Meeting Transcription Using Asynchronous Distant Microphones

Takuya Yoshioka, Dimitrios Dimitriadis, Andreas Stolcke, William Hinthorn, Zhuo Chen, Michael Zeng, Xuedong Huang

Speech and Dialog Research Group, Microsoft, U.S.A.

{tayoshio,didimit,anstolck,wihintho,zhuc,nzeng,xdh}@microsoft.com

Abstract

We describe a system that generates speaker-annotated transcripts of meetings by using multiple asynchronous distant microphones. The system is composed of continuous audio stream alignment, blind beamforming, speech recognition, speaker diarization, and system combination. While the idea of improving the meeting transcription accuracy by leveraging multiple recordings has been investigated in certain specific technology areas such as beamforming, our objective is to assess the feasibility of a complete system with a set of mobile devices and conduct a detailed analysis. With seven input audio streams, our system achieves a word error rate (WER) of 22.3% and a speaker-attributed WER (SAWER) of 26.7%, and comes within 3% of the close-talking microphone WER on non-overlapping speech. The relative gains in SAWER over a single-device system are 14.8%, 20.3%, and 22.4% for three, five, and seven microphones, respectively. The full system achieves a 13.6% diarization error rate, 10% of which are due to overlapped speech.

Index Terms: meeting transcription, asynchronous distributed microphones, distant speech recognition, speaker diarization, system combination, blind beamforming

1. Introduction

Speaker-attributed automatic speech recognition (ASR) of natural meetings has been one of the very challenging tasks since the early 2000s, when the NIST Rich Transcription Evaluation series [1] started. Systems developed in the early days yielded high error rates, especially when distant microphones were used as input. However, with the rapid progress in conversational speech transcription [2, 3], far-field speech recognition [4–7], and speaker identification and diarization [8, 9], realizing accurate meeting transcription from a distance seems to be within reach, especially with microphone arrays. In addition to microphone array setups, single-microphone systems have also been evaluated.

Using multiple asynchronous audio capturing devices, such as mobile phones and laptops, adds another dimension to the task. On the one hand, the use of the spatially distributed microphones allows us to measure acoustic events at different points in a room. Therefore, meeting attendees may want to lend their own recording devices to the transcription system, i.e., they can put their devices on the table and connect with their devices to a server to improve transcription quality. On the other hand, while there are several pioneering studies [10], it is unclear what the best strategies are for consolidating multiple asynchronous audio streams and to what extent they work for natural meetings in online and offline setups.

In this paper, we investigate a meeting transcription architecture based on asynchronous distant microphones by combining both front-end and back-end techniques. The resulting system is analyzed through experiments on real-world meeting recordings. Our proposed system is designed to generate word recognition results in real time and then provide improved speaker-attributed transcriptions with limited latency.

In addition to the end-to-end system analysis, we make the following specific contributions: we examine the idea of “leave-one-out beamforming” in the asynchronous multi-microphone setup. This method was proposed to benefit from both beamforming and system combination approaches but tested only with synchronized signals [11]. The computational cost required for calculating multiple beamformers can be reduced by taking advantage of the properties of spatial covariance matrices. We investigate a similar diversity-preserving strategy for acoustic model fusion. Further, we describe system combination schemes that take account of both word recognition and speaker attribution. Finally, we show results based on incremental ROVER that processes the ASR and diarization outputs with low latency.

2. Task and System Overview

We record a meeting with M audio capturing devices, such as cell phones, tablets, and laptops. The devices can be randomly placed at any locations in a conference room. The acoustic signal picked up by each device is transmitted to a common server. The server then generates a speaker-attributed transcription of the meeting conversation in real time as it receives the signals from the devices. In this paper, we assume that all meeting attendees have enrolled in the system and have provided their voiceprints for speaker identification.

Figure 1 shows the processing flow of the proposed system. The input signals received by the server are misaligned for various reasons such as clock drift on each recording device, differences in on-device signal processing, packet generation, and signal transmission channels. As in Fig. 1, the audio stream alignment module constantly corrects the inter-channel signal misalignments. This is followed by a beamforming module, which receives the M time-aligned audio signals and yields N enhanced signals. In this paper, we deal with the case where $M = N$ while this is not a requirement. Each enhanced signal is fed to a speech recognition module to produce a real-time transcription as well as n-best recognition hypotheses with word-level time marks. The diarization module then generates a speaker label sequence for each of the segments detected by the ASR decoder by utilizing both word-level time marks and speaker embeddings extracted from the enhanced audio. Eventually, the speaker labels and word hypotheses are gathered by the system combination module to yield a final speaker-attributed word transcription.

