The FBK system for the CHiME-4 challenge

Marco Matassoni, Mirco Ravanelli, Shahab Jalalvand, Alessio Brutti, Daniele Falavigna

Fondazione Bruno Kessler, Trento, Italy

{matasso,mravanelli,jalalvand,brutti,falavi}@fbk.eu

Abstract

This paper describes the ASR system submitted by FBK to the CHiME-4 challenge for the single channel track. The proposed solution employs multiple subsystems, whose DNNs are trained with different training criteria and strategies (i.e. diverse training material, with and without batch normalization). A "self" adaptation of acoustic models is applied to each subsystem, relying on a blind estimate of the accuracy of automatic transcriptions. This adaptation, performed in a batch fashion over the entire evaluation set, significantly improves the performance of each subsystem. The final output is obtained by combining the multiple transcriptions through ROVER, which provides a further improvement, reducing the average WER on the evaluation set from 22.3% to 16.5%.

1. Introduction

In a number of application scenarios (e.g., home automation, smart cars, robots), performance of automatic speech recognition (ASR) is heavily affected by noises of various types, competing speakers and reverberation effects. The CHiME challenges [1, 2, 3, 4] represent an excellent framework to evaluate signal enhancement and noise-robust acoustic models for ASR in such realistic conditions. Built upon the previous CHiME-3 challenge, the CHiME-4 dataset comprises utterances recorded by a 6-channel tablet-based microphone array. The recognition task is the automatic transcription of read sentences from the Wall Street Journal (WSJ) corpus, acquired in four noisy conditions; [4] illustrates training, development and evaluation data sets released for the competition. The results in [3] proved the effectiveness of signal enhancement approaches combined with the use of hybrid acoustic models based on deep neural networks hidden Markov models (DNN-HMMs) [5, 6, 7, 8].

In this submission we consider the *1ch*-track of the challenge, focusing specifically on deep learning techniques and building upon our previous submission for the CHiME-3 challenge [9], where an effective two-pass strategy was explored. In that work the DNNs employed to recognize each input stream (beamformed or single channels) were re-trained using the corresponding automatic transcription generated with the baseline acoustic models. A MAP selection procedure, at sentence level, produced the improved final transcriptions.

For the current *1ch*-track CHiME-4 challenge, only a single channel is available in the decoding pass and the multiple hypotheses generated for a final ROVER combination are derived from systems exploiting not only different training material, as done in [9], but also introducing a variety of DNN architectures. Secondly, we improve the model adaptation stage, replacing the standard retraining on the whole adaptation set with a more sophisticated solution, which enhances the adaptation with effective instance weighing and selection criteria. Finally, the combination of the hypotheses provided by the sub-systems is based on our previous work on driving ROVER with segment-based ASR quality estimation [10].

The paper presents in Section 2 the approach and the main features of the proposed system while Section 3 describes the steps of the processing pipeline and Section 4 reports the corresponding WER results. Section 5 concludes the work, presenting possible future directions.

2. Main characteristics

The main features explored in our current submission are the introduction of diverse DNN architectures in order to be able to rank, select and combine multiple hypotheses after an effective DNN adaptation stage; Figure 1 shows the blocks detailed in Section3.

In particular, we explored the use of *batch-normalized DNNs*. Training DNNs is indeed complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This problem, known as internal covariate shift, slows down the training of deep neural networks. Batch normalization [11] addresses this issue by normalizing the mean and the variance of each layer for each training mini-batch, and back-propagating through the normalization step. It has been long known that the network training converges faster if its inputs are properly normalized [12] and, in such a way, batch normalization extends the normalization to all the layers of the architecture. However, since a per-layer normalization may impair the model capacity, a trainable scaling parameter γ and a trainable shifting parameter β are introduced in

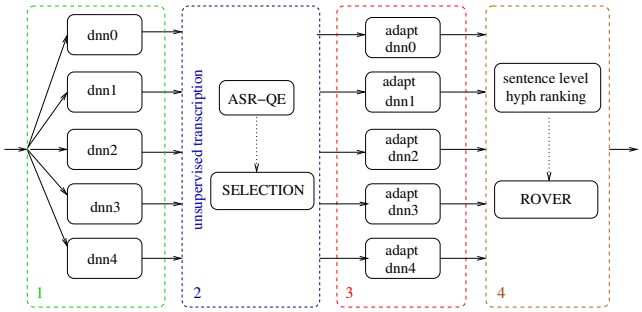

Figure 1: The architecture of the proposed CHiME-4 automatic transcription system, characterized by a four-steps pipeline.

each layer to restore the representational power of the network. The above-mentioned systems, implemented with Theano [13], are coupled with the Kaldi toolkit [14] to form a context-dependent DNN-HMM speech recognizer.

