

4. Deep Neural Network

Our deep neural network (DNN) systems are trained in three steps. The first step is the pre-training for initialization purpose. The second phase is the cross entropy training and the last step is the sequence training [4].

Levantine	RDT(PLP + BBN MLP)	69.07
	RDT(PLP + BUT MLP)	67.77
	RDT(PLP + BBN MLP + BUT MLP)	66.15
Farsi	RDT(PLP + BBN MLP)	62.08
	RDT(PLP + BUT MLP)	61.15
	RDT(PLP + BBN MLP + BUT_MLP)	58.74

Table 2: *Reduction in word error rate for using MLP features with RDT for GMM-HMM system*

For pre-training, we start with a randomly initialized shallow network. We randomly select 20% of the available training data and train the network for one epoch. After that, we add one more layer with randomized weights, and re-train with another set of randomly selected data. This procedure continues until the target number of layers is reached.

After pre-training, we perform cross entropy training using the entire training set. In which, 5% of the data is held out for cross validation purpose. This cross validation set is used to monitor the progress of the training by computing the frame accuracy.The learning rate starts with 0.002. If the improvement of the frame accuracy falls below 0.25% absolute, the learning rate is reduced by half for each epoch. This scheme has shown to be effective in DNN training [5].

After cross entropy training, we perform one iteration of sequence training which optimizes for minimum phone error (MPE). The training is smoothed by a technique called frame smoothing (f-smoothing) proposed in [6] where the discriminative objective function is interpolated with the cross entropy function.

For Levantine, we trained three DNN systems. The first one uses the final features from RDT combining PLP and BBN MLP features. The second one is similar but it uses BUT MLP features for combination. The last one is using RDT to combine PLP, BBN MLP and BUT MLP features. For Farsi, we built the same DNN systems as well. All DNNs have 4 layers and each layer contains 1024 hidden units. Sequence training is applied to the DNN systems using BUT MLP features only due to the time constraint. Table 3 summarizes the performance of our DNN systems in WER.

Levantine	RDT(PLP + BBN MLP)	69.84
	RDT(PLP + BUT MLP)	67.02
	RDT(PLP + BBN MLP + BUT MLP)	66.39
Farsi	RDT(PLP + BBN MLP)	61.07
	RDT(PLP + BUT MLP)	59.87
	RDT(PLP + BBN MLP + BUT MLP)	57.55

Table 3: *Word error rate for DNNs using MLP features with RDT*

5. Optimizing Speech-to-Text System for Keyword Search

To optimize our STT system for WER, Powell's method [14] is employed to estimate the weights used in the decoder to combine scores linearly for Viterbi search by reducing the WER of a set of N-best lists. The scores are acoustic model score, language model score, pronunciation score, phone insertion penalty, silence insertion penalty and word insertion penalty. However, reducing WER does not necessarily improves KWS performance. When WER is high, the optimization would prefer deletions to insertions as it takes more risk to hypothesize a word. Deletions may degrade KWS performance because of worse recall. In this work, we propose to optimize weighted WER instead. $WER_{weighted} = \frac{S+\alpha D+\beta I}{N}$, where S, D and I are number of substitutions, number of deletions and number of insertions, respectively. α is a constant which is greater than 1 to emphasize deletions in the objective function, and β a constant less than 1 to deemphasize insertions. N is the total number of tokens in the reference. Due to time constraint, the technique is only applied on the GMM-HMM Levantine KWS systems.

As shown in Table 4, by using $\alpha = 1.3$ and $\beta = 0.3$, when compared to the baseline system, recall is increased from 86.87% to 89.14% for the DEV partition and from 92.56% to 94.42% for the TEST partition while the WER increases from 69.07% to 72.35%. In addition to the improvement in recall, false alarm rate reduces from 5.39% to 4.45% for the DEV partition and 1.54% to 0.87% for the TEST partition. The baseline system is optimized to reduce WER.

