BUT BABEL system for spontaneous Cantonese

Martin Karafiát, František Grézl, Mirko Hannemann, Karel Veselý, and Jan "Honza" Černocký

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic

karafiat,grezl,ihannema,iveselyk,cernocky@fit.vutbr.cz

Abstract

This paper presents our work on speech recognition of Cantonese spontaneous telephone conversations. The key-points include feature extraction by 6-layer Stacked Bottle-Neck neural network and using fundamental frequency information at its input. We have also investigated into robustness of SBN training (silence, normalization) and shown an efficient combination with PLP using Region-Dependent transforms. A combination of RDT with another popular adaptation technique (SAT) was shown beneficial. The results are reported on BABEL Cantonese data.

Index Terms: speech recognition, discriminative training, bottle-neck neural networks, region-dependent transforms

10.21437/Interspeech.2013-582

1. Introduction

This paper presents our recent effort to build an automatic speech recognition (ASR) system for Cantonese spontaneous telephone conversations. The work was mainly driven by our participation in the BABEL project ("Babelon" consortium coordinated by BBN). Unlike the "classical" style of ASR development with almost infinite time and generous resources, BABEL aims at building keyword spotting systems for languages with limited resources in limited amount of time. This time varies from almost one year to just one week at the end of the project.

So far, the best keyword spotting systems developed are always based on Large Vocabulary Continuous Speech Recognition (LVCSR) front-end. Accuracy of such keyword spotting system correlates with LVCSR because both tasks require high quality acoustic models. Consequently, our initial focus was put into LVCSR on four Babel languages released in the 1st year: Cantonese, Pashto, Tagalog and Turkish. The main development was done on Cantonese and the approaches were checked on other languages later.

The basis of our system is a state-of-the-art Hidden Markov model/Gaussian mixture model (HMM/GMM) recognizer that our group has been developing since 2004 [1]. For this work, we concentrated on three main topics:

1. Neural Network (NN) based feature extraction. The recently released Stacked Bottle Neck architecture [2] was found to overcome the Bottle-Neck one. NNs are nowadays widely used for acoustic modeling as Deep Neural Networks [3], which sparkled our interest to experiment with deeper architectures also in Bottle-Neck feature extraction (see section 5.2).
2. Cantonese is a tonal language, tonal information has an important influence in speech recognition. Therefore, we investigated into using fundamental frequency (f0) and also probability of voicing as additional features processed by NN (see section 5.2).
3. Finally, discriminatively trained Region-Dependent Transforms (RDT) [4] provided an additional improvement on top of NN based features (that are already discriminatively trained to reduce Frame Error Rate !). Using RDT with speaker-adaptive training (SAT) was investigated as well. The definition of RDT is given in section 3 and experimental results are in 5.3

2. Neural Network features in speech recognition

Neural networks were used to generate Bottle-Neck (BN) or Stacked Bottle-Neck (SBN) features. We introduced the SBN structure in [2]; the scheme is given in figure 1. It contains two NNs: the BN outputs from the first one are stacked, downsampled, and taken as an input vector for the second NN. This second NN has again a BN layer, of which the outputs are taken as input features for GMM/HMM recognition system. Our previous study [5] has shown that BN neurons with linear activation functions provide better performance.

3. Region-Dependent Transform

In the RDT framework, an ensemble of linear transformations is trained, typically using the discriminative Minimum Phone Error (MPE) criterion [6]. Each transformation corresponds to one region in partitioned feature space. Each feature vector is then transformed by a linear transformation corresponding to the region the vector belongs to. The resulting (generally non-linear) transformation has the following form:

$$F_{RDT}(\mathbf{o}_t) = \sum_{r=1}^{N} \gamma_r(t) \mathbf{A}_r \mathbf{o}_t, \quad (1)$$

where $\mathbf{A}_r$ is linear transformation corresponding to rth region, and $\gamma_r(t)$ is probability that feature vector $\mathbf{o}_t$ belongs to rth region. The probabilities $\gamma_r(t)$ are typically obtained using a GMM (pre-trained on the input features) as mixture component posterior probabilities. Usually, RDT parameters $\mathbf{A}_r$ and ASR

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. Next, it was supported by, the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070 and Czech Ministry of Education project No. MSM0021630528. M. Karafiat was supported by Grant Agency of the Czech Republic post-doctoral project No. P202/12/P604.