Another technique explored in this work is *DNN adaptation*. The usual way to adapt a DNN trained on a large set of data, given a much smaller set of adaptation data, is to retrain the DNN over the adaptation set, which could lead to overfitting the model on the adaptation data. A solution to prevent these detrimental effects is to adopt a conservative learning procedure by adding a regularization component to the loss function. The adaptation technique proposed here is based on a Kullback-Leibler divergence (KLD) regularization [15]. KLD regularization can be implemented through cross-entropy minimization between a new target probability distribution and the current probability distribution. Moreover, this regularization binds directly the DNN output probabilities rather than the model parameters; as a consequence, the method can be easily implemented with any software tool based on back-propagation, without introducing any modification.

In addition, we evolved our previous system by exploiting a recently developed *automatic quality estimator* (QE), which is able to provide (sentence by sentence) a confidence score related to the expected word error rate (WER%). Automatic assessment methods can be used to select audio data for unsupervised training [16], active learning of acoustic models [17, 18], combination of multiple transcription hypotheses into a single and more accurate one [19]. The proposed technique, which has shown promising in both ASR and machine translation applications [20, 10], contributed to this submission in two ways. First, we used the confidence scores to automatically select the best subset of utterance for the unsupervised adaptation step. Secondly, we exploit such a confidence score to rank multiple hypothesis prior to a standard system combination based on ROVER, as done in our previous submission.

3. System implementation

The architecture of our proposed system, depicted in Fig.1, is based on four steps: generation of preliminary transcriptions using the models trained on the noisy channels; quality estimation of the resulting hypotheses and selection of suitable adaptation sentences according to WER predictions; DNN adaptation using KLD regularization; systems combination through ROVER.

3.1. Step 1: multiple DNN-based speech recognizers

With the final purpose of improving system diversity, different DNNs have been considered. All the DNNs use the standard 40 fMLLR features used in the CHiME-4 baseline recipe [4]. Such features are then gathered into a context windows of 11 consecutive frames prior to feeding a feed-forward DNN. A Stochastic Gradient Descend (SGD) algorithm is used as DNN optimizer.

A first system (*dnn0*) based on the CHiME-4 baseline has been trained using one single channel (CH5), as originally proposed. A second DNN (*dnn1*), is trained following again the baseline recipe but exploiting all the six channels available in the training-set (CH1-CH6).

In addition, a set of batch-normalized DNNs are trained (*dnn2-4*). For these systems (due to time and computational restrictions) the standard training-set (based on channel 5 only) was used. The adopted batch-normalized DNNs are based on Rectified Linear Units (ReLU) and employ drop-out (with a drop-out rate of $\rho = 0.2$). Moreover, to further improve the system performance, the labels for DNN training are derived by a forced-alignment over the close-talking signals. Such an approach has been studied in [21]. The first batch-normalized DNN (*dnn2*) is based on six hidden layers composed of 2048 neurons. A second batch-normalized DNN (*dnn3*) is trained with the same architecture, but exploiting features derived by an automatic classification of the environment. More specifically, a DNN is trained using the environmental labels in the training set and the posterior probabilities generated by such a network are concatenated with the standard fMLLR features. The last batch-normalized DNN (*dnn4*), inspired by our recent work on joint training [22], concatenates a speech enhancement and a speech recognition deep neural network, whose parameters are jointly updated as if they were within a single bigger network. More precisely, in the joint training framework we perform a forward pass, compute the loss functions at the output of each DNN (mean-squared error for speech enhancement and cross-entropy for speech recognition), compute the corresponding gradients, and back-propagate them though.

