STATIC AND DYNAMIC STATE PREDICTIONS FOR ACOUSTIC MODEL COMBINATION

Kshitiz Kumar, Yifan Gong

Microsoft Corporation, Redmond, WA

{kshitiz.kumar, yifan.gong}@microsoft.com

ABSTRACT

Acoustic model combination (AMOC) is an active research area. Model combination techniques are critical for many automatic speech recognition (ASR) scenarios, and provide frameworks to combine diverse acoustic models to boost ASR performance. We scope this work in the broad framework of AMOC, and present static and dynamic state combinations of acoustic models. We motivate and rationalize the benefits from our combination techniques, and present many applications and extensions. We apply our work in the context of combining a generic and a scenario-specific (dedicated) acoustic model; we train the proposed model with an ASR objective to best align with ASR performance. We conduct our experiments on large-vocabulary ASR task with over 30k hours of training data. Compared to generic model, we demonstrate a strong 6% word error relative reduction (WERR) in average across a variety of tasks, and specifically 25% and 8% WERR for far-field speaker and an emerging car scenario.

Index Terms— LSTM, Model Combination, Digital Assistant, Model Adaptation, Acoustic State Prediction

1. INTRODUCTION

Deep learning has been instrumental in bringing speech products to mass markets. We are witnessing the creation of many on-device as well as on-cloud speech applications that deliver strong ASR performance. Deep learning also enabled digital personal assistants in Cortana, Alexa, Google Home and Siri, that have become an important resource for everyday use. Today we expect the speech products to work well in not just controlled environments but also in acoustic scenarios including noise, far-field conditions, non-native speech, kids, whisper, natural conversation, and side-speech etc. These expectations and the availability of massive training data and computing power, offer many new opportunities and challenges in the speech research.

The goal of any speech application is to produce the highest possible accuracy given reasonable constraints in computing power and latency. Over the past years, speech researchers have developed a variety of algorithms and architectures to learn speech models, as well as, speech features that are robust to acoustic scenarios [1]. Recently the deep long-short term memory (LSTM) models in [2, 3] demonstrated further improvements over an earlier application of deep learning in DNN models [4, 5, 6, 7, 8]. LSTM models explicitly control the memory of the network in terms of input and forget gate modules, this provides a control over the information flow in the network and alleviates the gradient vanishing problem associated with deep networks [9]. The newer advances in deep learning also include end-to-end systems in [10, 11, 12, 13].

Besides speech features and model structures, SR systems also leverage techniques in model or speaker adaptation [14, 15, 16, 17] that personalize models for a specific scenario or speaker. Adaptation techniques provide significant value on top of speaker-independent (SI) models. This study focuses on single microphone but given multiple microphones, we can also apply a variety of beamforming and stream combination techniques in [18, 19].

The scope of our work lies in the broad area of acoustic model combination (AMOC). A number of AMOC techniques have been developed for SR applications that show significant improvement in accuracy [20]. The importance of AMOC is obvious from recent speech benchmarks in [21] where most competing systems use a variety of AMOC techniques. Besides speech, model combinations are also critical in most other deep learning areas, including image-net classification [22]. In our work, we propose static and dynamic combination of acoustic model states and demonstrate significant accuracy improvements. The state combination of models [23] by itself isn't new but our precise contribution is to learn a state-dependent set of weights in a data-driven framework that aligns with ASR training criterion. These weights can be static, *i.e.* fixed for the models, or dynamic, where the weights are obtained from a prediction model. Besides presenting word error rate (WER) metrics, we also analyze our approach and discuss interesting findings.

We present our work in the context of combining 2 acoustic models (AM): (1) a generic AM, that's trained to work very well for a broad range of acoustic scenarios, (2) a dedicated model that's specifically trained for a far-field speaker scenario. Our objective is to combine above 2 models such that the single combined model shows strong gains for all the scenarios over the previous best results in that scenario. And more specifically we expect to broaden the scope and accuracy of our acoustic models to diverse scenarios. This is specially important for server applications, where many speech developers can link to our cloud service, and expect robust ASR performance despite their acoustic application environment, audio processing pipeline, and speaker base etc. With a single combined model, we can deliver on the expectations without asking developers about their evolving acoustic scenarios and applications. Despite all the advances in computing resources, building a large scale ASR model requires significant investments in experimenting, training and testing. So with combined model we can focus our investments in building a single underlying model to serve most of our customers.

