BOOSTED ACOUSTIC MODEL LEARNING AND HYPOTHESES RESCORING ON THE CHIME-3 TASK

Shahab Jalalvand[1,2], *Daniele Falavigna*[1], *Marco Matassoni*[1], *Piergiorgio Svaizer*[1]
Maurizio Omologo[1]

[1] SHINE research unit, Fondazione Bruno Kessler (FBK), 38123 Povo, Trento, Italy
[2] School of International Doctorate, University of Trento, 38123 Povo, Trento, Italy
(jalalvand,falavi,matasso,svaizer,omologo)@fbk.eu

ABSTRACT

Speech recognition in a realistic noisy environment using multiple microphones is the focal point of the third CHiME challenge. Over the baseline ASR system provided for this challenge, we apply state of the art algorithms for boosting acoustic model learning and hypothesis rescoring to improve the final output.

To this aim, we first use the automatic transcription of each channel to re-train the acoustic model for that channel and then we apply linear language model rescoring to find a better solution in the n-best list. LM rescoring is performed using an efficient set of N-gram and Recurrent Neural Network LM (RNNLM) trained on a wisely-selected text set.

In the experiments, we show that the proposed approach improves not only the individual channel transcription, but also the enhanced channels produced by MVDR and delay-and-sum beamforming.

Index Terms— Speech enhancement, machine learning algorithms, adaptive estimation.

1. INTRODUCTION

Automatic speech recognition (ASR) in hands-free conditions is gaining increasing interests recently, for instance in home and office automation, smart cars, humanoid robots. In such applications, ASR should operate in environments where noises of various types, competing speakers and reverberation effects heavily affect the performance, usually satisfactory in controlled acoustic conditions. A popular approach is based on microphone arrays that allow the implementation of several enhancement techniques: beamforming, denoising and dereverberation [1].

In this paper we describe the ASR system developed for the CHiME-3 challenge, where noisy utterances are recorded by a 6-channel tablet-based microphone array. As addressed in the past by CHiME evaluation campaigns, the recognition task is the automatic transcription of read sentences from the Wall Street Journal (WSJ) corpus, acquired in several noisy conditions. The reader is referred to the paper [2] presented by the challenge organizers for the description of training, development and evaluation data sets released for the competition.

To develop the system we used the Kaldi open source toolkit [3] based on hybrid acoustic model: a deep neural network (DNN) [4, 5, 6, 7] estimates posterior probabilities that replace the emission probabilities given by Gaussian Mixture Models (GMMs) associated to the states of context dependent Hidden Markov Models (HMMs). To train acoustic models (AMs) we exploited the procedure released by the CHiME-3 organizers, with the addition of a final decoding step that employs a DNN trained in an unsupervised way.

For language model (LM) training we used the data set required by the challenge rules, however we carried out a final re-scoring step over n-best hypotheses using a linear combination of several LMs, including recurrent neural network language models (RNNLMs) [8, 9].

The main contributions of our submitted system, over the baseline, are: 1) the enhancement technique based on delay-and-sum (DS) contrasting it with the minimum variance distortionless response (MVDR) beamformer provided as baseline; 2) the DNN re-training using observation labels derived from automatic transcriptions of the evaluation sets. 3) Moreover, we carried out re-scoring of n-best lists with a linear combination of different LMs trained both over the complete set of official textual data and over a set of *task specific* documents, automatically extracted from the same set of training data. 4) The final hypotheses are obtained applying selection, at segment level, of rescored hypotheses given by a single channel and by the two enhanced signals.

The "multi-pass" decoding approach proposed here is methodologically similar to the HMM/GMM adaptation with MLLR transformations [10], although in our case a complete retraining of DNN is performed using the automatic transcription of the evaluation set. However, it has to be noticed that DNN adaptation can be carried out through other different semi-supervised learning approaches such as the ones described e.g. in [11, 12, 13, 14].

The paper is organized as follows: Section 2 introduces the proposed architecture, presenting some baseline results. The procedure for LM training is described in Section 3, while Section 4 describes the techniques for hypotheses selection and n-best list rescoring. In Section 5 the experimental results are reported and discussed in Section 6. Finally, Section 7 summarizes the major findings of the work, introducing possible directions for future investigations.

2. SPEECH RECOGNITION SYSTEM

Figure 1 shows the complete recognition system organized in the multiple blocks described below.

