8th International Workshop on Speech Processing in Everyday Environments (CHiME 2024)
6 September 2024, Kos, Greece

BUT/JHU System Description for CHiME-8 NOTSOFAR-1 Challenge

Alexander Polok[1], *Dominik Klement*[1,2], *Jiangyu Han*[1], *Šimon Sedláček*[1], *Bolaji Yusuf*[1], *Matthew Maciejewski*[2], *Matthew Wiesner*[2], *Lukáš Burget*[1]

[1]Brno University of Technology
[2]Johns Hopkins University

ipoloka@fit.vutbr.cz, xkleme15@stud.fit.vutbr.cz, ihan@fit.vutbr.cz, saimon.sedlacek@gmail.com, iyusuf@fit.vutbr.cz, matt@mmaciejewski.com, wiesner@jhu.edu, burget@fit.vutbr.cz

Abstract

This paper presents our method for tackling the CHIME-8 challenge's NOTSOFAR-1 task, which requires participants to perform multi-speaker automatic speech recognition (ASR) using audio from distant microphone arrays. We modify the Pyannote3 diarization pipeline, incorporating pre-trained WavLM as local EEND to adapt effectively to new domains, and we introduce two diarization-aware approaches to ASR by conditioning Whisper on diarization outputs for target-speaker ASR. The first method, which we refer to as Query-Key Biasing, modifies Whisper's attention mechanism and positional embeddings with a learnable attention mask to exclude non-target speaker segments in the audio. The second method, called Frame-Level Diarization-Dependent Transformations, applies affine, diarization-dependent transformations with trainable parameters to the inputs of one or more transformer blocks. We also extend both the ASR and diarization systems to a multichannel setup by incorporating cross-channel communication into our models. Finally, we report the performance of these approaches on the NOTSOFAR-1 dataset.

Index Terms: multi-talker speech recognition, CHiME-8, NOTSOFAR-1, target-speaker

1. Introduction

Self-supervised models [1, 2, 3], LLMs [4, 5] and Whisper-style supervised models [6, 7], have demonstrated that scaling up models to use more parameters and large amounts of data can enable the development of high-performing automatic speech recognition (ASR) systems, even in relatively challenging environments. However, these models have primarily been used in single-speaker, single-channel ASR systems, whereas most conversations are multi-talker, often recorded with multiple microphones. Approaches to handle this scenario generally combine multiple systems that perform source separation, speaker segmentation, overlapped speech detection, post hoc speaker clustering, and ASR to produce speaker-attributed transcripts.

We aim to use Whisper in a relatively simple setup that avoids many of these components. At a high level, we first diarize the conversations and then fine-tune Whisper to perform target speaker ASR. However, rather than conditioning on speaker embeddings, we propose two methods for conditioning directly on frame-level diarization outputs. The advantage of this approach is that the ASR system not only has access to an instance of the target speaker speech, but also the labeled instances of non-target speech. For the Whisper fine-tuning, we use the ground truth speaker segmentation. For each training example, one of the speakers in the input conversation is designated as the target speaker and only the transcript of his/her speech is considered as the target output sequence. Training segments that have transcripts for more than one speaker are repeated during training so that each speaker in the conversation can play the role of the target speaker.

The first of our methods, dubbed Query-Key Biasing, utilises the diarization outputs to produce a target speaker mask. This mask expands keys and queries to modify attention scores and focus better on the desired speaker frames. As the masked audio may contain overlapped speech, this approach, in theory, trains Whisper to perform target speaker ASR while handling large regions of silence and non-target speaker speech.

In the second approach, named Frame-Level Diarization Dependent Transformations (FDDT), the model has more fine-grained access to diarization outputs, which are incorporated by applying learned frame-level transformations corresponding to target-speaker and non-target-speaker audio at the input of transformer blocks in the encoder.

Some of our fine-tuned models suffered from some of the well-known problems of the original Whisper model: hallucinations, especially in regions of silence, and unreliable timing information. For these reasons, we also fine-tune the Whisper model using the CTC objective. All our code is publicly available at `https://github.com/BUTSpeechFIT/CHiME-8_NOTSOFAR-1`.

