END-TO-END MULTI-SPEAKER SPEECH RECOGNITION

Shane Settle[1,2], Jonathan Le Roux[1], Takaaki Hori[1], Shinji Watanabe[1], John R. Hershey[1]

[1]Mitsubishi Electric Research Laboratories (MERL), USA
[2]TTI-Chicago, USA

ABSTRACT

Current advances in deep learning have resulted in a convergence of methods across a wide range of tasks, opening the door for tighter integration of modules that were previously developed and optimized in isolation. Recent ground-breaking works have produced end-to-end deep network methods for both speech separation and end-to-end automatic speech recognition (ASR). Speech separation methods such as deep clustering address the challenging *cocktail-party* problem of distinguishing multiple simultaneous speech signals. This is an enabling technology for real-world human machine interaction (HMI). However, speech separation requires ASR to interpret the speech for any HMI task. Likewise, ASR requires speech separation to work in an unconstrained environment. Although these two components can be trained in isolation and connected after the fact, this paradigm is likely to be sub-optimal, since it relies on artificially mixed data. In this paper, we develop the first fully end-to-end, jointly trained deep learning system for separation and recognition of overlapping speech signals. The joint training framework synergistically adapts the separation and recognition to each other. As an additional benefit, it enables training on more realistic data that contains only mixed signals and their transcriptions, and thus is suited to large scale training on existing transcribed data.

***Index Terms*—** deep clustering, speaker-independent multi-talker speech separation, end-to-end asr, cocktail party problem

1. INTRODUCTION

The introduction of deep learning has lead to significant performance improvements in many different domains. End-to-end deep network-based automatic speech recognition (ASR) has recently reached the state-of-the-art performance obtained by conventional hybrid systems [1, 2, 3, 4]. End-to-end ASR systems use encoder-decoder recurrent neural networks (RNNs) to directly convert sequences of input speech features to sequences of output labels without any explicit intermediate representation of phonetic/linguistic constructs. Implementing the entire recognition system as a monolithic neural network removes the dependence on ad-hoc linguistic resources. It also greatly improves the ease of discriminative training and integration with other systems.

In this paper, we exploit these properties to extend ASR to recognition of multiple overlapping speakers. Recognizing speech amidst a cacophony of multiple speakers is a longstanding challenge known as the cocktail party problem. Solving this problem would enable dramatically better technology for real-world human machine interaction (HMI). To this end, researchers have long sought an intermediate goal of single-channel speaker-independent multi-speaker speech separation, a challenging problem in its own right. However, dramatic advances have recently been made by way of the deep clustering method [5, 6, 7, 8].

Deep clustering trains a powerful deep neural network to project each time-frequency (T-F) unit to a high-dimensional embedding vector such that the embeddings for the T-F unit pairs dominated by the same speaker are close to each other, while those for pairs dominated by different speakers are farther away. The speaker assignment of each T-F unit can thus be inferred from the embeddings by simple clustering algorithms, to produce masks that isolate each single speaker. The original method using k-means clustering [5] was extended to allow end-to-end training through unfolded clustering steps using a permutation-free mask inference objective [6]. This objective was first introduced in [5] to train a network directly estimating T-F masks for comparison with deep clustering, without success. It was later adopted in [9] as the so-called permutation-invariant training (PIT). These deep learning methods demonstrate superior performance over previous attempts at speech separation, including graphical modeling [10], spectral clustering [11], and computational auditory scene analysis (CASA) [12] approaches.

Speech separation and ASR are greatly synergistic: speech separation requires ASR to interpret the speech for any HMI task. Likewise, ASR requires speech separation to work in an unconstrained environment. These two components can be trained in isolation and connected after the fact, as in [6]. However, the deep clustering training paradigm in [6] relies on signal-level ground truth references for the individual sources. In natural recordings with reverberant acoustics, such signal-level reference is generally unavailable, and the only alternative would be simulation. However, data with natural acoustics and transcriptions of the speech is readily available. This motivates combining the two systems and jointly training them for recognition. Before now, completely different types of methods were used for each task, and such a combination was more difficult to consider. Now that the best practice for both tasks has converged toward deep networks, there is little barrier to such combinations.