- Block 1. The multi-channel front-end provides a number of signals either selecting a favorable channel (e.g., CH5) or applying some enhancement techniques; in our system, beside the provided baseline algorithm based on MVDR [2, 15], we use a simple delay-and-sum beam-former applied to the five frontal microphones (see Section 2.1).
- Block 2. The first decoding pass is carried out with the recognizer [16] included in the framework of CHiME-3 challenge, based on hybrid acoustic model (GMM/DNN) and trained on the provided matched data (note that we don't make use of additional training data for acoustic modelling).
- Block 3. The three sets of hypotheses produced in the first decoding step (one for each of CH5, MVDR and DS enhanced signals) are fused into a single transcription by selecting, utterance by utterance, the one that exhibits the maximal posterior probability. The final transcription (which is assumed to have a word error rate lower than that of each single transcription) is used as supervision for both re-aligning the acoustic observations of the input streams to transcribe and re-assigning the output labels of DNNs.
- Block 4. The DNNs are re-trained with the latter "unsupervised" labels. The input acoustic observations are recognized in a second decoding step with the new retrained DNNs and for each input stream a corresponding set of n-best lists is generated.
- Block 5-6. N-best lists of each set are both re-scored and re-ordered with a linear combination of language models (see Section 4) in order to provide 1-best transcriptions to the final module (Block 6 in Figure 1) that, similarly to Block 3, selects the utterances with the highest posterior probabilities.

2.1. Multi-channel processing

We tested different beamforming methods, including coherence-based weighting and post-filter schemes, and compared the ASR results with those obtained by means of the MVDR processing (with diagonal loading) provided by the speech enhancement baseline. Working with the development data set, we found, quite surprisingly, that the best performance on real data was obtained with a simple beamforming consisting in uniform weighting of the rephased signals of the 5 frontal microphones. According to this approach, the beamformed signal $Y(\omega, t)$ is obtained from the vector $\mathbf{X}(\omega, t)$ of microphone signals as

$$Y(\omega, t) = \mathbf{V}^T(\omega, t)\mathbf{X}(\omega, t) \tag{1}$$

i.e. the array is steered by means of the vector

$$\mathbf{V}(\omega, t) = \begin{bmatrix} g_1 e^{-j\omega\tau_1(t)} \\ g_2 e^{-j\omega\tau_2(t)} \\ \vdots \\ g_M e^{-j\omega\tau_M(t)} \end{bmatrix} \tag{2}$$

where $M = 6$ is the number of microphones, τ_i are the estimated times of flight from the source to the i-th microphone (as provided in the baseline front-end processing) and g_i are boolean coefficients determining whether the i-th channel has to be included or not in the beamforming. Specifically, g_2 is always 0 in our case, as we exclude microphone 2, while for $i \neq 2$ it is $g_i = 1$ unless a microphone failure is detected. We replaced the baseline failure test based on energy with one based on the inter-channel coherence.

As reported in Section 5, this simple standard beamforming outperformed the proposed baseline algorithm in the case of real data. In principle more sophisticated beamforming schemes [17] are expected to produce benefits in terms of directivity and array-gain, but robustness in adaptive beamforming is achieved only if the appropriate statistics of the desired signal and of the interferers are known, or can be measured reliably from the data. This in turns requires that the signal and noise processes can be assumed to be stationary in the short term, and that the observation interval is long enough.

On the other side, delay-and-sum beamforming, although less effective in rejecting directional noise and in suppressing stationary interferers, is quite robust to small steering mismatch and to varying conditions, leading to lower distortion in the signal passed to the recognizer.

2.2. System baselines

The speech recognizers used for producing the multiple hypotheses are directly derived from the provided baseline, represented by GMM-based and DNN-based systems, as detailed in [2]. The acoustic features are 13 mel-frequency cepstral coefficients (MFCCs), sliced by 3 frames next to the central frame and projected down to 40 dimensions using linear discriminant analysis (LDA) and Maximum Likelihood Linear