We have settled on this architecture based on several considerations: it supports both beamforming and later-stage system combination approaches, which are found to be beneficial together [11]. Also, we perform diarization after speech recog-

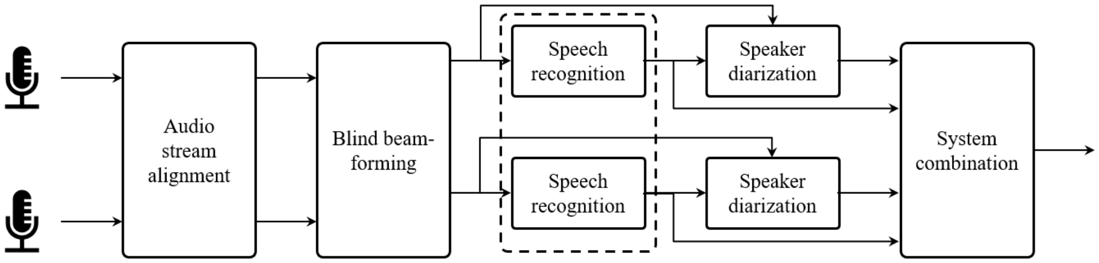

Figure 1: *Processing flow diagram of proposed meeting transcription system with asynchronous multi-microphone input. ASR may be performed jointly across channels as implied by the dotted box.*

nition, unlike in many previous systems. Since diarization typically has a longer algorithmic delay this allows preliminary recognition results to be displayed in real time. Finally, system combination, coming last, is designed to merge and benefit both word recognition and speaker attribution.

3. System Components

3.1. Audio Stream Alignment

The audio stream alignment module picks one of the input streams as a reference and aligns each of the other signals to the reference signal. To align a signal to the reference one, we first detect the time lag between the two signals and then adjust the non-reference signal. For this purpose, we have two variable-length first-in-first-out buffers for each stream: one for time lag detection, one for output generation. After a few seconds (2 s in our experiments), we extract as many samples from the output buffer as those pushed to the buffer. These samples are given to the downstream modules.

At T-second intervals ($T = 30$ in our experiments), we calculate the cross-correlation coefficients between the two signals stored in the time-lag detection buffer and pick the sample lag L that maximizes the cross-correlation value. We decimate the samples in the non-reference stream's output buffer by $|L|$ if $L > 0$. Otherwise, we increase the number of samples by $|L|$. This can be done with resampling. The time-lag detection buffer can then be refreshed.

At the beginning of the alignment processing, we may calculate the cross-correlation more frequently (e.g., every 1 s) until we find a significant peak in the cross-correlation sequence. This 'global' time lag can also be used to adjust the output wait time. In an online client-server setting, the global time lag is small. When we apply the system to offline independent recordings, the global time lag can be in the order of minutes. In this case, we may use a sliding window to first obtain an approximate estimate of the global time lag and then fine-tune the estimate by using the sample-level cross-correlation as described above.

3.2. Blind beamforming

For beamforming, we adopt a mask-based blind processing approach [12, 13]. This approach was shown to perform as well as carefully designed beamformers that utilize array geometry information [14].

Mask-based blind beamforming. Assuming M microphones to be available, an enhanced short time Fourier transform (STFT) coefficient can be computed as an inner product of the M-dimensional beamformer coefficient vector $\boldsymbol{w}_f$ and an input multi-channel STFT coefficient vector, where subscript f denote a frequency bin index. In one formulation, the beamformer coefficient vector is estimated with a minimum variance distortionless response (MVDR) principle as

$$\boldsymbol{w}_f = \boldsymbol{\Phi}_{\mathrm{N},f}^{-1}\boldsymbol{\Phi}_{\mathrm{S},f}\boldsymbol{r} \big/ \mathrm{tr}\big(\boldsymbol{\Phi}_{\mathrm{N},f}^{-1}\boldsymbol{\Phi}_{\mathrm{S},f}\big), \tag{1}$$

where $\boldsymbol{\Phi}_i$ denotes a spatial covariance matrix ($i = \mathrm{S}$ for speech; $i = \mathrm{N}$ for noise) while $\boldsymbol{r}$ is a one-hot unit vector which has 1 at a reference microphone position, which may be chosen based on a maximum signal-to-noise ratio (SNR) principle [15]. The speech and noise spatial covariance matrices are estimated using spectral masks.