We develop a fully end-to-end, jointly trained deep learning system for separation and recognition of overlapping speech signals. In the joint training framework, separation and recognition are synergistically adapted to each other, leading to improved performance. Related work used a hybrid DNN/HMM architecture for ASR, rather than an end-to-end recognizer and relied on oracle alignments during training [13]. Our model avoids the use of oracle alignments, but our experiments do rely on a pre-trained separation system.

2. SPEECH SEPARATION

2.1. Deep clustering

The deep clustering approach trains a deep network to generate an embedding vector for each T-F element. The objective is to pull together the embeddings for the T-F unit pairs dominated by the same speaker, while pushing apart the embeddings of T-F pairs dominated

This work was done while S. Settle was an intern at MERL.

by different speakers[5, 6]. At test time, the T-F elements belonging to the same speaker can be inferred using a clustering algorithm on the learned embeddings.

More formally, for a mixture spectrogram with N T-F elements and C speakers we can define a label matrix $Y \in \mathbb{R}^{N\times C}$ such that $y_{i,c} = 1$, if T-F element i is dominated by source c, and $y_{i,c} = 0$ otherwise. The ith row, y_i, is thus a unit-length indicator vector for the speaker that dominates T-F element i. The ordering of the C speakers has an arbitrary permutation, whereas the *ideal affinity matrix*, YY^{T}, provides a permutation-invariant representation of the same information. This matrix $(YY^{\mathsf{T}})_{i,j} = 1$ if T-F elements i and j are dominated by the same speaker, and otherwise $(YY^{\mathsf{T}})_{i,j} = 0$. The network learns to produce a matrix $V \in \mathbb{R}^{N\times D}$ composed of unit-length D-dimensional embedding vectors v_i such that the affinity matrix VV^{T} approximates the ideal affinity matrix. At training time, deep clustering minimizes the following objective function with respect to V for each training mixture:

$$\begin{aligned}\mathcal{L}_{\text{DC}}(V, Y) &= \|VV^{\mathsf{T}} - YY^{\mathsf{T}}\|_{\text{F}}^2 \\ &= \|V^{\mathsf{T}}V\|_{\text{F}}^2 + \|Y^{\mathsf{T}}Y\|_{\text{F}}^2 - 2\|V^{\mathsf{T}}Y\|_{\text{F}}^2, \end{aligned} \quad (1)$$

where the embedding matrix $V \in \mathbb{R}^{TF\times D}$ and the label matrix $Y \in \mathbb{R}^{TF\times C}$ are respectively obtained by vertically stacking all the embedding vectors v_i and all the one hot vectors y_i in an utterance. Using powerful deep neural networks, this algorithm has obtained remarkable improvements over conventional methods on single-channel speech separation tasks [5, 6].

Our recent study [14] found that further improvements could be obtained using an alternative cost function based on whitening the embeddings in a k-means objective:

$$\begin{aligned}\mathcal{L}_{\text{DC,W}}(V, Y) &= \|V(V^{\mathsf{T}}V)^{-\frac{1}{2}} - Y(Y^{\mathsf{T}}Y)^{-1}Y^{\mathsf{T}}V(V^{\mathsf{T}}V)^{-\frac{1}{2}}\|_{\text{F}}^2 \\ &= D - \text{tr}\big((V^{\mathsf{T}}V)^{-1}V^{\mathsf{T}}Y(Y^{\mathsf{T}}Y)^{-1}Y^{\mathsf{T}}V\big). \end{aligned} \quad (2)$$

As proposed in [14], we use soft weights to reduce the influence of T-F bins with very low energy at training time. We use here magnitude ratio weights W_{MR} defined as the ratio of the mixture magnitude at T-F bin i over the sum of the mixture magnitudes at all bins within an utterance: $w_i = |x_i|/\sum_j |x_j|$, where $|x|$ is the magnitude of the mixture.