1.1. Related Work

Target speaker ASR (TS-ASR) models generally rely on speaker-specific information [8, 9, 10, 11], such as target speaker enrollment or existing speaker embeddings, or embedding models such as x-vectors extractors [12]. TS-ASR models trained on a limited number of speakers, as is the case with the NOTSOFAR-1 challenge [13], may struggle to generalize effectively to new speakers or different acoustic conditions. Training these models often requires additional components, such as x-vector extractors, in order to compute speaker embeddings. Other approaches to target speaker ASR first attempt to separate the speech and then assign speech in the separated streams to speakers, again using some enrollment speech or embeddings [14]. More recently, especially in the streaming scenario, this enrollment speech has been provided by "prompting" the model with examples of previously recognized speech [15, 16], or instruction tuning of Whisper style models [17].

2. Target Speaker ASR

This section describes two methods for conditioning Whisper on diarization outputs and extending the Vanilla Whisper.

10.21437/CHiME.2024-4

2.1. Query-Key Biasing for Target Speaker ASR

One way to condition on frame-level diarization outputs is to use them as target speaker time masks when transcribing audio segments with one or more speaker turns.

For simplicity, let us assume the number of attention heads is one. Let $W_q, W_k \in \mathbb{R}^{d\times d}$ be the query and key projection matrices and $q_i, k_j \in \mathbb{R}^d$ the query and key respectively, where d is the attention embedding dimensionality. The attention score between q_i, k_j is computed as:

$$a_{ij} = softmax(\frac{(W_q q_i)^T (W_k k_j))}{\sqrt{d}}). \tag{1}$$

If we assume that acoustic information is aligned across time, masking out non-target speaker frames forces Whisper to ignore information irrelevant to the target speaker transcript (i.e. other speakers, silence, etc.). However, pure attention masking leaves Whisper no chance for unmasking and possibly attending to non-target frames, which makes adaptation and speaker tracking learning impossible.

As a solution, we decided to bias the encoder self-attention and the decoder cross-attention by extending queries, keys and initializing corresponding projections in the following way:

$$\hat{q}_i = \begin{bmatrix} q_i \\ 1 \end{bmatrix}, \hat{k}_j = \begin{bmatrix} k_j \\ -c \end{bmatrix}, \hat{W}_{q,k} = \begin{bmatrix} W_{q,k} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2}$$

where $c \in \mathbb{R}_0^+$ is a bias factor that is set to 0 if k_j corresponds to a target speaker frame, and to a predefined constant otherwise.

It is easy to observe that, after initialization, if k_i represents a target speaker frame, a_{ij} remains intact. On the other hand, if k_j represents a non-target speaker frame, the dot-product inside the softmax changes as:

$$\hat{W}_q \hat{q}_i = \begin{bmatrix} W_q q_i \\ 1 \end{bmatrix}, \hat{W}_k \hat{k}_j = \begin{bmatrix} W_k k_i \\ -c \end{bmatrix}, \tag{3}$$

$$\begin{bmatrix} (W_q q_i)^T & 1 \end{bmatrix} \begin{bmatrix} W_k k_i \\ -c \end{bmatrix} = (W_q q_i)^T (W_k k_j) - c. \tag{4}$$

It is important to note that fine-tuning the Whisper model with extended queries and keys changes the extended attention projection matrices, which controls the level of attention biasing.

2.1.1. Shifted Positional Embedings

Masked silences within an utterance can lead to hallucinations and instability during Whisper's training. The main reason is that the decoder is cross-attending to discontinuous parts of the encoder embedding sequence. Therefore, we shift the positional embeddings on target speaker frames and repeat the previous ones on the non-target ones, which ensures that the embeddings the decoder attends to have continuous positions.

2.2. Frame-Level Diarization Dependent Transformations

As an alternative to query-key biasing, we can augment the frame-level speech representations with an encoding of the *type* of speech that is present in each frame of the audio. Figure 1 depicts our approach, dubbed Frame-Level Diarization Dependent Transformations (FDDT), which is described in detail below.