The rest of this work is organized in following: we briefly review a few model combination techniques relevant to our work in Sec. 2. We propose static state prediction in Sec. 3, and follow up with dynamic prediction in Sec. 4. We present our experiments and results in Sec. 5, and focus on analysis and extensions of our work in Sec. 5.1. We conclude our work in Sec. 6.

2. ACOUSTIC MODEL COMBINATION

Given our focus on acoustic model combinations, we briefly discuss a few relevant model combination techniques. These techniques may have multiple objectives. A common goal is to build diverse models that do great in particular scenarios; we expect to train models with complementary error patterns, that we can leverage and combine. We can build models for a broad group of speakers, *ex* gender-dependent models, and combine them. We can apply similar rational and build models for native and non-native speakers of English. We may also build diverse models from different speech features. We can also train models for a variety of acoustic model states, *senones*. Along with acoustic model, model combination can use language models too. Besides the diversity in the training data and algorithms, we can also develop model combination techniques that include: (1) feature-level combination, where we concatenate or join information from multiple speech features and train a single model, (2) state combination, there we combine acoustic states information, (3) confusion network combination (CNC) [24], hypothesis combination with ROVER [20] with confidence scores [25], and frame or classifier-based system combinations in [26, 27].

Typically state and hypothesis combination demonstrate larger gains [21]. We scope our work in the context of combining acoustic states that builds a single underlying acoustic model requiring a single ASR engine instance. In contrast, combining ASR hypotheses requires respective engine instances, along with developing tools and infrastructure to execute those instances to combine the hypotheses within reasonable computing and latency constraints. So compared to ROVER-like techniques, our work significantly saves the computing and speech deployment resources.

3. STATIC STATE PREDICTION (SSP)

State combination of the acoustic models is an effective technique in the broad scope of model combination. We apply our work to LSTM-RNN models that consist of a few layers of LSTM cells [9] along with a Softmax layer at top. The context-dependent tied triphones constitute the acoustic states, and the model predicts a distribution over the states for a given frame of speech features. Combining the models at the state level, *i.e* equivalently combining the predicted distributions for respective models has been applied before. However, most previous work [23] used a fixed state-independent weight for model combination, where the weight is tuned on the task of interest. The specific contribution of our work is to better analyze the combination weights and ingest new capabilities in the state combination framework. In particular, we aspire to: (a) build a data-driven framework for learning the state combination weights, (b) use ASR criterion to learn the model weights to be best aligned with ASR performance, (c) incorporate state-dependent capabilities in the combination weights, (d) static as well as dynamic prediction framework for the weights. We also take opportunities for deeper analysis of our findings and accuracy results.

For our work, we also consider an effective baseline that combines the model states with a weight of 0.5, *i.e.* equal weights for the 2 models. We show interesting properties and trade-offs of the baseline combination in Sec. 5, where we see gains on some tasks but sub-optimal performance on other tasks. We take motivations from the baseline, rationalize improvements and regressions, and build our work to achieve near optimal performance for all scenarios by embedding new intelligence in the state combination framework. The baseline combination points to significant scope with better predicting the combination weights. We work on this motivation, and propose to train the combination weights in the framework of the ASR itself. We motivate a data-driven framework for learning the combination weights and present static state prediction (SSP) and dynamic state prediction (DSP) approaches. We focus on SSP in this section.