Particular attention should be devoted to the initialization of the γ parameter. Contrary to [11], where it was initialized to unit variance ($\gamma = 1$), in this work we have observed better performance and convergence prop-

erties with a smaller variance initialization ($\gamma = 0.1$). A similar outcome was found in [23, 24], where fewer vanishing gradient problems are empirically observed with small values of γ in the case of recurrent neural networks.

3.2. Step 2: Quality Estimation

The transcriptions generated by each hybrid DNN-HMMs systems are processed by a system that automatically estimates the WERs of each sentence. The approach makes use of a supervised regression method that effectively exploits a combination of "glass-box" and "black-box" features [20, 10]. Glass-box features, similar to confidence scores, refer to the one extracted when the ASR features such as lattice and confidence scores are available, and capture information inherent to the inner workings of the ASR system that produced the transcriptions. The black-box ones, instead, are extracted by looking only at the signal and the transcription. On one side, they try to capture the *difficulty* of transcribing the signal while, on the other side, they try to capture the *plausibility* of the output transcriptions. In both cases, the information used is independent of knowledge about the ASR system, making the approach of [20] ASR QE applicable to a wide range of scenarios in which the only elements available for quality prediction are the signal and the transcription. The extensive experiments in different testing conditions discussed in [20, 10] indicate that regression models based on Extremely Randomized Trees (XRT) [25] can achieve competitive performance, being able to outperform strong baselines and to approximate the true WER scores computed against reference transcripts. For the experiments reported here we trained two different XRT based regressor on the CHiME-4 development sets: dt05_simu and dt05_real, and used the resulting models on the related evaluation sets.

3.3. Step 3: DNN unsupervised adaptation

The WER predictions of the sentences in each evaluation set are hence used to build adaptation sets containing sentences of mid-high quality. In particular, for these experiments we selected all the sentences with a predicted WER below 20%. The selected material is used to perform "self" DNNs adaptation (i.e. we are using, as adaptation sets, selected subsets of the test data.

The KLD regularization introduced for the adaptation step is implemented through cross-entropy minimization between a new target probability distribution and the current probability distribution. The new target distribution is obtained as a linear interpolation of the original distribution and the distribution computed via forced alignment with the adaptation data:

$$P[s_i|o_t] = (1-\alpha)\hat{p}[s_i|o_t] + \alpha\overset{*}{p}[s_i|o_t] \quad 0 \leq \alpha \leq 1 \quad (1)$$

Note that, in Eq. 1, $\alpha = 0$ is equivalent to a "pure" retraining of the DNN over the adaptation data, while $\alpha = 1$ means that the output probability distribution of the adapted DNN is forced to follow that of the original DNN . What one can expect is that the optimal value of α is close to 0 when the size of the adaptation set is large and the transcriptions of the adaptation sentences are not affected by errors (i.e. in supervised conditions). Conversely, when the size of the adaptation set is small and/or its transcription can be affected by errors (i.e. in the case of unsupervised adaptation), α should increase.

DNNs are adapted to the acoustic conditions of each evaluation set: we adapt a different DNN for each one of the two sets: dt05 and et05. The automatic supervision of each adaptation set is given by the ASR hypotheses generated in the first decoding pass of Figure 1.

A final decoding step is then carried out using the adapted DNNs, followed by the LM rescoring pass included in the CHiME-4 baseline (based on a linear combination of 5-gram LM and RNNLM).

Table 1: Average WER (%) for the each systems and the final combination

Track	System	Dev		Test	
		real	simu	real	simu
1ch	sys0	10.42	12.54	20.09	18.17
	sys1	9.02	10.98	17.21	16.52
	sys2	9.64	11.48	18.44	17.40
	sys3	9.65	11.52	18.26	16.99
	sys4	10.02	12.92	18.62	18.23
	comb	**9.02**	**9.51**	**16.87**	**16.09**

3.4. Step 4: hypotheses combination

A common way to combine multiple ASR hypotheses is through ROVER [CITE]. However, the behaviour of ROVER strongly depends on the order of the hypotheses[CITE], and the overall performance could substantially improve if the ARS transcription are ranked according to they accuracy [10]. Therefore, the ASR transcriptions, obtained after the unsupervised DNN adaptation, are automatically ranked at sentence level using the QE system described in [10]. We train an automatic ranking system for each development data sets (dt05_simu and dt05_real), and used it to rank the sentence hypotheses of the evaluation sets: et05_simu and et05_real.