2.2. Chimera++ network

Permutation-free objectives have been used in several papers [5, 6, 15] to train conventional mask-inference (MI) networks for speech separation. While these objective functions were originally based on the magnitude spectrum approximation (MSA), [16] showed that the phase-sensitive spectrum approximation (PSA) outperforms MSA for separating speech from non-stationary interference, we use a (truncated) PSA objective similarly to [15].

In [14], we found that using a logistic sigmoid activation for the last layer together with an objective function measuring a truncated phase-sensitive approximation using the L_1 distance led to the best results among MI networks:

$$\begin{aligned}&\mathcal{L}_{\text{MI,tPSA},L_1} = \\ &\min_{\pi\in\mathcal{P}} \sum_c \Big\| \hat{M}_c \circ |X| - \text{T}_0^{|X|}\left(|S_{\pi(c)}| \circ \cos(\theta_X - \theta_{\pi(c)})\right) \Big\|_1, \end{aligned} \quad (3)$$

where $\mathcal{P}$ is the set of permutations on $\{1, \ldots, C\}$, $|X|$ and θ_X are the magnitude and phase of the mixture, $\hat{M}_c$ the c-th estimated mask, $|S_c|$ and θ_c the magnitude and phase of the c-th reference source, and $\text{T}_a^b(x) = \min(\max(x, a), b)$.

In [17], a chimera network is introduced that combines deep clustering with mask inference in a multi-task learning fashion, leveraging the regularizing property of the deep clustering loss and the simplicity of the mask-inference network. In the original chimera network, the mask inference branch grows out from the embedding layer. In [14], we proposed to use an improved architecture, referred to as chimera++, which predicts a mask directly from the BLSTM hidden layer output, yielding a conceptually simpler and computationally faster network. The speaker separation loss we are minimizing is a weighted sum of the deep clustering loss and the MI loss:

$$\mathcal{L}_{\text{ss}} = \alpha_{\text{DC}}\mathcal{L}_{\text{DC,W}}(V, Y) + (1 - \alpha_{\text{DC}})\mathcal{L}_{\text{MI,tPSA},L_1} \quad (4)$$

At run time, we only need the MI output to make predictions.

3. SPEECH RECOGNITION

We review the hybrid CTC/attention architecture which we introduced in [18, 3, 19] to better utilize the strengths and mitigate the shortcomings of each approach.

3.1. Connectionist temporal classification (CTC)

CTC [20] maps an input sequence to an output sequence of shorter length. We assume here that the input to our model is a T-length sequence of frame activations $X = \{x_t \in \mathbb{R}^d | t = 1, \cdots, T\}$ and the output is an L-length character sequence $C = \{c_l \in \mathcal{U} | l = 1, \cdots, L\}$ from a set of distinct characters $\mathcal{U}$. CTC introduces a "blank" symbol to give a one-to-one correspondence between inputs X and outputs $Z = \{z_t \in \mathcal{U} \cup \texttt{<blank>} | t = 1, \cdots, T\}$. By using conditional independence assumptions, the posterior distribution $p(C|X)$ can then factorized as follows:

$$p(C|X) \approx \underbrace{\sum_Z \prod_t p(z_t|z_{t-1}, C) p(z_t|X)\, p(C)}_{\triangleq p_{\text{ctc}}(C|X)} \quad (5)$$

The CTC objective is defined as $\mathcal{L}_{\text{ctc}} = -\log p_{\text{ctc}}(C|X)$, which does not include the language model $p(C)$.

We use a stacked BLSTM network to obtain the framewise posterior distribution $p(z_t|X)$ conditioned on all inputs X:

$$p(z_t|X) = \text{Softmax}(\text{Lin}(\mathbf{h}_t)) \quad (6)$$

$$\mathbf{h}_t = \text{BLSTM}(X). \quad (7)$$

3.2. Attention-based encoder-decoder

Attention-based methods use the chain rule to directly estimate the posterior $p(C|X)$ without making conditional independencce assumptions as with CTC:

$$p_{\text{att}}(C|X) = \prod_l p(c_l|c_1, \cdots, c_{l-1}, X). \quad (8)$$

We define $\mathcal{L}_{\text{att}} = -\log p_{\text{att}}(C|X)$ as the attention-based objective.