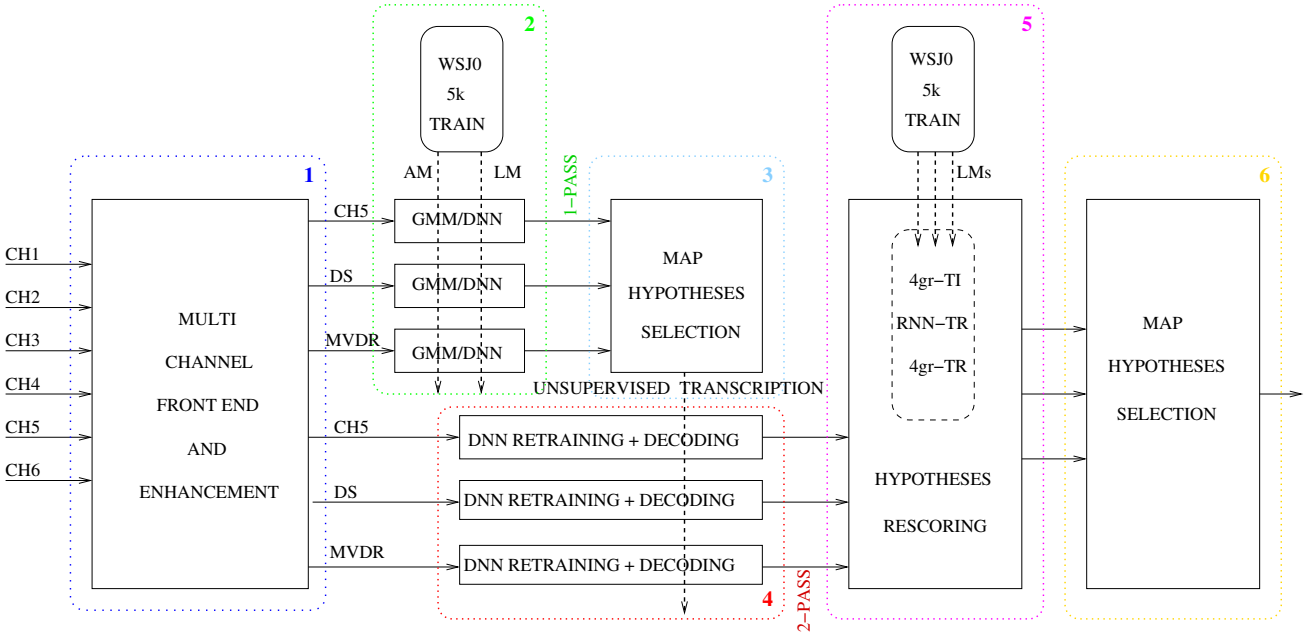

Fig. 1. Architecture of the proposed multi-pass recognition system, characterized by six main processing blocks.

Transform (MLLT). A single feature-space Maximum Likelihood Linear Regression (fMLLR) transform for each training speaker is then estimated along with speaker-adaptive training (SAT) for building triphone HMMs, represented by 2,500 tied-states and 15,000 Gaussians. The DNN system is trained using the Karel's setup [18] included in the KALDI toolkit. An 11 frames context window (5 frames at each side) is used as input to form a 440 dimensional feature vector. The DNN have 7 hidden layers each with 2048 neurons. The DNN is trained in several stages including Restricted Boltzmann Machines (RBM) pre-training, mini-batch Stochastic Gradient Descent training, and sequence-discriminative training using state-level Minimum Bayes Risk (sMBR).

Models	Input	dt05		et05	
		real	simu	real	simu
GMM	CH5	18.63	18.60	33.22	21.89
GMM	DS	12.26	14.42	22.60	23.33
GMM	MVDR	20.00	9.78	37.26	11.07
DNN	CH5	16.80	14.54	32.61	20.22
DNN	DS	10.28	12.04	20.23	24.79
DNN	MVDR	17.58	8.23	33.14	10.97

Table 1. WERs achieved with different input streams (CH5, DS and MVDR) and with different acoustic (GMM, DNN) models on all tasks.

Table 1 gives the percentages of word error rates (WERs) achieved on the different development and test sets (hereinafter we will refer them as "tasks") with the baseline systems (both GMM and hybrid based) fed with acoustic observation stemming from CH5 single channel and from both MVDR and DS based beamformers.

As one can note in Table 1 there are relevant differences in performance along all the tasks, showing a sort of complementary behaviour among the three input streams. Especially DS exhibits better performance than MVDR based one on real tasks, while the opposite holds for simulated tasks. This fact will be discussed in Section 6.

3. LANGUAGE MODELS

For language modelling we decided to make use of both n-gram LM and RNNLM. The latter type of LM has been proven to be effective both for decoding acoustic observations [12] and, in combination with traditional n-gram LMs, for rescoring the ASR hypotheses [9, 8]. The main advantage of RNNLM is its capability to capture a long history (thanks to its recursive links) to compute the conditional probability of a word, unlike the traditional n-gram LM, in which the length of the history cannot exceed n-1 words.