Based on the diarization output, to each speech frame, we assign one of four possible *STNO* labels: Silence, Target-speaker speech, Non-target speaker(s) and Overlapping speech containing the target-speaker. Here, we convert Whisper into a target-speaker ASR model by inserting an affine transformation (diagonal linear transformation and bias) at the input of every encoder self-attention layer, which operates frame-by-frame on the hidden representations. This affine transformation is conditioned on the *STNO* labels, and it, therefore, changes from frame to frame depending on the frame label, i.e., we have 4 different affine transformations (for each label), and the *STNO* labels select which one is used for which frame. The transformations are initialized to identity functions so as not to disrupt the data flow in the well-trained Whisper model. The parameters of all the affine transformations are the new trainable parameters, which are fine-tuned together with all the original Whisper parameters for the target-speaker ASR task.

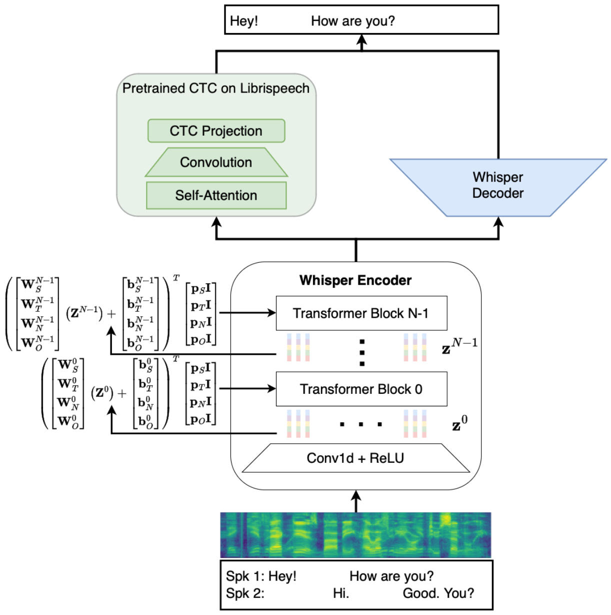
Hey!
How are you?
Pretrained CTC on Librispeech
CTC Projection
Convolution
Self-Attention
Whisper Decoder
Whisper Encoder
Transformer Block N-1
Transformer Block 0
Conv1d + ReLU
$\left(\begin{bmatrix} \mathbf{W}_S^{N-1} \\ \mathbf{W}_T^{N-1} \\ \mathbf{W}_N^{N-1} \\ \mathbf{W}_O^{N-1} \end{bmatrix} (\mathbf{Z}^{N-1}) + \begin{bmatrix} \mathbf{b}_S^{N-1} \\ \mathbf{b}_T^{N-1} \\ \mathbf{b}_N^{N-1} \\ \mathbf{b}_O^{N-1} \end{bmatrix}\right)^T \begin{bmatrix} \mathbf{p}_S\mathbf{I} \\ \mathbf{p}_T\mathbf{I} \\ \mathbf{p}_N\mathbf{I} \\ \mathbf{p}_O\mathbf{I} \end{bmatrix}$
$\left(\begin{bmatrix} \mathbf{W}_S^{0} \\ \mathbf{W}_T^{0} \\ \mathbf{W}_N^{0} \\ \mathbf{W}_O^{0} \end{bmatrix} (\mathbf{Z}^{0}) + \begin{bmatrix} \mathbf{b}_S^{0} \\ \mathbf{b}_T^{0} \\ \mathbf{b}_N^{0} \\ \mathbf{b}_O^{0} \end{bmatrix}\right)^T \begin{bmatrix} \mathbf{p}_S\mathbf{I} \\ \mathbf{p}_T\mathbf{I} \\ \mathbf{p}_N\mathbf{I} \\ \mathbf{p}_O\mathbf{I} \end{bmatrix}$
$\mathbf{z}^{N-1}$
$\mathbf{z}^{0}$
Spk 1: Hey! How are you?
Spk 2: Hi. Good. You?

Figure 1: *Proposed Diarization-Conditioned Model. An input audio segment with possibly more than one speaker is extended with hard or soft frame-level diarization outputs* $\begin{bmatrix} p_S^t & p_T^t & p_N^t & p_O^t \end{bmatrix}^T$ *for each of the STNO classes and every frame at time* t*. Affine transformations shown as additions to the left of the Whisper model are applied to the intermediate outputs* $\mathbf{Z}^l$*, of layer,* l*, to create new embeddings. The final frame-level embedding is a convex combination of these embeddings for each frame.*

2.3. Target-speaker ASR decoding

During inference, real diarization system outputs can be used instead of ground-truth diarization. Each input conversation is decoded once for each speaker present in the diarization system hypothesis. We use the Whisper sequential long-form decoding, which also produces timestamps for the beginning and end of each segment of each speaker. During inference, the frame-level diarization outputs can be used to make either hard or soft decisions about each frame label. When using soft decisions, the soft labels are used as weights to obtain a convex combination of the 4 affine transformations.