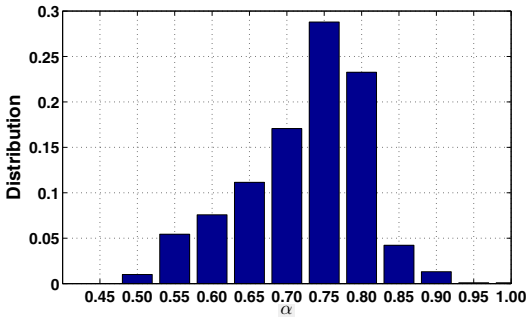

Fig. 1. *Distribution of state-dependent combination weights α from static state prediction.*

The SSP approach is essentially identical to the DSP approach demonstrated in Fig. 2, except that SSP excludes the prediction cell for combination weights and reduces to time-independent combination weights $\alpha[k]$. For SSP, we represent the combined model states as:

$$S[k] = \alpha[k] \cdot S_1[k] + (1 - \alpha[k]) \cdot S_2[k] \quad (1)$$

where, the state combination weights $\alpha[k]$, where k indicates an acoustic state, is state dependent with dimension as #states in the acoustic model, and S_1 and S_2 are the state predictions from the 2 acoustic models. We initialize all $\alpha[k]$ to a fixed value, say i. We use standard ASR training criterion to train state-dependent combination weights. We also evaluated an alternative approach where we restrain $\alpha[k]$ to be identical for all states but found it to be sub-optimal than state-dependent $\alpha[k]$. The choice of the initialization parameter i likely depends on the application scenario, and the nature of the models. For our application we chose and verified $i = 0.5$ to work well. We applied combination to the Softmax S_1 and S_2, as well as the corresponding pre-Softmax values, and consistently found pre-Softmax combination to work better and use that in our work.

3.1. State-independent combination weight as a special case of state-dependent weights

We demonstrate that the state-independent combination weights, *i.e.* with identical $\alpha[k]$ for all states, is a special case of the state-dependent weights. We begin from the general case of state-dependent model combination in eq. 1. We consider a special case where for a particular speech frame, only one of the states, say k, is dominant for both the models, and rest of states, *i.e.* $S_1[j]$ and $S_2[j]$ are either 0 or significantly small for $j \neq k$. In that restricted scenario, eq. 1 is equivalent to:

$$S = \alpha[k] \odot S_1 + (1 - \alpha[k]) \odot S_2, S_1[j] \approx 0, S_2[j] \approx 0 \text{ for } j \neq k \quad (2)$$

Where $\odot$ indicates element-wise product. The predicted α is identically $\alpha[k]$ for all states k in above state-independent combination approach.

Next, in comparison to the baseline combination with identical α for all acoustic states, SSP offers additional advantages. SSP training aligns with the ASR training objectives to learn state-dependent

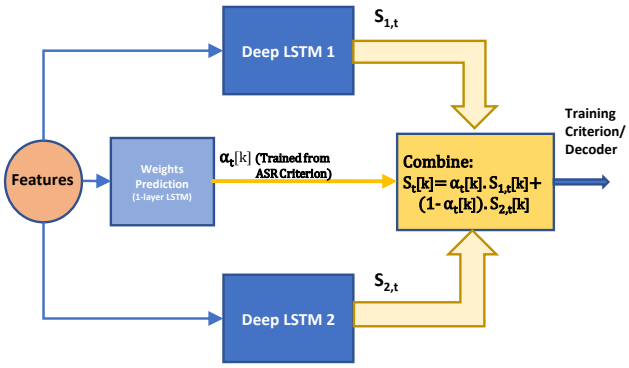

Fig. 2. *A general framework for dynamic acoustic model combination. A small prediction model (combination cell) dynamically evaluates the model combination weight $\alpha_t[k]$ for time instant t and acoustic state k. We can also train a time-independent $\alpha[k]$ for static state prediction; it doesn't require a prediction model.*

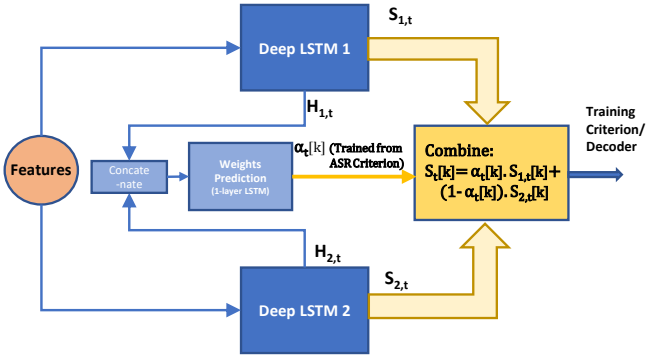

Fig. 3. *An extension of the dynamic acoustic prediction where the input to the prediction model is obtained from the concatenation of the model hidden states.*