In our experiments, a neural network trained to minimize the mean squared error between clean and enhanced log-Mel features was used [14]. The spectral masks were estimated for every 1 s-batch. The beamformer coefficients are also updated accordingly.

Strategies for generating multiple different outputs. System combination relies on errors being partly uncorrelated among inputs. For this reason, [11] suggested manipulating early-fusion approaches to keep the outputs as decorrelated as possible, specifically using a leave-one-out approach to beamforming. Two such schemes are investigated in this work.

In the first scheme, called the all-channel approach, we rotate the 1's position in unit vector $\boldsymbol{r}$ from the first element to the last to create different beamformer coefficient vectors based on Eqn. (1). A potential drawback of this approach is that the beamformer outputs might not retain enough diversity among different channels because they are still based on the same input signals.

The second, "leave-one-out", scheme forms an acoustic beam by using $M-1$ channels while varying the left-out microphone in a round-robin manner. This scheme requires M different $(M-1)$-dimensional noise spatial covariance matrices to be inverted in order to calculate M beamformers based on Eqn. (1). It can be shown that all the M inverse spatial covariance matrices of size $M-1$ can be derived from a shared M-dimensional inverse spatial covariance matrix by utilizing the matrix inversion properties of block and permutation matrices. Therefore, both two schemes can be run with similar computational cost.

3.3. Speech recognition

The speech recognition module converts an incoming audio signal to an n-best list with word-level time marks. In the experiments reported later, we used a conventional hybrid ASR system, consisting of a latency-controlled bidirectional long short term memory (LSTM) acoustic model (AM) [16] and a weighted finite state transducer decoder. Our AM was trained on 33K hours of in-house audio data, including close-talking, distant-microphone, and artificially noise-corrupted speech. Decoding was performed with a trigram language model (LM). Whenever a silence segment longer than 300 ms was detected, the decoder generated an n-best list, which was rescored with both a 5-gram trained on 100B words and an LSTM-LM. The latter used two 2048-unit recurrent layers and was trained on 2B words.

3.4. Speaker diarization

Given a speech region detected by the speech recognition module, speaker diarization assigns a person label to each word in the top recognition hypothesis. We adopt an approach consisting of three steps: d-vector generation, segmentation, and speaker identification. With our decoder configuration, each incoming speech region typically contains up to 20 words.

The d-vector generation step calculates speaker embeddings [17] for every fixed time interval (320 ms in our system). We trained a ResNet-style embedding extraction network [18] on the VoxCeleb corpus [19] to generate 128-dimensional d-vectors.

The speaker segmentation step decomposes the received word sequence into speaker-homogeneous subsegments. This is performed with an agglomerative clustering approach [20,21] by using the d-vectors as observed samples. Initially, every single word comprises a unique subsegment. For every neighboring subsegment pair, the degree of proximity between the two subsegments is estimated in the embedding space. The closest pair is then merged to form a new subsegment. The proximity is defined as the cosine similarity between the mean d-vectors. This process is repeated until the cosine similarity drops below a threshold (0.15 in our experiments).

Finally, a speaker label is assigned to each subsegment. In this paper, we assume that a list of meeting attendees is available. For each subsegment, a segment-level embedding is computed by averaging the d-vectors over the subsegment. Likewise, the embedding of each speaker is pre-computed from enrollment audio samples, which were around 30 s long. The speaker label that gives the highest cosine similarity to the subsegment embedding is selected.

3.5. System combination

System combination consolidates the multiple speaker-attributed ASR results to produce a final transcription result. ROVER [22] and confusion network combination (CNC) [23, 24] are two popular system combination approaches. The goal of this step is to combine evidence from all channels, after beamforming, for both word and speaker recognition. As discussed in Section 3.4, a speaker label is assigned to every word based on the acoustics of the available audio streams. For purposes of ROVER, the speaker identities are encoded as audio channel numbers. Then, they are submitted to the NIST ROVER algorithm [22] along with the word hypotheses, which combines them by aligning words based on dynamic programming and their time marks and extracting the words with the highest vote count. We have modified the interface to the ROVER algorithm in such a way that this process can be invoked online, as new speaker-attributed word hypotheses become available from the diarization module, by using a sliding window shared across streams. Due to misalignment between different decoder outputs, some words may appear twice. We run a simple filter removing the duplicates.