2.4. Hybrid CTC/Attention fine-tuning of Whisper

For the systems based on FDDT, we use the Hybrid CTC/Attention architecture [18] to mitigate the known issue of hallucinations, which can be observed for encoder-decoder models [19]. However, using CTC with the Whisper vocabulary can result in large memory requirements as well as inefficient training and inference. Therefore, for the CTC branch, we used a cascade of one self-attention layer, two convolutional layers and a CTC head (a linear layer and softmax) appended to the last layer of the Whisper encoder. The stride of two is used for each convolutional layer, resulting in a final output frame rate of 12.5 Hz (80ms for each frame). The output labels considered by the CTC head are the blank symbol and all Whisper decoder output tokens (including the time-stamp tokens). The combination of the scores from the Whisper decoder and the CTC decoder is used in the final decoding only with small models as described in [18]. Language and task-specific Whisper tokens are masked for CTC.

2.5. Multichannel ASR decoding

We apply the hybrid CTC/attention model to multichannel ASR by executing the single-channel ASR model separately on each channel in parallel. Specifically, we average the encoder activations at the 8th layer across channels and propagate the resulting averaged activation through the remaining encoder layers and the decoder. Inspired by Mošner et al. [20], we also experimented with different strategies to train the complete multichannel system and facilitate information exchange between channels, but the results were not highly promising.

3. Diarization

Any diarization system could be used in the TS-ASR approaches described in Section 2. We describe some modifications to a standard Pyannote diarization pipeline [21] that were not only helpful in the multichannel scenario but also for single-channel diarization.

3.1. Single-channel Diarization

The single-channel speaker diarization system builds upon the two-stage Pyannote diarization pipeline [21]. It first segments the input signal into overlapping segments. Local end-to-end diarization (EEND) is applied to each segment. For each segment and each speaker in the segment, an embedding is extracted using his/her non-overlapping speech. Speaker embeddings from all segments are clustered using agglomerative hierarchical clustering (AHC) to find cross-segment speaker correspondence. The AHC is applied with the constraint that embeddings coming from the same segment (which are different speakers according to EEND) do not end up in the same cluster. Finally, local EEND decisions, which are frame-by-frame speaker activity probabilities, are aggregated over the overlapping segments by averaging the probabilities of the corresponding speakers in the corresponding (overlapping) frames. The resulting soft attributions of all frames in the whole conversation to speakers form the output of the diarization system. This output can be used to derive the hard or soft STNO labels described in Section 2.2.

Our local EEND system uses a pre-trained WavLM Base+ model. Following the SUPERB strategy [22], a single sequence of features is generated by taking a weighted average of individual WavLM layer outputs. The weights are trainable parameters. This sequence is then fed to a stack of one linear layer, layer norm, 4 conformer layers and a final classification head. The classification head is trained on the local segments using the powerset loss [23]. For the EEND training, the parameters of WavLM and the newly added layers are jointly trained.

3.2. Multi-channel Diarization

Inspired by [20], we extend our single-channel system for multi-channel processing as follows: The input signal from each channel is processed in parallel by the first 4 WavLM attention layers. The hidden activations are then averaged across channels and processed by the remaining WavLM layers. The first 4 layers of WavLM are further extended to enable comparison and exchange of information across channels. This allows the model to access cross-channel information, which can help in extracting important clues for diarization, such as identifying the direction of arrival. Let $\mathbf{H}_c^l$ be the activations at the output of l-th WavLM layer from c-th channel. For the communication between channels, we first calculate the average activations

$$\mathbf{T}^l = \frac{1}{C}\sum_{c=1}^{C}\mathbf{H}_c^l. \tag{5}$$

These average activations are then frame-wise concatenated with the original channel-specific activations and transformed by a Linear layer back to the original dimensionality as