4. Experimental evaluation

4.1. Submitted system

Table 1 reports the results obtained with each subsystems and with their final combination. The systems labeled as "*sys0-4*" refer to five DNNs (*dnn0-4*) after the unsupervised adaptation. The system "*comb*" represents the final ROVER combination.

We can observe that, as expected, the best single sys-

Table 2: WER (%) per environment for the submitted system

Track	Envir.	Dev		Test	
		real	simu	real	simu
1ch	BUS	12.41	8.01	24.57	12.01
	CAF	8.70	12.12	18.36	18.36
	PED	6.23	7.30	13.60	15.57
	STR	8.73	10.59	10.96	18.23

tem is *sys1*, since it is trained with all the available channels. However, the performance obtained with batch-normalized DNNs (*sys2*-*sys4*) are rather competitive with *sys1*, even if such systems are training with a single channel only. However, the comparison between *sys0* (no batch-norm) and *sys2* (with batch norm) confirms the significant benefits obtained with such a technique. Results also reveals that the addition of the environmental features seems to give only minor benefits (compare *sys2* and *sys3*). We also found that, differently to what we experimented in [22], the joint training systems (*sys3*) performs slightly worse than a single DNN case. The last row of Table 1 reports the results obtained by combining all the considered systems. The performance obtained with the latter system for each noise conditions is reported in Table 2.

4.2. Updated system

The importance of the quality of automatic transcriptions for the adaptation pass suggested us to introduce a modification in the system architecture, i.e. to make use of an additional combination stage after the initial decoding step; indeed, it is possible to automatically rank [10] also the hypotheses generated in the pass-1 and select the "best" one as supervision for all the systems in pass-3. Table 3 shows the results obtained with this new adaptation strategy, represented in Figure 2. An additional gain is achieved, indicating that the improved transcription obtained exploiting the diversity of multiple systems produces better adapted DNN models.

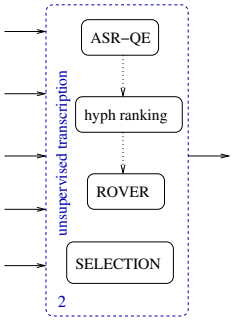

Figure 2: The updated pass-2: a unique QE-based supervision is derived for all the DNN systems.

Table 3: Average WER (%) for the updated system in which the pass-2 produces a single supervision for all the DNN systems.

Track	System	Dev		Test	
		real	simu	real	simu
1ch	comb (new)	**8.45**	**10.56**	**16.17**	**15.20**

5. Discussion and conclusions

In this work we have proposed a refinement of the system previously submitted to the CHiME-3 challenge [9]. The *two-pass decoding* combined with *automatic data selection* for DNN adaptation benefited from previous experience on quality estimation of ASR hypotheses in the framework of ASR system combination [10].

To perform data selection we applied ASR quality estimation, using automatic WER prediction as a criterion to isolate subsets of the adaptation data featuring variable quality. As a result, ASR QE-based data selection, in combination with KLD-based DNN adaptation, provides a significant advantage. Instead, the diversity of the hypotheses generated by DNNs trained on different channels or with different procedure (batch-normalization) is quite limited and the final combination step provides small improvements with respect to the single systems.

Overall, the experimental results confirm the effectiveness of the proposed approach that, using the provided training set and the baseline language models, allows to improve from 22.3% to 16.5% WER (average on the evaluation set).

Finally, note that the regularization coefficient α in Eq. 1 can be made dependent on the quality of each test sentence (e.g., by predicting the corresponding WER or by implementing a specific training phase for estimating sentence dependent α_k, being k the identifier of the k^{th} test utterance) allowing to implement a soft scheme for DNN adaptation: this approach has given promising results on the recognition of a data set of children speech [26].