For CNC-based system combination, we devised an alternative algorithm that currently operates in batch mode. On each channel, for each speech segment, the decoder generates n-best lists, which are aligned into confusion networks (CNs). The speaker recognition output from each channel is also encoded as a CN, using special tags for the speaker identities, interspersed with 1-best word hypotheses. We modified the CN algorithms in SRILM [25] to support aligning word and speaker CNs, and augmented the usual minimal edit distance objective function with a time-misalignment penalty. The end result of the modified CNC is that n-best word hypotheses from all channels are merged with the speaker information, and the speakers and words with highest combined posteriors can be decoded jointly.

Table 1: *AM combination. Results for one particular meeting. Numbers should not be compared with those of other tables.*

	ROVER		CNC	
	%WER	%SAWER	%WER	%SAWER
Baseline	25.9	28.2	25.9	28.2
Sum	25.4	28.4	24.7	28.2
Max	22.5	27.5	22.5	27.2
Max 6 of 7	23.8	27.2	22.1	26.7
Max 3 of 7	24.2	26.8	22.3	26.9

3.6. Acoustic model combination

In addition to the channel-fusion approaches described above, i.e., beamforming and system combination, it is also possible to combine frame-level senone posterior probabilities from multiple streams before ASR decoding [26]. While this approach is not integrated into the end-to-end system yet, we have investigated the effectiveness of senone-level AM fusion, with strategies aimed at increasing the diversity of the output results for later processing with ROVER or CNC.

The baseline results (first row) in Table 1 use senone posteriors from a single channel, produced by the AM and used as input to the decoder. Next, the sum and max of senone posteriors across channels are investigated. This results in a single word hypothesis stream, with ROVER/CNC combining speaker hypotheses only. Similar to the leave-one-out strategy for beamforming, we can preserve diversity by sampling from the channels, followed by hypothesis combination. In the last two rows of Table 1, we present results with 6-out-7 senone fusion (resulting in 7 different senone subsets), and 3-out-7 with 35 outputs. In the latter case, we sample 7 of the 35 possible outputs to reduce computation. Either way, the 7 resulting decoding outputs are routed to system combination as before.

4. Experiments and Results

4.1. Data and metrics

We conducted a series of experiments to analyze the performance of the system described so far. We recorded five internal meetings; three meetings were recorded with seven independent consumer devices, four of which were iOS devices and three based on Android. All devices were different products. The other two meetings were recorded with a seven-channel circular microphone array. For these meetings, we did not make use of the fact that the signals were synchronous and let the signals through the entire pipeline including the audio stream alignment module. Those meetings took place in several different rooms and lasted for 30 minutes to one hour each, with three to eleven participants per meeting. The meetings were neither scripted nor staged; the participants conducted normal work discussions and were familiar with each other. Partly as a result,

Table 2: *WERs and SAWERs using seven microphones.*

Sys. Comb.	Beamforming	%WER	%SAWER
None	None	27.0	34.4
(real time)	All channels	24.8	30.8
	Leave one out	24.9	30.9
ROVER	None	25.3	28.5
(online)	All channels	24.2	27.4
	Leave one out	24.2	27.2
CNC	None	22.8	27.7
(offline)	All channels	22.5	26.9
	Leave one out	22.3	26.7
IHM + reference diarization		14.4	14.4

Table 3: *Impact of number of microphones for system based on leave-one-out beamforming and CNC.*

No. of microphones	1	3	5	7
%WER	27.0	24.0	22.7	22.3
%SAWER	34.4	29.3	27.4	26.7

about 10% of all speech occurred in overlap with at least one other speaker. Reference transcriptions were created by professional transcribers based on both close-talking and far-field recordings.

The system outputs were scored with NIST's scoring toolkit [27] to calculate both standard, speaker-agnostic word error rates (WERs) and speaker-attributed WERs (SAWERs). For the latter, a word is counted as correct only if both the word label and its speaker are identified correctly. Note that these metrics count overlapped speech as any other. Since our system, at present, does not attempt to separate overlapping speech we thus have a floor on the error rate of about 10%.

4.2. Speech transcription accuracy

Table 2 shows the results for various configurations. For the systems that do not perform any form of system combination, seven different results were obtained, each corresponding to a different one of the microphones, and the averages are reported in the table. As a best case condition, and to calibrate the difficulty of the distant-microphone task, the final table row gives results for individual head-mounted microphones (IHM), with reference speaker segmentation.

The best system, combining beamforming and CNC, achieved substantial improvement over the single microphone system. The WER and SAWER relative gains were 17.4% and 22.4%, respectively. Relative to the IHM scenario as a floor, WER and SAWER were reduced by 37% and 39%, respectively.