The LMs used in the re-scoring steps are trained either over all of textual documents provided for the CHiME-3 competition or on a subset of them. The whole training set (named *np_data*) comprises around 37 millions (37M) of words (belonging to the WSJ0 corpus) from which also the baseline 3-gram LM, employed in the frame based decoding passes, is

trained. Then, given the latter corpus the following LMs are built.

- A "task independent" 4-gram LM (named **4gr-TI**) trained over the whole *np_data* material using the Kneser-Ney discounting method [19].
- A *task related* RNNLM (**RNN-TR**), as detailed below, using a subset of whole *np_data* training corpus, containing around 11M words. The latter subset was automatically selected after having ordered the sentences in the *np_data* corpus according to their perplexity values, given by a 3-gram LM trained over the automatic transcriptions of the dt05-real development set. This latter training subset is the one given by the first decoding pass of the DS input stream of Figure 1.
- 12 "task related" 4-gram LMs (**4gr-TR**), one for each task and input stream. To accomplish this task, we trained a 3-gram LM over the automatic transcriptions produced in the first decoding pass of each input stream and computed, first, the perplexity of each sentence contained in *np_data*. Then, we reordered all training sentences according to the resulting perplexity values, and extracted the top ones in order to build, for each task and for each stream, a "task related" corpus, containing around 10M words, over which the corresponding, above mentioned, **4gr-TR** LM was trained.

All of the n-gram LMs described above were trained using the IRST-LM toolkit described in [20]. In addition we tried, as an alternative to the sentence perplexity measure, a scoring function based on the n-gram ratio [21] which gave worse performance on the "real" development set (i.e. dt05-real).

To build RNNLMs we used the toolkit provided in [22], while optimization of some parameters (basically number of hidden neurons and the number of output classes) was carried out on the development set.

Table 2 (first four rows) gives some statistics of each of the LM training text corpora, including LM perplexity values, computed on dt05-real, corresponding to task-independent and "DS" task-related LMs.

Note that according to the above description all LMs included in Table 2, except 4gr-TR, are common to all tasks and input streams, while 4gr-TR LMs are specific to both input streams and tasks.

As one can expect, 4gr-TI (task independent) LM gives a lower perplexity value than the baseline LM (3gr-TI) and a higher perplexity than task related LMs, both 4gr-TR and RNN-TR (this demonstrates the effectiveness of the proposed automatic selection procedure). Also important to notice, looking at last two rows of Table 2, is the perplexity gain achieved using the LM linear combination approach described in the next section, that will reflect into corresponding reduction in word error rate, as will be shown in Section 5.

Info LM	# Training Words	# Training Vocab	PPL on dt05-real
3gr-TI(baseline)	37M	165K	119.2
4gr-TI	37M	165K	107.8
RNN-TR	10M	48K	109.5
4gr-TR	10M	48K	95.0
4gr-TI⊕RNN-TR	–	–	52.4
4gr-TI⊕RNN-TR ⊕4gr-TR	–	–	50.2

Table 2. Statistics of language models used in the rescoring phase and related perplexity (PPL) values.

4. N-BEST LISTS RESCORING

As shown in Figure 1, in the first decoding pass, employing DNN based AM, we generate n-best lists from word lattices (for the experiments reported in this paper we use a value of $n = 100$). Then, n-best hypotheses are re-scored, at utterance level, via a linear combination of LMs. In generating n-best lists, using the KALDI commands, we remove from every entry in the list the contribution of the baseline 3-gram LM, keeping only the information related to the AM. The AM log-likelihood (AM_S) is scaled with the LM weight (A_{LM}) and summed to a score given by a linear combination of LM log-probabilities of each utterance in the list.

In the experiments reported below we used several combinations of LMs for computing re-scored log-probabilities $\log P^{res}[w_1 \dots w_K \mid \mathbf{O}]$ for each hypothesis $W = w_1 \dots w_K$:

$$\begin{aligned} \log P^{res}[W \mid \mathbf{O}] = & \quad \frac{AM_S}{A_{LM}} + \sum_i \lambda_i \log P_{LM_i}[W \mid \mathbf{O}] \\ = & \quad \log P^{res}_{LM_1 \oplus \dots \oplus LM_I}[W \mid \mathbf{O}] \end{aligned} \tag{3}$$

where LM_i indicates one of the LMs described in Section 3 and the interpolation coefficients λ_i can be estimated in order to optimize some objective function (e.g the word error rate or the perplexity) on a development set. Note in equation 3 the notation $LM_1 \oplus \dots \oplus LM_I$ which refers to the LMs entering the linear combination and that will be used in the experimental section. Then, the resulting re-scored n-best lists are reordered and the best hypothesis is sent to the evaluation procedure.