$$\bar{\mathbf{T}}_c^l = \mathrm{LN}(\mathrm{Linear}(\mathbf{H}_c^l||\mathbf{T}^l)), \tag{6}$$

where $||$ and LN are concatenation and layer normalization, respectively. The resulting sequence is added to the original channel-specific activations, which gives us a new modified activation sequence

$$\hat{\mathbf{H}}_c^l = \mathbf{H}_c^l + \bar{\mathbf{T}}_c^l, \tag{7}$$

which serves as the input to the following WavLM layer processing c-th channel. The layer normalization has a trainable multiplicative parameter which is initialized to a very small value (1e-2). Therefore, at the initialization, the matrices $\bar{\mathbf{T}}_c^l$ also have very small values, so they do not disrupt the data flow in the pre-trained WavLM model. Then the input to the following Conformer layers is the weighted sum of averaged representations from all channels.

3.3. Diarization Systems

We used two diarization systems for the challenge: a single-channel system and a multi-channel system. The single-channel diarization system is described in Section 3.1. Specifically, we use 8s overlapping segments with 0.8s shift. The Local end-to-end diarization (EEND) is applied to each segment, where at we allow at most 4 speakers per segment. For the powerset loss, we assume a maximum of 4 speakers and 2 overlapping speakers (forming 11 powerset classes).

The model is trained on the compound data set of the AMI MixArray and beamformed CHiME-6 training data, the 200h simulated data provided by the NOTSOFAR-1 organizers, and the NOTSOFAR-1 training data. The dev2 set is used for validation. The ResNet-34-based x-vector extractor trained using the WeSpeaker toolkit [24] on the VoxCeleb2 dataset is used to extract the local speaker embeddings.

For the multi-channel system, we trained on the compound dataset of AMI, 200h simulated data, and NOTSOFAR-1 training data to train our multi-channel model.

4. Submitted systems

Results of the submitted systems in terms of time-constrained minimum permutation Word Error Rate (tcpWER) are shown in Table 1. We also report the diarization error rate (DER) of the diarization systems used for each system.

4.1. FDDT Large – single channel (System 1)

This system uses the Frame-Level Diarization Dependent Transformations approach to the Target Speaker ASR described in Section 2.2. It is based on the Whisper Large v3 model fine-tuned on single-channel NOTSOFAR-1 training data and beamformed AMI. The Hybrid CTC/Attention approach described in Section 2.4 is used to guide the fine-tuning of the system. However, the CTC is not used during decoding as it did not improve the performance of this particular system in our experiments.

The model is trained in three stages. First, the parameters of the original Whisper encoder are frozen, and the newly added CTC parameters are trained for 2 epochs on 960 hours of Librispeech training data. Secondly, the CTC head is trained for 2 epochs on the single-channel NOTSOFAR-1 training data and beamformed AMI training data. Finally, the whole system is then fine-tuned for 10 epochs using both CTC and the sequence cross-entropy objective as in [24].

Hard ground-truth *STNO* labels are used during training. However, for the decoding of the development and evaluation data, we found improved performance when using the soft probabilistic *STNO* labels derived from the actual output of our diarization system, as described in Section 3.

The approximate total time for inference on NOTSOFAR-1 eval-small was 4.83 hours, 10 NVIDIA RTX A5000 GPUs, and 40 Intel(R) Xeon(R) CPU E5-2640 v4 CPUs were used. Fine-tuning the model took approximately 20 hours on 4 NVIDIA RTX A6000. 1065 hours of speech, including AMI and Librispeech were used.

4.2. FDDT Large+FT – single channel (System 2)

Th system from the previous section is further fine-tuned on the NOTSOFAR-1 train and dev1 data for 2 epochs.

4.3. QK-Bias Large – single channel (qk_biasing)

This system uses the query-key biasing approach to the Target Speaker ASR described in Section 2.1. It is based on the Whisper Large v3 model fine-tuned on single-channel NOTSOFAR-1 training data and beamformed AMI. During the decoding, the query-key biasing is driven by the hard decisions made by the single-channel variant of our diarization system described in Section 3. The Hybrid CTC/Attention approach described in Section 2.4 is not used for the system.