We can see that both beamforming and system combination (either with ROVER or CNC) contributed to the final performance, even though both steps combined information across channels. CNC provided the largest performance gain. While beamforming yielded a smaller gain, it is more easily used for real-time applications. The leave-one-out scheme provided slightly larger gains than the all-channel beamforming when combined with system combination, especially CNC, confirming our rationale in Section 3.2.

Table 3 shows the WERs and SAWERs for different numbers of microphones. There is a clear correlation between the number of microphones and the amount of improvement over the single channel system. Even with only three microphones, our system yielded relative gains of 11.1% and 14.8% in WER and SAWER, respectively.

To assess the speech recognition accuracy when a single person is speaking, we scored the results only against segments that did not contain any forms of overlap.[1] Note that this discarded 58% of the words. The results are shown in Table 4. By comparing the numbers with the results of Table 2, we can see that the system produced around 25% more accurate transcriptions for the non-overlapped segments. For the full system, the WER on non-overlapped speech is only 3.0% worse than with close-talking microphones. Considering that the overlaps make up about 10% of the speech duration, this result shows that segments including overlaps are more affected by the speaker-microphone distance.

Table 4: *Speaker-independent WERs for non-overlapped segments. SDM: single distant microphone. BF: beamforming*

System	SDM	BF	BF + CNC	IHM
%WER	20.6	18.1	16.2	13.2

Table 5: *Diarization error on speaker-attributed ASR output. The percentage of overlapped speech is 10.0%, and accounts for most of the missed speech.*

	Misses	FAlarms	SpkrErr	DER
Avg. by channel	10.5	3.3	1.8	15.6
CNC output	10.2	2.4	1.0	13.6

4.3. Speaker diarization accuracy

We took the speaker-attributed recognition output, added 0.5 s of extra duration at the margins of contiguous output from the same speaker, and evaluated the result according to the NIST "Who spoke when" task [21]. Note that our task is not speaker-agnostic diarization, but recognizing the known speakers. Also, we are not trying to recognize overlapping speakers, so about 10% of speech is missed, thus putting a floor on the missed speech and overall diarization error rate (DER).

Table 5 gives the speaker diarization error of the system, by channel and for the combined output. The false alarm rate is quite low since the recognizer acts as a very conservative speech detection engine. Similar to word recognition, CNC reduces the speaker error (44% relative) by pooling speaker label posterior probabilities across all channels.

5. Conclusion

We studied a meeting transcription architecture for asynchronous distant microphones, combining front-end and back-end techniques, and evaluated it on real meeting recordings. We found that both front-end (blind beamforming) and back-end (model or system combination) algorithms improve word error, speaker-attributed word error, and diarization error metrics. Both beamforming and senone posterior fusion can be made more effective in conjunction with system combination by using leave-one-out techniques. System combination was generalized such that it benefits both word and speaker hypotheses. On non-overlapped speech, the error rate is only 3% absolute worse than with close-talking microphones. We found similar results on NIST meeting evaluation data (14.6% WER on non-overlapping speech, 2.3% worse than IHM), reported in an expanded version of this paper [28]. In summary, our study shows the effectiveness of multiple asynchronous microphones for meeting transcription in real-world scenarios. A major remaining challenge is recognition of overlapped speech [29].

[1] This was done by using NIST's asclite with the "-overlap-limit 1" option.

6. References

[1] J. G. Fiscus, J. Ajot, and J. S. Garofolo, "The Rich Transcription 2007 meeting recognition evaluation," in *Multimodal Technologies for Perception of Humans. International Evaluation Workshops CLEAR 2007 and RT 2007*, ser. Lecture Notes in Computer Science, R. Stiefelhagen, R. Bowers, and J. Fiscus, Eds. Berlin: Springer, 2008, vol. 4625, pp. 373–389.

[2] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *Proc. Interspeech*, Aug. 2017, pp. 132–136. [Online]. Available: https://arxiv.org/abs/1703.02136

[3] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 conversational speech recognition system," in *Proc. IEEE ICASSP*, Apr. 2018, pp. 5934–5938. [Online]. Available: https://arxiv.org/abs/1708.06073

[4] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani, "The NTT CHiME-3 system: advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. IEEE ASRU*, 2015, pp. 436–443.