A planned direction for further investigations is the introduction of more effective types of neural network architectures (Convolutional Neural Networks or Long-Short Term Memory Recurrent Neural Networks [27]), both for improving the overall performance of the related ASR systems and for augmenting the diversity of the hypotheses. In this way both the quality of the supervision and the efficacy of hypotheses combination are expected to increase.

6. References

[1] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME Speech Separation and Recognition Challenge," *Computer Speech & Language*, vol. 27, no. 3, pp. 621–633, 2013.

[2] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The Second CHiMEspeech Separation and Recognition Challenge: An Overview of Challenge Systems and Outcomes," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Olomouc, Czech Republic, 2013, pp. 162–167.

[3] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines," in *Proc. of the 15th IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, Arizona, USA, 2015, pp. 1–9.

[4] E. Vincent, S. Watanabe, A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language, to appear*, 2016.

[5] G. Hinton, L. Deng, D. Yu, and Y. Wang, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 9, no. 3, pp. 82–97, 2012.

[6] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Trans. on Audio Speech and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[7] P. Swietojanski and S. Renals, "Hybrid Acoustic Models for Distant and Multichannel Large Vocabulary Speech Recognition," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomuc, Czech Rep., 2013, pp. 285–290.

[8] S. Renals and P. Swietojanski, "Neural Networks for Distant Speech Recognition," in *Proc. of Hands-free Speech Communication and Microphone Arrays (HSCMA) Wokshop*, Villers-les-Nancy, 2014, pp. 172–176.

[9] S. Jalalvand, D. Falavigna, M. Matassoni, P. Svaizer, and M. Omologo, "Boosted Acoustic Model Learning and Hypotheses Rescoring on the CHiME-3 Task," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, Arizona, USA, 2015, pp. 409–415.

[10] S. Jalalvand, M. Negri, D. Falavigna, and M. Turchi, "Driving ROVER With Segment-based ASR Quality Estimation," in *Proc. of ACL*, Beijing, China, July 2015.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. of ICML*, 2015, pp. 448–456.

[12] Y. LeCun, L. Bottou, G. Orr, and K. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer Berlin Heidelberg, 1998, pp. 9–50.

[13] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[14] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. of ASRU*, 2011.

[15] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-Divergence Regularized Deep Neural Network Adaptation for Improved Large Vocabulary Speech Recognition," in *Proc. of ICASSP*, Vancouver (Canada), May, 26-31 2013, pp. 7893–7897.

[16] L. Lamel, J.-L. Gauvain, and G. Adda, "Investigating Lightly Supervised Acoustic Model Training," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, Salt Lake City, USA, 2001, pp. 477–480.

[17] G. Riccardi and D. Hakkani-Tur, "Active Learning: Theory and Applications to Automatic Speech Recognition," *IEEE Transaction on Speech and Audio Processing*, vol. 13, no. 4, pp. 504–511, 2005.

[18] A. Facco, D. Falavigna, R. Gretter, and V. Vigano, "Design and Evaluation of Acoustic and Language Models for Large Scale Telephone Services," *Speech Communication*, vol. 48, no. 2, pp. 176–190, 2006.

[19] J. G. Fiscus, "A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. Santa Barbara, CA, USA: IEEE, 1997, pp. 347–354.

[20] M. Negri, M. Turchi, D. Falavigna, and J. G. C. de Souza, "Quality Estimation for Automatic Speech Recognition," in *Proc. of COLING*, Dublin, Ireland, 2014.

[21] M. Ravanelli and M. Omologo, "Contaminated speech training methods for robust DNN-HMM distant speech recognition," in *Proc. of INTERSPEECH 2015*, pp. 756–760.

[22] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Batch-normalized joint training for DNN-based distant speech recognition," in *Proc. of SLT 2016*.

[23] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," *arXiv preprint arXiv:1603.09025*, 2016.

[24] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "A network of deep neural networks for distant speech recognition," in *submitted to ICASSP 2016*.

[25] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely Randomized Trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[26] M. Matassoni, D. Falavigna, and D. Giuliani, "Cross and Self Adaptation of DNN for Recognition of Children Speech," in *Proc. of SLT*, San Diego (CA), Usa, December 2016.

[27] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *Interspeech*, 2014.