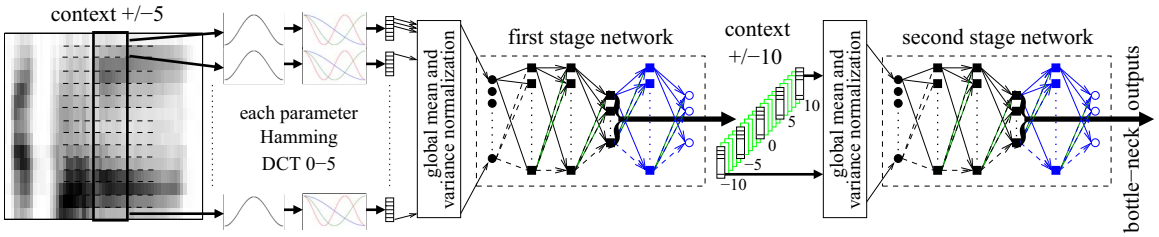

Figure 1: *Stacked Bottle-Neck Neural Network feature extraction.*

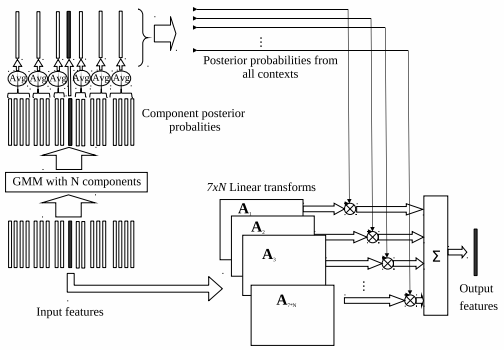

Figure 2: *Region Dependent Transform.*

acoustic model parameters are alternately updated in several iterations. While RDT parameters are updated using discriminative MPE criterion, ML update is typically used for acoustic model parameters [7],[4].

RDT can be seen as a generalization of previously proposed fMPE discriminative feature transformation. The special case of RDT with square matrices $\mathbf{A}_r$ was shown [4] to be equivalent to fMPE with offset features as described in [8]. From the fMPE recipe [7], we have adopted the idea of incorporating context information by considering $\gamma_r(t)$ corresponding not only to the current frame but also to the neighboring frames. From our experience, such incorporation of contextual information leads to significantly better results compared to the RDT style proposed in [4], where feature vectors of multiple frames were stacked at the RDT input. Therefore, our RDT configuration (figure 2) is very similar to the one described in the fMPE recipe.

4. System description

4.1. Data

The BABEL data[1] simulate a case of what one could collect in limited time from a completely new language: it consists of two parts: scripted (speakers read text through telephone channel) and conversational (spontaneous telephone conversations). The test data contains conversational speech only. See table 1 for details.

Ideally, the type of training data should be consistent with the test, which would call for training on conversational part only. However, according to our experiment on NN system (see table 7), we gained 0.2% absolute by using scripted data, so we used both parts for the training.

[1]Collected by Appen `http://www.appenbutlerhill.com`

Table 1: *Data*

Data	No. of speakers	size [h]
training-conv	965	109
training-scripted	399	29
training-all	1364	138
test	20	2.5

The phoneme set consists of 15 unvoiced phonemes and 24 voiced phonemes where 6 tones are distinguished, summing up to 158 phonemes.

4.2. Baseline system and PLP feature extraction

Speech recognition system is HMM-based with cross-word tied-states triphones, with approximately 4500 tied states and 18 Gaussian mixtures per state, trained from scratch using mix-up maximum likelihood training. Final word transcriptions are decoded using 3gram Language Model (LM) trained only on the transcriptions of training data[2].

Mel-PLP features are generated in classical way, the resulting number of coefficients is 13. Deltas, double- and in HLDA system also triple-deltas are added, so that the feature vector has 39, respectively 52, dimensions. Cepstral mean and variance normalization is applied with the means and variances estimated per conversation side. HLDA is estimated with Gaussian components as classes to reduce the dimensionality to 39. In our experiments leading to the best results, the PLP features are forwarded through HLDA and concatenated with SBN features.

4.3. SBNs for feature extraction

The input features of the first NN (figure 1) are 15 Critical-Band Energies (CRBE) obtained with a Mel filter-bank, with conversation-side-based mean subtraction applied. 11 frames of these features are stacked and a Hamming window multiplies the time evolution of each parameter [9]. Finally, DCT is applied, of which 0th to 5th coefficients are retained, making the size of the feature vector $15 \times 6 = 90$.