As a final step, given some experience gained in the past [23, 24] on segment based system combination, we decided to apply a procedure that automatically selects among the best transcriptions produced by separate ASR systems (e.g. the ones related to the three different input streams in Figure 1) the one that gives the "best score" at segment level. In the above mentioned paper we experimented several scoring functions for selecting the best segment transcriptions, as well as different approaches for ranking them in order to feed a combination system based on ROVER [25, 24]. For this work,

	CH5 1-step	MVDR 1-step	DS 1-step	CH5 2-step	MVDR 2-step	DS 2-step	Avg of Gains	Gain on Best
Baseline	16.8	17.6	10.3	10.3	11.5	8.3	0.0	0.0
4gr-TI	15.8	16.7	9.8	10.1	11.1	7.8	-0.6	-0.5
4gr-TI⊕RNN-TR	14.3	15.6	8.7	9.3	10.2	7.1	-1.6	-1.2
4gr-TI⊕4gr-TR⊕RNN-TR	14.3	15.6	8.6	9.2	10.2	7.1	-1.6	-1.2

Table 3. Performance (%WER) achieved on the real development set dt05-real for separated input streams CH5, MVDR and DS.

	CH5 1-step	MVDR 1-step	DS 1-step	CH5 2-step	MVDR 2-step	DS 2-step	Avg of Gains	Gain on Best
Baseline	14.4	8.2	12.1	9.0	6.7	7.5	0.0	0.0
4gr-TI	14.1	8.3	11.7	8.9	6.5	7.4	-0.2	-0.1
4gr-TI⊕RNN-TR	12.4	7.0	10.3	8.0	5.9	6.7	-1.3	-0.8
4gr-TI⊕4gr-TR⊕RNN-TR	12.7	7.1	10.3	7.9	6.0	6.8	-1.2	-0.7

Table 4. Performance (% WER) achieved on the simulated development set dt05-simu for separated input streams CH5, MVDR and DS.

however, we decided to only perform some simple experiments based on the usage of both sentence posterior and word posterior probability values for scoring the different utterance hypotheses. The result that we found is that maximization of the sentence posterior (i.e., the weighed product of LM and AM probabilities) gives the best performance. Note that the latter approach is the same employed in the MAP selection module depicted in Figure 1 (Block 3).

5. EXPERIMENTS AND RESULTS

Tables 3 and 4 report the results, in terms of word error rates (WER), obtained on development data sets for the two conditions, real and simulated, respectively. The results, according to Figure 1, are given for: the best performing single channel (CH5), the MVDR and DS enhanced input streams. In addition, performance are given for both the first and second decoding pass, carried out as described in Section 2.

Rows of Tables 3 and 4 refer to the usage of various linear combination of LMs, according to equation 3, for rescoring n-best lists. Finally, the last two columns of the Tables give the average (among all transcriptions) WER reduction w.r.t. the baseline system, as well as the improvement w.r.t. the best baseline transcription.

In a similar way Tables 5 and 6 give performance obtained on real and simulated evaluation set, respectively.

Note that WERs values given in the Tables were computed using the Speech Recognition Scoring Toolkit (SCTK, version 2.4.0) distributed by NIST[1], which gives, according to our observations, slightly worse performance than the scoring scripts employed in the KALDI toolkit. Furthermore, although not explicitly shown in the Tables for clarity reasons, the "task related" LM (4gr-TR) is specific of both tasks and input streams. This means that each WER value reported in the last rows of the Tables was obtained with a system setup that depends also (but not only) on real and simulated conditions and, therefore, the corresponding transcriptions are not compliant with the rules of CHiME-3 challenge (i.e. *no different tuning can be used for real and simulated data*). Therefore, the setup used to generate our final submissions, common to both real and simulated data, corresponds to the results shown in the third rows (the ones referred with 4gr-TI⊕RNN-TR) of Tables 3, 4, 5 and 6 (we remind that RNN-TR, although trained on task-dependent data, is shared among tasks and input streams). Finally, each group of three transcriptions, resulting from the second decoding steps of each input stream (i.e. CH5, MVDR and DS), is sent to the MAP selection module (Block 6 of Figure 1) to produce the final submissions, whose performance are given in Table 7.