In Eq. (8), $p(c_l|c_1, \cdots, c_{l-1}, X)$ is obtained by

$$p(c_l|c_1, \cdots, c_{l-1}, X) = \text{Decoder}(\mathbf{r}_l, \mathbf{q}_{l-1}, c_{l-1}) \quad (9)$$

$$\mathbf{h}_t = \text{Encoder}(X) \quad (10)$$

$$a_{lt} = \text{Attention}(\{a_{l-1}\}_t, \mathbf{q}_{l-1}, \mathbf{h}_t) \quad (11)$$

$$\mathbf{r}_l = \sum_t a_{lt}\mathbf{h}_t. \tag{12}$$

Eq. (10) converts inputs $X = \{\mathbf{x}_t\}_{t=1}^T$ into framewise hidden vectors $\mathbf{h}_t$ in an encoder network where Encoder$(X) \triangleq$ BLSTM(X). Attention$(\cdot)$ in Eq. (11) is based on a location-based attention mechanism with convolutional features, as described in [21]. A decoder network is another recurrent network conditioned on the previous output c_{l-1}, the hidden vector $\mathbf{q}_{l-1}$, and the character-wise hidden vector $\mathbf{r}_l$. We use Decoder$(\cdot) \triangleq$ Softmax(Lin(LSTM$(\cdot)$)).

3.3. Multitask learning

Attention-based models make predictions conditioned on all the previous predictions, and thus can learn language-model-like output contexts. However, without strict monotonicity constraints, these attention-based decoder models can be too flexible and may learn sub-optimal alignments or converge more slowly to desirable alignments.

In the hybrid system, the BLSTM encoder is shared by both the CTC and attention decoder networks in Eqs. (7) and (10). Unlike the attention model, the forward-backward algorithm of CTC enforces monotonic alignment between speech and label sequences during training. This approach helps to guide the system toward monotonic alignments. The multi-task objective to be minimized becomes:

$$\mathcal{L}_{\text{ASR}} = -\big(\lambda \log p_{\text{ctc}}(C|X) + (1-\lambda)\log p_{\text{att}}(C|X)\big), \tag{13}$$

with a tunable parameter $\lambda : 0 \le \lambda \le 1$.

3.4. Decoding

The inference step of attention-based speech recognition is performed by output-label synchronous decoding with a beam search. However, we also take the CTC probabilities into account to find a better aligned hypothesis to the input speech [19], i.e. the decoder finds the most probable character sequence $\hat{C}$ given speech input X, according to

$$\hat{C} = \arg\max_{C \in \mathcal{U}^*} \{\lambda \log p_{\text{ctc}}(C|X) + (1-\lambda)\log p_{\text{att}}(C|X)\}. \tag{14}$$

In the beam search process, the decoder computes a score of each partial hypothesis. During the beam search, the number of partial hypotheses for each length is limited to a predefined number, called a *beam width*, to exclude hypotheses with relatively low scores, which dramatically improves the search efficiency.

4. JOINT SPEECH SEPARATION AND RECOGNITION

To connect these network components into a joint system, we use the masks output from the chimera++ network to extract each source, from which we compute the log-mel filterbank features for recognition. In order to choose the source-transcript permutation during training, two natural options are to use either the permutation π_{sig} that minimizes the signal-level approximation error for the separated signals, or the permutation π_{asr} that minimizes the ASR loss:

$$\pi_{\text{sig}} = \arg\min_{\pi \in \mathcal{P}} \sum_c \big\| \hat{M}_c \circ |X| - |S_{\pi(c)}| \big\|_{\text{F}}^2, \tag{15}$$

$$\pi_{\text{asr}} = \arg\min_{\pi \in \mathcal{P}} -\sum_c \big(\lambda \log p_{\text{ctc}}(C_c|X_{\pi(c)}) + (1-\lambda)\log p_{\text{att}}(C_c|X_{\pi(c)})\big). \tag{16}$$