4.4. FDDT Small – multi-channel (system1_mc)

This system is the multi-channel variant of the Target Speaker ASR based on the FDDT approach described in Sections 2.2 and 2.5. It is based on the Whisper Small model fine-tuned on single-channel NOTSOFAR-1 training data and beamformed AMI. The Hybrid CTC/Attention approach described in Section 2.4 is used during both system fine-tuning and decoding.

4.5. GSS Med – multi-channel (system2_mc)

Our second multichannel system stands as a simple juxtaposition to the original multichannel baseline, attempting to improve on some of the components by utilizing our diarization approach and a fine-tuned ASR model. First, we run multichannel diarization on the NOTSOFAR-1 data as described in Section 3. Using the diarization outputs, we perform multichannel source separation and enhancement using Guided Source Separation (GSS) [25, 26] to obtain enhanced speech segments for individual speakers. We subsequently fine-tune Whisper Medium on the GSS segments produced for the train set of NOTSOFAR-1. The fine-tuning runs for up to 5 epochs, using the dev2 GSS segments for validation. We select the best checkpoint in terms of raw WER performance (20.5%) on the dev2 GSS segments. We run decoding with a beam size of 10 and a length penalty of 0.5.

Table 1: *Comparison of Baseline non-target-speaker ASR models and the target-speaker models developed during the CHiME-8 challenge on the NOTSOFAR-1 dev2 set. DER for the corresponding diarization system is also reported. Query-key biasing (QK-Bias) appears to perform slightly worse than Frame-Level Diarization Dependent Transformations (FDDT).*

Model	tcpWER [%]	DER [%]
Baseline SC	45.8	-
Baseline MC	31.6	
QK-Bias Med	51.3	10.9
QK-Bias Large	48.7	
FDDT Large – sc	36.5	
FDDT Large+FT – sc	35.9	
FDDT Small – mc	36.9	10.4
FDDT Large – mc	33.2	
GSS Med – mc	29.6	

5. Conclusions

We proposed and tested two methods for addressing the speaker-attributed ASR problem by converting pre-trained supervised ASR models into target-speaker ASR models, directly conditioned on diarization outputs. Our best models outperformed the baseline without requiring any additional speech separation components.

However, we identified several limitations in our work. Firstly, it remains unclear to what extent the method itself contributes to the improvements, as opposed to the gains resulting from adapting the baseline system to the specific dataset. Secondly, it is uncertain whether the method generalizes well across different datasets or whether the method can be effectively applied to other pre-trained systems. Additionally, some experiments incorporated a CTC head while others did not, and different model sizes were employed, making it difficult to accurately assess the improvements.

Since the submission of this paper, and during the preparation of the camera-ready version, we have addressed some of these limitations. Specifically, we evaluated our system on the AMI, NOTSOFAR-1, and Libri2Mix datasets, demonstrating the generalization of the approach across different datasets. We also analyzed the impact of including the CTC head, using more data and parameters, and proposed an improved method for initializing new parameters [27]. For our diarization system, more experiments and analysis can be found in [28].

6. Acknowledgements

The work was supported by Czech Ministry of Interior projects Nos. VK01020132 ”112” and VB02000060, ”NABOSO”, by European Union’s Horizon Europe project No. SEP-210943216

”ELOQUENCE” and by Horizon 2020 Marie Sklodowska-Curie grant ESPERANTO, No. 101007666. Computing on IT4I supercomputer was supported by the Ministry of Education, Youth and Sports of the Czech Republic through e-INFRA CZ (ID:90254).

7. References

[1] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[2] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, “Hubert: How much can a bad teacher benefit asr pre-training?” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 6533–6537.

[3] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi *et al.*, “Scaling speech technology to 1,000+ languages,” *Journal of Machine Learning Research*, vol. 25, no. 97, pp. 1–52, 2024.

[4] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.

[5] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[7] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan, Y. Sudo, M. Shakeel, K. Choi, J. Shi, X. Chang *et al.*, “Owsm v3. 1: Better and faster open whisper-style speech models based on e-branchformer,” *arXiv preprint arXiv:2401.16658*, 2024.

[8] N. Kanda, S. Horiguchi, R. Takashima, Y. Fujita, K. Nagamatsu, and S. Watanabe, “Auxiliary interference speaker loss for target-speaker speech recognition,” *arXiv preprint arXiv:1906.10876*, 2019.