[1] see http://www.itl.nist.gov/iad/mig/tools/ for detailed information

6. DISCUSSION

First row of Tables 3, 4, 5 and 6 gives the performance obtained with the baseline systems described in Section 2.

Comparing first row of the Tables with the second one, related to a re-scoring phase using only the task independent 4-gram LM trained over all available text data, we notice a general performance improvement. This is in accordance with the perplexity values shown in Table 2. Re-scoring using the linear combination of 4gr-TI and RNN-TR LMs (4gr-TI⊕RNN-TR) gives further significant improvements for all tasks and input streams (this is still in accordance with perplexity values given in Table 2). However, we point out that, although the weights for the linear combination should be optimized on some validation set (see section 4), for this work we assign the same value to each of them, only imposing the constraint that their sum is one. Last rows of Tables refer to the usage of

	CH5 1-step	MVDR 1-step	DS 1-step	CH5 2-step	MVDR 2-step	DS 2-step	Avg of Gains	Gain on Best
Baseline	32.6	33.1	20.2	19.2	20.0	15.5	0.0	0.0
4gr-TI	30.7	31.5	18.8	17.9	18.7	14.3	-1.5	-1.2
4gr-TI⊕RNN-TR	29.5	30.1	17.7	17.3	17.9	13.5	-2.4	-2.0
4gr-TI⊕4gr-TR⊕RNN-TR	29.6	29.9	17.5	17.2	17.9	13.4	-2.5	-2.1

Table 5. Performance (%WER) achieved on the real evaluation set et05-real for separated input streams CH5, MVDR and DS.

	CH5 1-step	MVDR 1-step	DS 1-step	CH5 2-step	MVDR 2-step	DS 2-step	Avg of Gains	Gain on Best
Baseline	20.2	11.0	24.8	10.9	8.1	11.4	0.0	0.0
4gr-TI	18.9	10.0	22.9	10.0	7.5	10.6	-1.1	-0.8
4gr-TI⊕RNN-TR	17.6	9.2	21.8	9.4	7.0	10.0	-1.9	-1.4
4gr-TI⊕4gr-TR⊕RNN-TR	17.8	9.0	21.6	9.3	6.9	9.8	-2.0	-1.6

Table 6. Performance (%WER) achieved on the simulated evaluation set et05-simu for separated input streams CH5, MVDR and DS.

the linear combination of all the LMs described in Section 3. As can be observed, the insertion in the combination of the task related 4-gram LM (4gr-TR) does not lead to significant decrease of the WER although, as previously mentioned, no LM weight optimization procedure was applied.

Looking at columns of Tables 3, 4, 5 and 6 we note, for the real condition (Tables 3 and 5), the superior performance of the enhancement technique based on delay and sum, w.r.t. the one (MVDR) provided in the framework of CHiME-3 challenge. On the contrary, MVDR approach exhibits better performance on the simulated conditions (Tables 4 and 6). This major effectiveness of the algorithm in simulated conditions is probably due to higher degree of stationarity with respect to real conditions. It is worth observing the "relevant" improvements achieved retraining the DNNs on alignments and labels derived from automatic supervisions. This leaves room to further investigations on both batch and incremental adaptation methods for DNNs. Finally, Table 7 reports WER values, for each task, resulting after sentence based MAP selection, as described in Section 4. These results correspond to our final submissions to CHiME-3 challenge. As can be noticed from Table 7 the selection of the *best* (according to MAP) sentence doesn't allow any improvement on all the tasks. Despite this fact, we believe that the usage of more effective system combination techniques, as those described and experimented in [23, 24], could give further improvements.

Environment	dt05		et05	
	real	simu	real	simu
BUS	8.7	5.5	17.7	6.3
CAF	6.4	7.0	14.1	7.7
PED	5.2	4.9	13.0	6.8
STR	8.8	6.2	9.2	7.7
Avg	7.2	5.9	**13.5**	7.1

Table 7. Performance (%WER) achieved with MAP utterance selection on the different tasks using 4gr-TI⊕RNN-TR LM linear combination for rescoring n-best lists (computed with the official scoring tool).