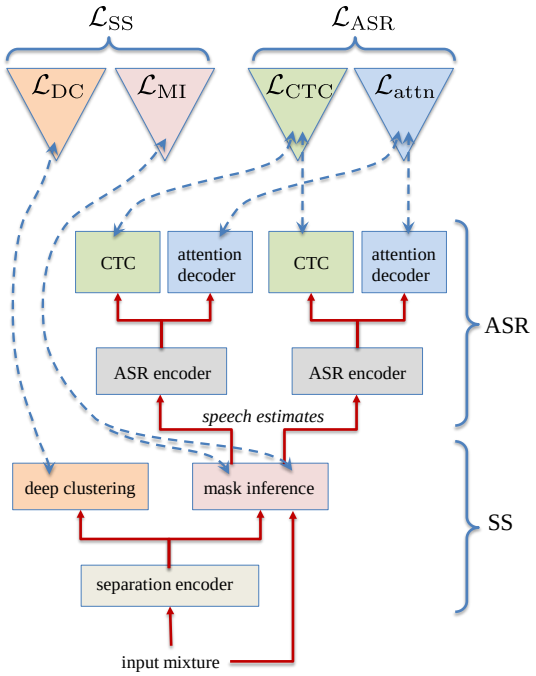

Fig. 1. End-to-end separation and recognition network

While we use π_{sig} in our experiments, π_{asr} has the advantage that it does not rely on the availability of a ground truth for the separated signals, and would thus allow for training on larger and acoustically more realistic data where only transcription-level labels are available.

5. EXPERIMENTAL SETUP

We evaluate our algorithms on the publicly-available wsj0-2mix dataset [5], which has been used by many studies since the debut of the deep clustering algorithm. It contains ∼30h training data and ∼10h validation data, both of which are created by randomly mixing two utterances of two randomly-chosen speakers from the WSJ0 training data (si_tr_s). Each mixture of the ∼5h testing data is generated by mixing two utterances from two randomly-chosen speakers in the WSJ0 validation (si_dt_05) and testing set (si_et_05). The SNR of each mixture is randomly drawn between 0 dB and 10 dB. There is no overlap between the speakers in the training and testing set. The sampling rate is 8 kHz. Our experiments consist of speech separation, speech recognition, and joint speech separation and recognition.[1]

5.1. Speech separation

The signal analysis uses a 32 ms square-root Hann window, with an 8 ms shift between frames. The 256-point DFT is performed to extract 129-dimensional log-magnitude features of each frame as inputs. The chimera++ network contains a 4-layer BLSTM network with 600 hidden units per direction (total 1200) with a dropout rate of 0.3 between layers. The deep clustering head has embedding dimension $D = 20$ following [14]. The objective is given by $\mathcal{L}_{\text{SS}}$ in Eq. (4).

Adam [23] is applied with $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$, batchsize is 16, and gradient clipping is used. After each epoch, the loss is calculated with respect to a held-out development set. If performance on the development set plateaus, η

[1] Our system is implemented using Chainer [22].

is decayed by a factor of 0.5. The model corresponding to the epoch with the lowest development set loss is evaluated on the test set.

At evaluation, the mask-inference head of the chimera++ network predicts source masks from the mixture. These masks are applied to the complex spectrum of the mixture to retrieve the predicted source signals.

5.2. Speech recognition

The recognition features are 120-dimensional log-mel filterbank coefficients+Δ+$\Delta\Delta$s. The encoder network is a 3-layer BLSTM. Each layer has 256 hidden units per direction (total 512) followed by a 256 unit projection. Subsampling is performed after the first and second layers such that an input of length T yields output of length $T/4$. The decoder network is a 1-layer LSTM with 256 hidden units. A location-aware attention scheme [21] is used with 10 convolutional filters of dimension 100. Network weights are drawn from the uniform distribution between -0.1 and 0.1. In all experiments, the CTC/Attention weight λ is set to 0.1 in $\mathcal{L}_{\text{ASR}}$. The final softmax layers in CTC and attention-based output have 51 dimensions (including characters, <sos>, <eos>, and <blank>).

Two recognition networks are considered: CLN-ASR and IBM-ASR. The CLN-ASR model is trained from clean sources, while the IBM-ASR model is trained from sources estimated by applying the ideal binary mask for each source to the mixtures.