[5] J. Du, Y. Tu, L. Sun, F. Ma, H. Wang, J. Pan, C. Liu, J. Chen, and C. Lee, "The USTC-iFlytek system for CHiME-4 challenge," in *Proc. CHiME-4 Worksh.*, 2016, pp. 36–38.

[6] B. Li, T. N. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Punduk, K. Chin, K. C. Sim, R. J. Weiss, K. W. Wilson, E. Variani, C. Kim, O. Siohan, M. Weintraub, E. McDermott, R. Rose, and M. Shannon, "Acoustic modeling for Google Home," in *Proc. Interspeech*, 2017.

[7] J. Li, R. Zhao, Z. Chen, C. Liu, X. Xiao, G. Ye, and Y. Gong, "Developing far-field speaker system via teacher-student learning," in *Proc. ICASSP*, 2018, pp. 5699–5703.

[8] A. Zhang, Q. Wan, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *Proc. IEEE ICASSP*, May 2019, pp. 6301–6305. [Online]. Available: https://ai.google/research/pubs/pub47549

[9] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is hard: some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *Proc. Interspeech*, 2018, pp. 2808–2812.

[10] S. Araki, N. Ono, K. Kinoshita, and M. Delcroix, "Meeting recognition with asynchronous distributed microphone array using block-wise refinement of mask-based MVDR beamformer," in *Proc. IEEE ICASSP*, 2018, pp. 5694–5698.

[11] A. Stolcke, "Making the most from multiple microphones in meeting recordings," in *Proc. IEEE ICASSP*, 2011, pp. 4992–4995.

[12] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. IEEE ICASSP*, 2016, pp. 196–200.

[13] T. Higuchi, T. Yoshioka, N. Ito, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *Proc. IEEE ICASSP*, 2016, pp. 5210–5214.

[14] C. Boeddeker, H. Erdogan, T. Yoshioka, and R. Haeb-Umbach, "Exploring practical aspects of neural mask-based beamforming for far-field speech recognition," in *Proc. IEEE ICASSP*, 2018, pp. 6697–6701.

[15] H. Erdogan, J. R. Hershey, S. Watanabe, M. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Proc. Interspeech*, 2016, pp. 1981–1985.

[16] S. Xue and Z. Yan, "Improving latency-controlled BLSTM acoustic models for online speech recognition," in *Proc. IEEE ICASSP*, 2017, pp. 5340–5344.

[17] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE ICASSP*, 2014, pp. 4052–4056.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Computer Vision, Pattern Recognition*, June 2016, pp. 770–778.

[19] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, Aug. 2017, pp. 2616–2620. [Online]. Available: http://arxiv.org/abs/1706.08612

[20] J. Gauvain, L. Lamel, and G. Adda, "Partitioning and transcription of broadcast news data," in *Proc. ICSLP*, 1998, pp. 1335–1338.

[21] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 5, pp. 1557–1565, Sep. 2006.

[22] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. IEEE ASRU*, 1997, pp. 347–354.

[23] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauché, C. Richey, E. Shriberg, K. Sönmez, F. Weng, and J. Zheng, "The SRI March 2000 Hub-5 conversational speech transcription system," in *Proc. NIST Speech Transcription Workshop*, College Park, MD, May 2000.

[24] G. Evermann and P. Woodland, "Posterior probability decoding, confidence estimation, and system combination," in *Proc. NIST Speech Transcription Workshop*, College Park, MD, May 2000.

[25] A. Stolcke, "SRILM—an extensible language modeling toolkit," in *Proc. ICSLP*, J. H. L. Hansen and B. Pellom, Eds., vol. 2, Denver, Sep. 2002, pp. 901–904.

[26] S. Tibrewala and H. Hermansky, "Sub-band based recognition of noisy speech," in *Proc. IEEE ICASSP*, pp. 1255–1258.

[27] J. G. Fiscus, J. Ajot, N. Raddle, and C. Laprum, "Multiple dimension Levenshtein edit distance calculations for evaluating automatic speech recognition systems during simulaneous speech," in *Proc. Int. Conf. Language Resources, Evaluation*, 2006, pp. 803–808.

[28] T. Yoshioka, Z. Chen, D. Dimitriadis, W. Hinthorn, X. Huang, A. Stolcke, and M. Zeng, "Meeting transcription using virtual microphone arrays," Microsoft Research, Tech. Rep. MSR-TR-2019-11, Revised July 2019. [Online]. Available: https://arxiv.org/abs/1905.02545

[29] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, "Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks," in *Proc. Interspeech*, 2018, pp. 3038–3042.