Meeting Transcription Using Virtual Microphone Arrays

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

Microsoft Technical Report MSR-TR-2019-11
{tayoshi, zhuc, didimit, wihintho, xdh, anstolck, nzeng}@microsoft.com

Abstract

We describe a system that generates speaker-annotated transcripts of meetings by using a virtual microphone array, a set of spatially distributed asynchronous recording devices such as laptops and mobile phones. The system is composed of continuous audio stream alignment, blind beamforming, speech recognition, speaker diarization using prior speaker information, and system combination. With seven input audio streams, our system achieves a word error rate (WER) of 22.3% and comes within 3% of the close-talking microphone WER on the non-overlapping speech segments. The speaker-attributed WER (SAWER) is 26.7%. 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 presented system achieves a 13.6% diarization error rate when 10% of the speech duration contains more than one speaker. The contribution of each component to the overall performance is also investigated.

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 most 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], 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 meetings by using a virtual microphone array, a set of spatially distributed asynchronous recording devices such as laptops and mobile phones. The system is composed of:

- Continuous audio stream alignment
- Blind beamforming
- Speech recognition
- Speaker diarization using prior speaker information
- System combination

With seven input audio streams, our system achieves a word error rate (WER) of 22.3% and comes within 3% of the close-talking microphone WER on the non-overlapping speech segments. The speaker-attributed WER (SAWER) is 26.7%. 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 presented system achieves a 13.6% diarization error rate when 10% of the speech duration contains more than one speaker. The contribution of each component to the overall performance is also investigated.

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.

In this paper, we investigate a meeting transcription 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-
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 $w_f$ and an input multi-channel STFT coefficient vector, where subscript $f$ denotes a frequency bin index. In one formulation, the beamformer coefficient vector is estimated with a minimum variance distortionless response (MVDR) principle as

$$w_f = \Phi_{S,f}^{-1} \Phi_{d,f} r / \text{tr} (\Phi_{S,f}^{-1} \Phi_{d,f}),$$

where $\Phi_i$ denotes a spatial covariance matrix ($i = S$ for speech; $i = N$ for noise) while $r$ 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 $r$ 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 speech recognition system, consisting of a latency-controlled bidirectional long short term memory (LSTM) acoustic model (AM)
an n-gram, an LSTM rescoring language model (LM), and a weighted finite state transducer (WFST) 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 LM. Whenever a silence segment longer than 300 ms was detected, the decoder generated an n-best list, which was rescored with the LSTM-LM.

### 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 [13] 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] 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 4.5, 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 [24] 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.

### 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 native...
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. 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 [28].

---

1. This was done by using NIST’s asluite 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. Steffenhagen, R. Bowers, and J. Fiscus, Eds. Berlin: Springer, 2008, vol. 4625, pp. 373–389.

[2] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, “Achieving a human parity in conversational speech recognition,” *CoRR*, vol. abs/1610.05256, 2016. [Online]. Available: http://arxiv.org/abs/1610.05256

[3] 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,” *CoRR*, vol. abs/1703.02136, 2017. [Online]. Available: http://arxiv.org/abs/1703.02136

[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. Workshop* Automat. Speech Recognition, Understanding, 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. Pundak, K. Chint, K. C. Sim, R. J. Weiss, K. W. Wilson, E. Variani, C. Kim, O. Siohan, M. Wentrauba, 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,” *arXiv preprint arXiv:1810.04719*, 2018.

[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 asymmetric distributed microphone array using block-wise refinement of mask-based MVDR beamformer,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 5694–5698.

[11] A. Stolcke, “Making the most from multiple microphones in meeting recordings,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 4992–4995.

[12] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 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. Int. Conf. Acoust., Speech, Signal Process.*, 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. Int. Conf. Acoust., Speech, Signal Process.*, 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. Int. Conf. Acoust., Speech, Signal Process.*, 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. Int. Conf. Acoust., Speech, Signal Process.*, 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,” *CoRR*, vol. abs/1706.08612, 2017. [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. Int. Conf. Spoken Language Process.*, 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. Workshop Automat. Speech Recognition, Understanding*, 1997, pp. 347–354.

[23] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gaddé, 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 Worksh.*, 2000.

[24] G. Evermann and P. Woodland, “Posterior probability decoding, confidence estimation, and system combination,” in *Proc. NIST Speech Transcription Worksh.*, 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. Int. Conf. Acoust., Speech, Signal Process.*, 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 simultaneous speech,” in *Proc. Int. Conf. Language Resources. Evaluation*, 2006, pp. 803–808.

[28] 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.

[29] *Proc. NIST Speech Transcription Worksh.*, 2000.