Adadelta [24] is applied with $\rho = 0.95$ and $\epsilon = 1 \times 10^{-8}$, batchsize is 20, and gradient clipping is used. After each epoch, the loss is calculated on a held-out development set. If performance on the development set plateaus, ϵ is decayed by 0.5. The model epoch with lowest development set loss is evaluated.

During decoding and evaluation, a beam search [25] is used with a beam size of 20. The highest probability sequences output by CTC are weighted by 0.1 to help further inform the system [18]. We report results both before and after language model (LM) rescoring [3].

5.3. Joint speech separation and recognition

To facilitate joint training, networks trained as in Sections 5.1 and 5.2 warm-start the joint system for further fine-tuning. These fine-tuning experiments include: (i) training for recognition with the separation network fixed, (ii) training the whole system for both separation and recognition, and (iii) training the whole system exclusively towards the objective of speech recognition. We compare their performance with the simple combination of the two systems without fine-tuning.

When training experiments (i) and (iii), Adadelta updates the recognition network and the whole network, respectively, under the objective $\mathcal{L}_{\text{ASR}}$. For experiment (ii), two optimizers are used: Adam updates the separation network under the weighted objective $\mathcal{L}_{\text{SS+ASR}} = \mathcal{L}_{\text{SS}} + \alpha_{\text{ASR}}\mathcal{L}_{\text{ASR}}$ where $\alpha_{\text{ASR}} = 0.01$, and Adadelta updates the recognition network under the objective $\mathcal{L}_{\text{ASR}}$. Adam's $\eta = 0.0001$, but otherwise optimizer parameters are the same as in Section 5.1 & 5.2. During training, the source-transcript permutation is determined as the one that minimizes signal-level error, π_{sig} in Eq. (15). Decoding and evaluation are done as in Section 5.2.

6. EVALUATION RESULTS

We train a separation network (SS) as described above in Section 5.1, achieving an SDR of 10.7 dB, on par with our previously reported results [6, 14]. We train two recognition networks as described above in Section 5.2, one on the original clean sources (CLN) without mixing, and the other on oracle separated signals obtained by applying ideal binary masks (IBM) to the mixture. The CLN-trained model obtains 9.8 % CER when evaluated on the clean sources of the evaluation set, while the IBM-trained model obtains 11.4 % CER when evaluated on the oracle IBM-separated test mixtures. These systems provide initial guidance to the joint system to speed up training. Joint systems whose ASR part is pretrained on clean sources and IBM sources are denoted by CLN-ASR-PT and IBM-ASR-PT, respectively.

Table 1. Oracle and baseline ASR results (CER, %, no LM $\rightarrow$ with LM rescoring) for system trained and tested on clean (CLN) data, system trained and tested on data obtained by applying ideal binary masks (IBM) to the mixture, and system trained on CLN and tested on the mixtures (MIX).

training	**test**	**eval**
CLN	CLN	9.8 $\rightarrow$ 6.6
IBM	IBM	11.4 $\rightarrow$ 9.0
CLN	MIX	79.2 $\rightarrow$ 79.1

Table 2. CER Evaluation Results (no LM $\rightarrow$ with LM rescoring)

Fine-tuning			**CLN-ASR-PT**		**IBM-ASR-PT**	
SS	**ASR**	**Loss**	**dev**	**eval**	**dev**	**eval**
$\times$	$\times$	–	35.8$\rightarrow$34.1	34.5$\rightarrow$32.0	25.3$\rightarrow$24.2	25.1$\rightarrow$23.1
$\times$	✓	$\mathcal{L}_{\text{ASR}}$	17.6$\rightarrow$18.9	18.0$\rightarrow$18.0	17.4$\rightarrow$18.7	17.9$\rightarrow$17.9
✓	✓	$\mathcal{L}_{\text{SS+ASR}}$	16.7$\rightarrow$16.3	16.9$\rightarrow$15.4	15.3$\rightarrow$14.0	15.8$\rightarrow$13.9
✓	✓	$\mathcal{L}_{\text{ASR}}$	14.7$\rightarrow$**13.3**	**15.2**$\rightarrow$**13.2**	**14.4**$\rightarrow$13.6	**15.2**$\rightarrow$13.4