System	WER	DEV		TEST	
		Recall	pFA	Recall	pFA
baseline	69.07	86.87	5.39	92.56	1.54
$\alpha = 1.3, \beta = 0.3$	72.35	89.14	4.45	94.42	0.87

Table 4: *Keyword search results for Levantine by optimizing decoding weights on weighted word error rate*

6. System Combination

In addition to the various systems developed at BBN, BUT also sent us their KWS outputs. Due to the fact that the systems are different in terms of acoustic features, acoustic models, and keyword search approaches, system combination should be beneficial. The procedure we followed for system combination was almost identical to that reported in [7], except that the order of the systems and the initial weights in Powell's method were determined based on the performance measure (pFA at 15% pMiss) instead of Actual Term-Weighted Value (ATWV).

The KWS systems used in the system combination for Levantine are shown in Table 5. The systems developed at BBN are listed in the first block of the table. There are 4 GMM-HMM and 4 DNN-HMM systems from BBN with different MLP features. The GMM-HMM systems are optimized on weighted WER for KWS. The outputs from BUT are in the second block of the table. Two different STT systems were developed at BUT: STK and Kaldi. The STK system is a GMM-HMM system using BUT MLP features while DNN-HMM is used in the Kaldi system which was trained on the PLP features only. Two sets of hits (one from whole-word decoding and another from sub-word decoding) were generated by the STK system. In addition to the hit list, BUT also sent us the lattices output from the Kaldi system. KWS are done on these lattices at BBN after converting them to consensus networks. The results show that the hit lists from BUT cannot reach the target pMiss because of low recall.

The systems are combined hierachically. As shown in the third block of the table, the systems are first combined among their own category. As compared to the best single system in the category, significant reductions in pFA are obtained from the first level system combination. For the BBN GMM-HMM systems, 67% relative reduction in pFA is achieved, 39% relative for BBN DNN-HMM systems and 3% for BUT Kaldi outputs. The final combination gives about extra 44% relative reduction in pFA.

Similarly, the systems for Farsi are shown in Table 6. There are 3 GMM-HMM systems and 4 DNN-HMM systems from BBN and they are listed in the first block of the table. Because of time constraint, no optimization on weighted WER is done on these BBN systems. The outputs from BUT are shown in the second block of the table. In addition to the 3 hit lists from the STK system, BUT also sent us the whole-word lattices from the STK system. Two DNN-HMM Kaldi systems were developed at BUT for Farsi. The Kaldi-1 system was trained on the conventional PLP features while Kaldi-2 on the enhanced PLP features. The systems are also combined hierarchically. The similar improvements are obtained from the system combination.

System	DEV		TEST	
	Recall	pFA	Recall	pFA
GMM BBN MLP	89.03	1.4937	88.38	1.4144
GMM BUT MLP	89.20	1.0962	88.87	1.1576
GMM GRAP BUT MLP	87.89	1.4401	87.72	1.4351
GMM BBN + BUT MLP	90.51	0.8216	90.02	0.8047
DNN BBN MLP	88.05	1.7973	87.56	1.7758
DNN BUT MLP	88.87	1.0769	88.71	1.0749
DNN SEQ BUT MLP	89.20	1.0289	89.20	1.0093
DNN BBN + BUT MLP	88.71	1.2372	88.54	1.2660
BUT STK Hit	76.92	–	76.92	–
BUT STK Subword Hit	69.07	–	69.07	–
BUT KALDI Lattices	86.91	1.5425	86.91	1.5362
BUT KALDI Hit	67.76	–	67.76	–
SYSCOM BBN-GMM	96.07	0.2707	93.45	0.2692
SYSCOM BBN-DNN	94.44	0.6295	93.45	0.6246
SYSCOM BUT-STK	78.56	–	78.56	–
SYSCOM BUT-KALDI	86.91	1.4956	86.91	1.5801
SYSCOM ALL	97.05	0.1200	95.25	0.1183

Table 5: *Hierarchical System Combination for Levantine*

7. Conclusion

The paper presents the techniques that we used to develop our keyword search system for the evaluation of the Phase 3 DARPA RATS program. We showed that about 13% absolute reduction in word error rate can be achived by employing Multi-Layer Perceptron for feature extraction and Deep Neural Network for acoustic model likelihood estimation. We also proposed a method to optimize a Speech-to-Text system to improve the keyword search performance, and significant reduction in keyword search errors is reported. Finally, we showed that tremendous reduction in keyword search errors is achived through system combination.