combination weights. This allows SSP to best leverage the state classification boundaries from individual models. We plot a histogram of the trained weights α in Fig. 1 and demonstrate that the training criterion indeed converges to a state-dependent α. We also observed that that the predicted α for states like "sil", "noise" strongly favored the generic model. In our work, we trained the generic model from a large corpus including mobile and close-talking data, so the generic model better learns the classification for silence and noise. In comparison, the dedicated model training predominantly consists of far-field and noisy data, where the classification boundaries for silence and noise are fuzzy. Overall, SSP learns a way to best leverage the classification boundaries from the individual models.

4. DYNAMIC STATE PREDICTION (DSP)

In this section we extend our work on SSP in Sec. 3 to dynamic state prediction (DSP). In SSP, we leveraged SR training criterion and trained state-dependent combination weights. Although, SSP provided strong gains in some scenario it's still sub-optimal and we seek opportunities to better leverage the task at hand. We realize that audio from different acoustic conditions exhibit different characteristics, therefore, static combination weights are likely sub-optimal. We leverage scenario-dependent combination by dynamically predicting time and state-dependent combination weights $\alpha_t[k]$ in:

$$S_t[k] = \alpha_t[k] \cdot S_{1,t}[k] + (1 - \alpha_t[k]) \cdot S_{2,t}[k] \quad (3)$$

Where, we use a prediction model to predict $\alpha_t[k]$ at time instant t and acoustic state k. We demonstrate this approach in Fig. 2. We have flexibility to use a variety of prediction models in the DSP framework. Our acoustic model consists of LSTM cells, so we naturally chose 1-layer LSTM cell to model and predict $\alpha_t[k]$. Our prediction model aligns well with the core ASR models; we reuse SR features and ASR training criterion to predict $\alpha_t[k]$.

We propose an extension of the DSP approach in Fig. 3. There we concatenate the hidden layer outputs from the SR models, and take that as an input for the prediction model. We base this work on our understanding that in a deep network, the initial layers normalize the features and make it robust across speakers and acoustic environments. Whereas, the upper layers gradually learn decision boundaries. We expect to test and improve the prediction performance with inputs from the hidden layer activation. Furthermore, the prediction model based on speech features doesn't include any information from the core ASR models; the proposed extension allows us to incorporate some information from the individual ASR models. We can also extend above to include features as well as hidden layer information for the prediction model.

5. EXPERIMENTS AND RESULTS

We conducted our experiments on a large vocabulary speech recognition task. We build a standard 6-layers unidirectional LSTM model with cross-entropy (CE) [4] criterion from a large data corpus from across Microsoft speech services in Xbox, Cortana, Conversation, and Speaker with a total of approximately 20k hours of production data. Subsequently we augment the data with noise and room impulse response for a total of around 30k speech hours. We use 80-dim log-Mel features for model training, the corresponding time window is 25-msec with 10-msec window shift. Our LSTM cells use 1024 memory units. We train our models to support near loss-less decoding with frame-skips for LSTM model evaluation. Following above, we build 2 acoustic models. We refer to the first acoustic model (AM) as "Generic" model, it's trained to handle a wide variety of scenarios across multiple speech endpoints. We also build a "dedicated" model for far-field Cortana Speaker scenario. Clearly we expect the "dedicated" model too to work reasonably well for most scenario with the strongest focus on improving Speaker scenario. For DSP prediction, we used a 1-layer LSTM with 512 memory units, and project the output to the 9k acoustic states in our model. We also tried a few advanced prediction models but the 1-layer LSTM retained almost all of the gains and was yet small and fast to train. We test our work on tasks across Cortana, Speaker, Car and Conversation, with over 100k utterances in total. We use a 5-gram language model with vocabulary of over 1M words.