7. CONCLUSIONS

In this paper we proposed two methods for improving the output of a multiple microphone speech recognition system provided in the third CHiME challenge.

The first method is based on retraining the acoustic model, specifically the DNNs, employed to recognize each input stream (both single or beam-formed channel) using the corresponding automatic transcription generated with existing DNN based acoustic models. We observed large improvements with this approach on the various evaluation sets, acquired both in real or simulated conditions. The second approach aims to re-score n-best lists with LMs trained on "task related" text documents, automatically extracted from the general training corpus. We again noticed significant improvements w.r.t.the baseline system. Finally, we applied a MAP selection procedure, at sentence level, for producing the improved final transcriptions to submit.

Future works will address sentence level quality estimation for generating enriched supervisions for DNN retraining/adaptation (in particular, we are currently investigating incremental DNN adaptation) as well as for the investigation of multi-channel signal enhancement approaches, to be used either as alternative to, or in combination with, DS and MVDR.

8. REFERENCES

[1] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, Digital Signal Processing - Springer-Verlag. Springer, 2001.

[2] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' Speech Separation and Recognition Challenge: Dataset, task and baselines," in *Proc. of IEEE ASRU Workshop*, Scottsdale, Arizona (USA), 2015.

[3] 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. Veselỳ, "The Kaldi Speech Recognition Toolkit," in *Proc. of ASRU*, Hawaii (US), December 2011.

[4] G. Hinton, Li Deng, Dong Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, November 2012.

[5] G.E. Dahl, Dong Yu, Li Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, Jan 2012.

[6] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. of ASRU*. IEEE, 2011, pp. 24–29.

[7] S. P. Rath, D. Povey, K. Veselỳ, and J. H. Cernockỳ, "Improved feature processing for deep neural networks," in *Proc. of Interspeech*, Lyon, France, August 2013, pp. 109–113.

[8] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. H. Cernocky, "Empirical Evaluation and Combination of Advanced Language Modeling Techniques," in *Proc. of Interspeech*, Florence, Italy, August 2011, pp. 605–608.

[9] T. Mikolov, S. Kombrink, L. Burget, J. H. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proc. of ICASSP*, Prague, Czech Republic, 2011, IEEE, pp. 5528–5531.

[10] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," vol. 12, pp. 75–98, 1998.

[11] X. Li and J. Bilmes, "Regularized adaptation of discriminative classifiers," in *Proc. of ICASSP*, Toulouse, France, May, 14-19 2006.

[12] Z. Huang, G. Zweig, and B. Dumoulin, "Cache based recurrent neural network language model inference for first pass speech recognition," in *Proc. of ICASSP*, 2014, pp. 6404–6407.

[13] K. Yao, D. Yu, F. Seide, H. Su, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Proc. of ICASSP*, Kyoto (Japan), March, 25-30 2012, pp. 366–369.

[14] 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.

[15] X. Mestre and M.A. Lagunas, "On diagonal loading for minimum variance beamformers," in *Proc. of ISSPIT*, Dec 2003, pp. 459–462.

[16] C. Weng, D. Yu, S. Watanabe, and F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. of ICASSP*, Florence, Italy, May 2014.

[17] H. L. Van Trees, *Detection, estimation, and modulation theory. Part IV. , Optimum array processing*, Wiley-Interscience, New York, 2002.

[18] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative Training of Deep Neural Networks," in *Proc. of Interspeech*, Florence, Italy, August 2011, pp. 2345–2349.

[19] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. of ICASSP*, 1995, pp. 181–184.

[20] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proc. of Interspeech*, Brisbane, Australia, September 2008, pp. 1618–1621.

[21] S. Maskey and A. Sethy, "Resampling Auxiliary Data for Language Model Adaptation in Machine Translation for Speech," in *Proc. of ICASSP*, Taipei,Taiwan, April 2009, pp. 4817–4820.

[22] T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocky, and S. Khudanpur, "Rnnlm - recurrent neural network language modeling toolkit," in *Proc. of ASRU*, Hawaii (US), 2010, IEEE.

[23] M. Negri, M. Turchi, C. J. de Souza, and D. Falavigna, " Quality Estimation For Automatic Speech Recognition," in *Proc. of COLING*, Dublin, Ireland, August 2014, pp. 1813–1823.

[24] 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.

[25] J. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER)," in *Proc. of ASRU*, Santa Barbara, CA, 1997, pp. 347–352.