The sizes of the both NNs were set to 1M weights for most of the experiments. When the best input features, structure and normalization were found, NN sizes were increased to 2M weights. Both NNs were trained to classify phoneme states (3 states per phoneme). These targets were generated by forced alignment with baseline PLP models (section 4.2) and stayed fixed during the training.

Final BN features produced by various NN structures were transformed by Maximum Likelihood Linear Transform (MLLT), which considers HMM states as classes. For any set

[2]This is coherent to BABEL rules, where *the provided data only* can be used for system training in the primary condition

Table 2: *PLP system, BN and SBN baselines.*

System	CER[%]
ML PLP	62.8
ML PLP-HLDA	61.2
MPE SAT-RDT PLP-VTLN-HLDA	52.0
BN - NoSilenceReducion (MLP5)	63.8
SBN - NoSilenceReducion (MLP6)	53.3

of features, new models were trained by single-pass retraining from PLP baseline system. Next, 12 maximum likelihood iterations followed to better settle new HMMs in the new feature space.

4.4. Pitch and voicing

We also experimented with F0 estimates and probability of voicing as additional features concatenated with CRBE. The estimation of F0 is based on normalized cross-correlation function. The maximum of this function indicates F0 value. Dynamic programming is used for smoothing. The implementation of F0 and probability of voicing estimation followed [10].

5. Experiments

5.1. PLP system

The baseline system trained using PLPs was giving 62.8% Character Error Rate (CER), resp. 61.2% with HLDA which is our standard feature post-processing (see Table 2). The improvement with PLPs using all discriminative approaches described below was about 9% absolute. The basic PLP based system generated forced alignments on phoneme-state-level that were used as targets for further NN training.

5.2. Stacked Bottle-Neck NN features

Basic Bottle-Neck NN architecture with only one neural network (the size of the BN layer is 30) performs about 1% worse than PLP baseline. Table 2 also shows big improvement by using SBN instead of standard BN, by almost 10%. Note that the results are not directly comparable, due to using different numbers of layers (6-layer SBN versus 5-layer BN), but we will see later in table 4, that for SBN, the gain from MLP6 is only 2%.

5.2.1. Silence in the training

We found that the data contained huge amount silence (more than 50%). Therefore, we hypothesized that NNs have been focusing too much in silence/speech recognition rather than phoneme-state classification. After removing silence, huge drop of frame accuracy (from 70% to 40%) was observed on cross-validation set during BN-NN training (due to removal of "easy" silence frames) but the final BN features gave us 3.2% absolute improvement (Table 3). The influence (or rather lack of influence) of silence removal is even more interesting with SBN architecture: according to Table 3, no drop-off accuracy is observed due to training on huge amount of silence: the first NN is obviously affected by silence but the second one reads already compressed information, therefore it can be better trained. Finally, we experimented with silence removal only for the training of the first NN (denoted as HalfSilenceReducion): the best result indicates that this generates an NN structure working the best with given segmentation. Unfortunately this improvement was lost when we compared "SilenceReduction" and "HalfSilenceReducion" on test segmentation given by Voice Activity Detection (significantly less silence), therefore we returned to "SilenceReduction" in the following experiments.

Table 3: *Silence reduction in standard bottle neck and stacked bottle neck architecture.*

System	CER[%]
BN - NoSilenceReducion (MLP5)	63.8
BN - SilenceReducion (MLP5)	60.6
SBN - NoSilenceReducion (MLP6)	53.3
SBN - SilenceReducion (MLP6)	53.3
SBN - HalfSilenceReducion (MLP6)	52.3

Table 4: *Number of layers in Stacked Bottle Neck NN.*

System	CER[%]
MLP5	55.2
MLP6	53.3
MLP7	53.7

5.2.2. Making the NN deep

Using more hidden layers (Deep NN) is now widespread in the community. Table 4 shows the effect of splitting parameters lying in the first hidden layer (before BN layer) into more layers. Both NNs in Stacked BN structure were split in the same way. We have shown that 6-layer architecture (so that 4 layers are active in feature generation after the last two layers are cut off) gives almost 2% absolute improvement, but splitting into even more layers do not help anymore, probably due to difficult initialization. We also experimented with Restricted Boltzmann Machine initialization [3], but we did not get any improvement. Therefore, 6-layer SBN was selected for further experiments.