[9] T. Moriya, H. Sato, T. Ochiai, M. Delcroix, and T. Shinozaki, “Streaming target-speaker asr with neural transducer,” *arXiv preprint arXiv:2209.04175*, 2022.

[10] Z. Huang, D. Raj, P. García, and S. Khudanpur, “Adapting self-supervised models to multi-talker speech recognition using speaker embeddings,” in *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP).* IEEE, 2023, pp. 1–5.

[11] N. Kanda, G. Ye, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, “End-to-end speaker-attributed asr with transformer,” *arXiv preprint arXiv:2104.02128*, 2021.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP).* IEEE, 2018, pp. 5329–5333.

[13] A. Vinnikov, A. Ivry, A. Hurvitz, I. Abramovski, S. Koubi, I. Gurvich, S. Peer, X. Xiao, B. M. Elizalde, N. Kanda, X. Wang, S. Shaer, S. Yagev, Y. Asher, S. Sivasankaran, Y. Gong, M. Tang, H. Wang, and E. Krupka, “Notsofar-1 challenge: New datasets, baseline, and tasks for distant meeting transcription,” in *Interspeech 2024*, 2024, pp. 5003–5007.

[14] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, “Single channel target speaker extraction and recognition with speaker beam,” in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP).* IEEE, 2018, pp. 5554–5558.

[15] S. Cornell, J.-w. Jung, S. Watanabe, and S. Squartini, “One model to rule them all? towards end-to-end joint speaker diarization and speech recognition,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2024, pp. 11 856–11 860.

[16] D. Raj, M. Wiesner, M. Maciejewski, L. P. Garcia-Perera, D. Povey, and S. Khudanpur, “On speaker attribution with surt,” *arXiv preprint arXiv:2401.15676*, 2024.

[17] H. Ma, Z. Peng, M. Shao, J. Li, and J. Liu, “Extending whisper with prompt tuning to target-speaker asr,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2024, pp. 12 516–12 520.

[18] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, “Hybrid ctc/attention architecture for end-to-end speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[19] Y. Peng, Y. Sudo, M. Shakeel, and S. Watanabe, “Owsm-ctc: An open encoder-only speech foundation model for speech recognition, translation, and language identification,” *arXiv preprint arXiv:2402.12654*, 2024.

[20] L. Mošner, R. Serizel, L. Burget, O. Plchot, E. Vincent, J. Peng, and J. Černocký, “Multi-Channel Extension of Pre-trained Models for Speaker Verification,” in *Proc. INTERSPEECH 2024, accepted*, 2023.

[21] H. Bredin, “pyannote. audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe,” in *Proc. Interspeech 2023*, 2023, pp. 1983–1987.

[22] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, “SUPERB: Speech Processing Universal PERformance Benchmark,” in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[23] A. Plaquet and H. Bredin, “Powerset multi-class cross entropy loss for neural speaker diarization,” *arXiv preprint arXiv:2310.13025*, 2023.

[24] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, “Wespeaker: A research and production oriented speaker embedding learning toolkit,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2023, pp. 1–5.

[25] N. Kanda, C. Böddeker, J. Heitkaemper, Y. Fujita, S. Horiguchi, K. Nagamatsu, and R. Haeb-Umbach, “Guided source separation meets a strong ASR backend: Hitachi/paderborn university joint investigation for dinner party ASR,” in *20th Annual Conference of the International Speech Communication Association, Interspeech 2019, Graz, Austria, September 15-19, 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 1248–1252. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-1167

[26] D. Raj, D. Povey, and S. Khudanpur, “GPU-accelerated Guided Source Separation for Meeting Transcription,” in *Proc. INTERSPEECH 2023*, 2023, pp. 3507–3511.

[27] A. Polok, D. Klement, M. Wiesner, S. Khudanpur, J. Černocký, and L. Burget, “Target speaker asr with whisper,” 2024. [Online]. Available: https://arxiv.org/abs/2409.09543

[28] J. Han, F. Landini, J. Rohdin, A. Silnova, M. Diez, and L. Burget, “Leveraging self-supervised learning for speaker diarization,” 2024. [Online]. Available: https://arxiv.org/abs/2409.09408