Table 2 shows the performance of combined separation and recognition systems under different training conditions. For all networks, the chimera++ part is initialized using the same pretrained separation network described above, while the ASR part is initialized from either CLN-ASR-PT or IBM-ASR-PT as indicated in the table. Without fine-tuning (SS: $\times$, ASR: $\times$ in Table 2), the system trained on clean data performs poorly, at 34.5% CER. The one trained on IBM data performs significantly better, with 9.4% absolute CER reduction. This indicates a clear advantage to the IBM-ASR-PT model in handling separated data, which could be expected since it was trained on binary-masked mixtures. After fine-tuning the ASR component with fixed separation (SS: $\times$, ASR: ✓, $\mathcal{L}_{\text{ASR}}$), performance improves while the gap in performance between the CLN-ASR-PT and IBM-ASR-PT models is closed significantly. Fine-tuning the whole network using both separation and recognition objectives (SS: ✓, ASR: ✓, $\mathcal{L}_{\text{SS+ASR}}$) achieves further relative improvements of 8% CER and 10% CER over the model with fixed SS for CLN-ASR-PT and IBM-ASR-PT, respectively. Finally, fine-tuning the whole network using only the recognition objective gives the best performance, at 15.2% CER for both CLN-ASR-PT and IBM-ASR-PT models, with no longer a gap in performance between the two pretraining schemes.

Language model rescoring often leads to significant improvements, of more than 2% absolute decrease in CER for the best models.

7. FUTURE DIRECTIONS

While we have shown promising results for end-to-end multi-speaker recognition, our method requires signal-level references for pretraining. Future work should investigate training on larger data with only transcription-level labels. Another interesting direction is to expand the system to work with an arbitrary number of sources, to support even more challenging and general scenarios.

8. REFERENCES

[1] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[2] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[3] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. Interspeech*, Aug. 2017.

[4] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv preprint arXiv:1612.02695*, 2016.

[5] J. R. Hershey, Z. Chen, and J. Le Roux, "Deep Clustering: Discriminative Embeddings for Segmentation and Separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[6] Y. Isik, J. Le Roux, Z. Chen, S. Watanabe, and J. R. Hershey, "Single-Channel Multi-Speaker Separation using Deep Clustering," in *Proc. Interspeech*, Sep. 2016.

[7] Z. Chen, Y. Luo, and N. Mesgarani, "Deep Attractor Network for Single-Microphone Speaker Separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Nov. 2017.

[8] ——, "Speaker-Independent Speech Separation with Deep Attractor Network," *arXiv preprint arXiv:1707.03634*, Jul 2017.

[9] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation Invariant Training of Deep Models for Speaker-Independent Multi-talker Speech Separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jul. 2017.

[10] J. Hershey, S. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Computer Speech & Language*, vol. 24, no. 1, 2010.

[11] F. Bach and M. Jordan, "Learning Spectral Clustering, with Application to Speech Separation," *The Journal of Machine Learning Research*, vol. 7, 2006.

[12] D. Wang and G. J. Brown, *Eds., Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Hoboken, NJ: Wiley-IEEE Press, Sep. 2006.

[13] D. Yu, X. Chang, and Y. Qian, "Recognizing Multi-Talker Speech with Permutation Invariant Training," in *arXiv preprint arXiv:1704.01985*, Mar. 2017.

[14] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative Objective Functions for Deep Clustering," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[15] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, 2017.

[16] H. Erdogan, J. Hershey, S. Watanabe, and J. Le Roux, "Phase-Sensitive and Recognition-Boosted Speech Separation using Deep Recurrent Neural Networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[17] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, "Deep Clustering and Conventional Networks for Music Separation: Stronger Together," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[18] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[19] T. Hori, S. Watanabe, and J. R. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Association for Computational Linguistics (ACL)*, 2017.

[20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine learning (ICML)*, 2006.

[21] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[22] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in NIPS*, 2015.

[23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.