System	DEV		TEST	
	Recall	pFA	Recall	pFA
GMM BBN MLP	93.29	0.2807	92.91	0.2799
GMM BUT MLP	91.88	0.2586	91.55	0.2604
GMM BBN + BUT MLP	91.42	0.2161	91.36	0.2298
DNN SEQ BBN MLP	91.94	0.2729	91.81	0.2704
DNN BUT MLP	91.49	0.2699	91.36	0.2694
DNN SEQ BUT MLP	91.55	0.3069	91.42	0.3470
DNN BBN + BUT MLP	92.26	0.2444	92.07	0.2436
BUT STK Subword Hit	80.85	–	80.79	–
BUT STK-CN Hit	91.04	0.3493	90.52	0.3493
BUT STK Hit	88.33	0.1687	88.33	0.1687
BUT STK Lattices	94.78	0.1858	94.00	0.1835
BUT KALDI-1 Hit	83.56	–	83.56	–
BUT KALDI-1 Lattices	91.94	0.3498	91.81	0.3462
BUT KALDI-2 Hit	83.37	–	83.37	–
BUT KALDI-2 Lattices	93.10	0.2760	93.10	0.2794
SYSCOM BBN-GMM	96.65	0.0853	95.42	0.0853
SYSCOM BBN-DNN	96.00	0.1148	94.84	0.1146
SYSCOM STK	95.16	0.1168	95.04	0.1166
SYSCOM KALDI	95.42	0.1412	94.91	0.1401
SYSCOM ALL	98.07	0.0566	0.9658	0.0564

Table 6: *Hierarchical System Combination for Farsi*

8. Acknowlegement

This paper is based upon work supported by the DARPA RATS Program. It has been approved for public release and distribution is unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

9. References

[1] B. Zhang, R. Schwartz, S. Tsakalidis, L. Nguyen, and S. Matsoukas, "White listing and score normalization for keyword spotting of noisy speech," in *Proc. of Interspeech*, Portland, Oregon, Sep 2012.

[2] F. Grézl, M. Karafiát, and L. Burget, "Investigation into bottleneck features for meeting speech recognition," in *Proceedings of Interspeech*, no. 9, Brighton, U.K., 2009, pp. 2947–2950.

[3] M. Karafiát, F. Grézl, M. Hannemann, K. Veselý, and J. Černocký, "BUT babel system for spontaneous cantonese," in *Proceedings of Interspeech*, no. 8, 2013, pp. 2589–2593.

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

[5] A. Senior, G. Heigold, M. Ranzato, and K. Yang, " An Empirical Study of Learning Rates in Deep Neural Networks for Speech Recognition ," in *ICASSP*, 2013, pp. 6724–6728.

[6] 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 *ICASSP*, 2013, pp. 6664–6668.

[7] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grezl, M. Hannemann, M. Karafiat, I. Szoke, K. Vesely, L. Lamel, and V.-B. Le, "Score normalization and system combination for improved keyword spotting," in *Proc. ASRU 2013*, Olomouc, Czech Republic, 2013.

[8] M. Athineos and D. Ellis, "Autoregressive modelling of temporal envelopes," *IEEE Trans. on Signal Processing*, vol. 55, pp. 5237–5245, 2007.

[9] S. Ganapathy, S. Thomas, and H. Hermansky, "Static and dynamic modulation spectrum for speech recognition," in *Proceedings of Interspeech*, Brighton, U.K., September 2009.

[10] J. Tchorz and B. Kollmeier, "A model of auditory perception as front end for automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 106, no. 4, pp. 2040–2050, 1999.

[11] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, pp. 495–518, 1995.

[12] L. Nguyen, S. Matsoukas, J. Davenport, F. Kubala, R. Schwartz, and J. Makhoul, "Progress in transcription of broadcast news using byblos," *Speech Commun.*, vol. 38, no. 1, pp. 213–230, Sep. 2002.

[13] T. Ng, B. Zhang, S. Matsoukas, and L. Nguyen, "Region dependent transform on mlp features for speech recognition," in *Proceedings of Interspeech*, Florence, Italy, August 2011, pp. 221–224.

[14] M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The Computer Journal*, vol. 7, no. 2, pp. 155–162, 1964.