Our objective is to leverage above models and combine them for: (a) the best overall performance from the combined model, (b) minimize any regression such that individual tasks operate at close to the best achievable accuracy from either of the 2 models. We present the 2 baseline results in Table. 1. We note strong performance for the generic model on a wide-variety of tasks, whereas, the dedicated model shows significant improvement for Speaker but it's weaker for other scenarios. There, the SSP approach improves

Table 1. *Static state prediction for model combination.*

Models	Cortana [%]	Speaker [%]	Car [%]	Conversation [%]	Avg. [%]
Generic AM	11.8	8.2	14.2	21.4	13.89
Dedicated AM	15.4	6.0	16.3	35.7	18.35
Comb. w/ 0.5	11.6	6.2	14.0	23.3	13.78
SSP	11.5	7.0	13.9	21.7	13.53

Table 2. *DSP for model combination. We present DSP flavors with speech features in Fig. 2 as well as concatenated hidden activations in Fig. 3 as input for the prediction model.*

Models	Cortana [%]	Speaker [%]	Car [%]	Conversation [%]	Avg. [%]
Generic AM	11.8	8.2	14.2	21.4	13.89
Oracle (Generic or Dedicated)	11.8	6.0	14.2	21.4	13.34
DSP-Features	11.7	6.0	13.9	21.4	13.26
DSP-H2	11.6	5.9	14.0	21.6	13.26
DSP-H4	11.7	6.0	14.1	21.6	13.35
DSP-H6	11.6	6.0	14.0	21.7	13.34
DSP-Features + Finetune	11.6	6.1	13.0	21.4	13.05
WERR over Generic AM	1.8	25.2	8.0	-0.2	6.0

the average WER from 13.89% for generic model, and 13.78% for the baseline combination with a fixed weight of 0.5, to 13.53%. We note that compared to generic model, the baseline combination improves some tasks but has significant regressions for Conversation task. Similarly, compared to dedicated AM, SSP regresses for speaker task.

Above indicates that SSP by itself is insufficient at generalizing to diverse speech applications, and that the ideal combination weights should also be a function of the task; this leads to the DSP framework for state combination. We present DSP results in Table 2. We also present a specific oracle result, where the WERs are the best possible for a task from the 2 AMs. In Table 2, "DSP-Features" indicates the DSP combination with speech features as input for the prediction model, and "DSP-H2" indicates the flavor in Fig. 3 that uses the hidden activations from the 2nd hidden layer (from bottom). On average the "DSP-H2" is better than using activations from upper layes in "DSP-H4" or "DSP-H6", and is similar to that for "DSP-Features". These methods improve the previous best 13.53% SSP WER in Table 1 to 13.26%. It's also satisfying to note that the WERs from the dynamic methods are better than the best WERs from the 2 individual models, as noted in the Oracle result. Furthermore, we take the opportunity for additional improvements in the DSP framework by finetuning the individual ASR models for a few training epochs. This leads to the best DSP result, and pushes the WER from 13.89% for the generic model to 13.05%, achieving an overall 6% WERR on average, and in particular 25% WERR for Speaker task, and 8% WERR for Car.

We also evaluated ROVER for model hypotheses combination but it regressed over individual best models; that's likely due to wider WER gap between the generic and dedicated model on most tasks.

Table 3. *Expected cost for model combination techniques.*

Models	Computing Cost
Generic AM	1x
Comb. with individual recognition results (ex - ROVER [20])	2x
Comb. with Static State Prediction (SSP)	1.2x
Comb. with Dynamic State Prediction (DSP)	1.24x

We note approximate computing costs in Table 3. Noting the cost for generic AM as $1x$, a ROVER-like approach that requires hypotheses from 2 models requires atleast $2x$ cost. The SSP and DSP approaches require computing additional LSTM models but retain the decoding cost in Generic AM. We find the computing cost for SSP and DSP methods to be $1.2x$ and $1.24x$, respectively. Compared to ROVER-like techniques, our proposed approaches show significant savings in computing cost.