5.2.3. Normalization

Usual pre-processing for NNs involves global mean and variance normalization of features. We used also conversation-side based mean normalization. This gives us a nice improvement of 0.3% compared to global normalization only (53.3% vs. 53.6%).

5.2.4. F0

Cantonese is a tonal language, therefore the fundamental frequency (f0) has significant effect on final system behavior. F0 is a "bad" feature in HMM modeling due to long constant parts in unvoiced regions. It is also not Gaussian distributed. Processing F0 through BN network encodes this information into feature space which can be easier modeled by HMM. Moreover, it should significantly help the NN to classify different versions of voiced phonemes. Both should lead to improved BN feature extraction.

By adding f0 (with derivatives) to the final feature stream, an absolute improvement of 1.8% is obtained (table 5). If, however, F0 is added into NN input, we obtain nice 3.3% absolute improvement. We experimented also with adding probability of voicing (m) — here, it did not provide any improvement but also no deterioration, and on other BABEL languages we found this feature useful (0-0.5% absolute), so it was retained. The final NN feature extraction structure is therefore SBN with f0 and

Table 5: *Adding additional features as input (SBN MLP6).*

System	CER[%]
SBN (CRBE)	53.3
SBN (CRBE)+f0_D_A	51.5
SBN (CRBE+logf0)	50.7
SBN (CRBE+f0)	50.0
SBN (CRBE+f0+m)	50.0
SBN (CRBE+f0+m) (4M weights)	49.2

Table 6: *Re-segmentation in HMM training and feature concatenation.*

System	OrigSegm CER[%]	VAD CER [%]
SBN	49.2	48.8
SBN reseg	47.7	47.3
PLP-HLDA+**SBN**+f0_D_A reseg	48.0	47.9

probability of voicing. Finally, we increased the sizes of NNs to 2M weights each (4M together) which gave us about 0.8% absolute improvement. This structure will be further denoted as **SBN**.

5.2.5. Silence in the training II.

Removing silence in NN training was found useful (section 5.2.1, therefore we analyzed this re-segmentation also in HMM training/test. For HMM training, we used the same segmentation as in NN training (based on forced alignment) and a Voice Activity Detection (VAD) based on NN was used for test. The first two lines in Table 6 indicate 1.9% absolute improvement by removing silence from training and also from test.

5.3. Region Dependent Transforms

The final feature stream was built by concatenation of PLP-HLDA (39 dimensions), **SBN** (30) and f0_D_A (3) adding up to final dimensionality 72. (Note, our experiments showed a marginal effect by using a VTLN on the PLP feature stream therefore it was not applied due to simplicity.) This system is 0.6% absolutely worse than **SBN** features only (Table6). It is caused by significant difference between **SBN** and baseline PLP based systems (**SBN** is much better) and also by having f0 twice in the system: one may question the independence of features, as f0 is already integrated in the **SBN** output. RDT should fix these problems.

5.3.1. Structure of RDT system

According to our previous experiments, GMM with 125 components was chosen. In the RDT system, posterior probabilities of GMM components for the current frame are stacked with the averages of posteriors for adjacent frames 1-2, 3-5 and 6-9 on the right and likewise on the left (i.e. 7 groups spanning 19 context frames in total). The resulting 7×125-dimensional vector serves as weights $\gamma_r(t)$ in (1) for corresponding 7×125 transformations: $F \times F$ matrices, where F is feature dimensionality, see block diagram in figure 2. In [11], we presented significant gain by adding such posterior probabilities from adjacent frames.

The GMM model is created by pooling and merging all Gaussian components from well trained baseline ML models. More details about the clustering can be found in [12].