5.1. Model Combination Applications, Analysis and Extensions

We presented our work in the context of developing and combining a dedicated Cortana speaker AM with a generic AM. We can also apply our proposed approach in other scenarios: (a) improving the performance for kids from a kids-specific AM, (b) similar to (a), improving performance for non-native English speakers, (c) combining narrowband and wideband-specific AMs, (d) extending to sequence criterion as training criterion, (e) speaker or environment adaptation by developing and finetuning a dedicated AM, (f) improving the baseline model by training a model under the combination criterion to further minimize the loss metric.

We expect an adequate understanding of the application, nature of the core AMs, and amount of training data etc. to best apply our work. Our work purely lies in the scope of AM training, so it's expected to be useful in the context of combination techniques like ROVER or re-ranking that can additionally leverage language model information and ASR hypotheses. We are also extending our approach to combine 3 or more models. The current DSP work uses a small prediction model; we are testing with prediction models that are specific to the AMs, as well as expanding the prediction model to simultaneously predict all combination weights. We are also applying our work to UK and Canadian English, and are seeing significant gains.

6. CONCLUSION

In this work we presented static and dynamic state prediction for acoustic model combination. We motivated a rational for state-dependent combination weights that's trained with an objective aligned to ASR training. We expanded our work to dynamic prediction by training a small prediction model to best learn and predict the combination weights. We applied our techniques to large scale enUS tasks and demonstrated an average 6% WERR over generic model by combining it with dedicated Speaker model, and in pariticular 25% WERR for Speaker and 8% WERR for Car tasks. Our work led to build a single underlying model to serve most of our customers. We also discussed many extensions and applications of our work.

7. REFERENCES

[1] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[2] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014, pp. 338–342.

[3] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Proc. Interspeech*, 2015.

[4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[6] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael L. Seltzer, Geoffrey Zweig, Xiaodong He, Jason D. Williams, Yifan Gong, and Alex Acero, "Recent advances in deep learning for speech research at microsoft," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8604–8608, 2013.

[7] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.

[8] G.E. Dahl, D. Yu, L. 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.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] Hasim Sak, Matt Shannon, Kanishka Rao, and Françoise Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *Proc. of Interspeech*, 2017.

[11] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Katya Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018.

[12] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NIPS*, 2015.

[13] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brake, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," *CoRR*, vol. abs/1508.04395, 2015.

[14] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. ICASSP*, 2013, pp. 7893–7897.

[15] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Proc. ICASSP*, 2014, pp. 6359 – 6363.

[16] K. Kumar, C. Liu, K. Yao, and Y. Gong, "Intermediate-layer dnn adaptation for offine and session-based iterative speaker adaptation," in *Interspeech*, 2015.

[17] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," 2014.

[18] Xiaofei Wang, Ruizhi Li, and Hynek Hermansky, "Stream attention for distributed multi-microphone speech recognition," *Proc. Interspeech 2018*, pp. 3033–3037, 2018.

[19] Cha Zhang, Dinei Florêncio, Demba E Ba, and Zhengyou Zhang, "Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 538–548, 2008.

[20] Jonathan G Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 347–354.

[21] Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal, "The fifth'chime'speech separation and recognition challenge: Dataset, task and baselines," *arXiv preprint arXiv:1803.10609*, 2018.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[23] W. Xiong, L. Wu, F. Alleva, Jasha Droppo, X. Huang, and Andreas Stolcke, "The microsoft 2017 conversational speech recognition system," *CoRR*, vol. abs/1708.06073, 2017.

[24] Gunnar Evermann and PC Woodland, "Posterior probability decoding, confidence estimation and system combination," in *Proc. Speech Transcription Workshop*. Baltimore, 2000, vol. 27, pp. 78–81.

[25] K. Kumar, T. Anastasakos, and Y. Gong, "Word characters and phone pronunciation embedding for ASR confidence classifier," in *Proc. ICASSP*, 2019.

[26] Björn Hoffmeister, Tobias Klein, Ralf Schlüter, and Hermann Ney, "Frame based system combination and a comparison with weighted rover and cnc," in *Ninth International Conference on Spoken Language Processing*, 2006.

[27] Björn Hoffmeister, Ralf Schlüter, and Hermann Ney, "iCNC and iROVER: The limits of improving system combination with classification?," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.