Table 7: *Removing of scripted data.*

System	VAD segm CER[%]
SBN reseg - all. data	47.3
SBN reseg - conv. data	47.5

Table 8: *RDT results.*

System	CER [%]
ML **SBN**	47.3
ML RDT30 **SBN**	46.0
ML PLP+**SBN**+f0_D_A	47.9
ML RDT72 PLP+**SBN**+f0_D_A	45.0
ML RDT72to69 PLP+**SBN**+f0_D_A	45.0
ML RDT72to69 PLP+**SBN**+f0_D_A CMLLR	44.4
ML SAT PLP+**SBN**+f0_D_A	45.8
ML SAT RDT72to69 PLP+**SBN**+f0_D_A	43.4
MPE SAT RDT72to69 PLP+**SBN**+f0_D_A	42.4

5.3.2. RDT results

According to table 8, RDT applied on **SBN** features improves the result by 1.3% absolute. When **SBN** feature stream is concatenated with PLP and F0 (with derivatives) it gives 2.3% improvement over pure **SBN** features. Therefore, RDT is obviously gaining from complementarity of PLPs.

Next, we played with dimensionality reduction by RDT. We found that small dimensionality reduction (by 3, corresponding to the size of F0 features) did not change the result. When we tried to go further, a decrease of accuracy was observed.

Constrained Maximum Likelihood Linear Regression (CMLLR) adaptation [13] over RDT feature stream gave 0.6% absolute improvement. Straightforward use of CMLLR is however dangerous, as CMLLR is estimated by Maximum Likelihood, therefore part of discriminability given by RDT is lost. To solve this problem, we used Speaker Adaptive Training (SAT) similarly to [14]: a set of CMLLR transforms was generated by ML model and RDT was estimated on top of CMLLR-rotated features. This gives a nice 1% additive gain to CMLLR estimated on top of RDT. The last line in Table 8 shows 1% additive improvement given by final discriminative retraining of HMM with MPE criterion, it is our final result.

6. Conclusions

The novel things we have brought to our BABEL Cantonese system include 6-layer Stacked Bottle-Neck features and using f0 at the input of this NN. We have also investigated into robustness of SBN training (silence, normalization) and shown an efficient combination with PLP and (again!) F0 features using Region-Dependent transforms. Last by not least, a combination of RDT with another popular adaptation technique (SAT) was shown beneficial.

Our future work will include extensive testing of the investigated approaches on other BABEL languages, and study of transforms in a DNN system, as suggested in [15].

7. References

[1] L. Burget, "Study of linear transformations applied to training of cross-domain adapted large vocabulary continuous speech recognition systems," Ph.D. dissertation, Brno University of Technology, 2009.

[2] F. Grezl, M. Karafiat, and L. Burget, "Investigation into bottleneck features for meeting speech recognition," in *Proc. Interspeech 2009*, no. 9, 2009, pp. 2947–2950.

[3] 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," *Signal Processing Magazine*, pp. 14–22, 2012.

[4] B. Zhang, S. Matsoukas, and R. Schwartz, "Recent progress on the discriminative region-dependent transform for speech feature extraction," in *Proc. of Interspeech2006*, Pittsburgh, PA, USA, Sep 2006, pp. 2977–2980.

[5] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *Proceedings of ASRU 2011*, 2011, pp. 42–47.

[6] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2003.

[7] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *in Proc. IEEE ICASSP*, 2005.

[8] D. Povey, "Improvements to fMPE for discriminative training of features," in *Proc. of Interspeech2005*, Lisbon, Portugal, Sep 2005, pp. 2977–2980.

[9] P. Schwarz, "Phoneme recognition based on long temporal context," Ph.D. dissertation, Brno University of Technology, 2009.

[10] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. Paliwal, Eds. New York: Elseviever, 1995.

[11] M. Karafiát, L. Burget, P. Matějka, O. Glembek, and J. H. Černocký, "ivector-based discriminative adaptation for automatic speech recognition," in *Proc. ASRU 2011*, dec 2011.

[12] D. Povey, L. Burget, M. Agarwal, P. Akyazi, A. Ghoshal, O. Glembek, K. N. Goel, M. Karafiát, A. Rastrow, R. Rose, P. Schwarz, and S. Thomas, "The subspace Gaussian mixture model-a structured model for speech recognition," *Computer Speech and Language*, vol. 25, no. 2, pp. 404–439, 2011.

[13] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," 1997. [Online]. Available: citeseer.ist.psu.edu/gales97maximum.html

[14] L. Chen, M. J. F. Gales, and K. K. Chin, "Constrained discriminative mapping transforms for unsupervised speaker adaptation," in *Proc. ICASSP*, Prague, Czech Republic, 2011.

[15] S. P. Rath, D. Povey, K. Vesely, and J. Cernocky, "Improved feature processing for deep neural networks," in *accepted to Interspeech*, 2013.