DOI: dx.doi.org/10.1109/ICASSP40776.2020.9053152

A MEMORY AUGMENTED ARCHITECTURE FOR CONTINUOUS SPEAKER IDENTIFICATION IN MEETINGS

Nikolaos Flemotomos[1*], *Dimitrios Dimitriadis*[2]

[1] Signal Analysis and Interpretation Lab, University of Southern California, Los Angeles, CA, USA
[2] Speech and Dialog Research Group, Microsoft, Redmond, WA, USA

ABSTRACT

We introduce and analyze a novel approach to the problem of speaker identification in multi-party recorded meetings. Given a speech segment and a set of available candidate profiles, a data-driven approach is proposed learning the distance relations between them, aiming at identifying the correct speaker label corresponding to that segment. A recurrent, memory-based architecture is employed, since this class of neural networks has been shown to yield improved performance in problems requiring relational reasoning. The proposed encoding of distance relations is shown to outperform traditional distance metrics, such as the cosine distance. Additional improvements are reported when the temporal continuity of the audio signals and the speaker changes is modeled in. In this paper, the proposed method is evaluated in two different tasks, i.e. scripted and real-world business meeting scenarios, where a relative reduction in speaker error rate of 39.28% and 51.84%, respectively, is reported when compared with the baseline.

Index Terms— speaker identification, diarization, memory networks, meeting analysis

1. INTRODUCTION

Speaker identification is the task of determining the identity of the person uttering a particular phrase, assuming a finite set of pre-enrolled speakers is given [1]. Applying a continuous automatic speaker identification system on recorded meetings with multiple participants can significantly affect the performance of several subtasks of the meeting analytics suite. For instance, correctly identifying the active speaker is an essential component for rich meeting transcriptions (Speaker-Attributed Automatic Speech Recognition - SA-ASR) [2, 3], speaker tracking [4], action item generation [5], or speaker adaptation for more reliable ASR outputs [6]. The main difference of the investigated task with speaker diarization is the use of speaker profiles, since the meeting participants are known in advance [7], i.e. the number of speakers and their acoustic identities are provided.

For speaker attribution tasks, an enrollment phase is required. During that phase, sample audio from the participants is collected and the target speaker profiles (or identities) are constructed. Continuous speaker identification can be thought of as a two-step problem, with a segmentation and a classification phase. First, the audio signal is segmented either uniformly [8] or based on estimated speaker change points [3]. These segments are assumed speaker-homogeneous[1]. Speech embeddings of each segment are extracted and then compared against all the available speaker profiles. By minimizing a particular distance metric (as described below), the most suitable speaker label is assigned to the segment [3]. Initially, the extracted speaker-specific features/embeddings were based on the total variability model [9], but lately bottleneck representations from deep learning architectures, such as the x-vectors [10], are used. The final decision relies either on the cosine [9, 3] or the PLDA [11] distance. The overall process is illustrated in Fig. 1.

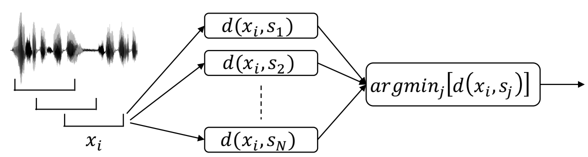


Fig. 1: State-of-the-art continuous speaker identification system: The speech signal is segmented uniformly and each segment x_i is compared against all the available speaker profiles $\{s_j\}_{j=1}^{N}$ according to a distance metric $d(\cdot,\cdot)$. A speaker label is assigned to each x_i minimizing this metric.

This approach poses some potential problems. First, uniform segmentation introduces a trade-off error related to the segment length: segments need to be sufficiently short to safely assume that they do not contain multiple speakers but at the same time it is necessary to capture enough acoustic information to extract a meaningful speaker representation. Further, the speaker embeddings are usually extracted from a network trained to distinguish speakers among thousands of candidates [10]. However, a different level of granularity in the speaker space is required, since only a small number of participants is typically involved in an interactive meeting scenario. Also, the distance metric used is often heuristic and/or dependent on certain assumptions which do not necessarily hold, e.g., assuming Gaussianity in the case of PLDA distance [11]. Finally, the audio chunks are treated independently and any temporal information about the past and future is simply ignored.

In this work, a data-driven, memory-based approach is proposed addressing some of the aforementioned challenges. Data-driven techniques perform remarkably well on a wide variety of tasks [12]; traditional architectures, though, may fail when the problem involves relational information between observations [13]. Speaker identification can be seen as a member of this class of tasks, since the final decision depends on the distance relations between speech segments and speaker profiles. Herein, the Memory-Augmented Neural Networks (MANNs) [14, 15] are proposed bridging the performance gap. Based on the success of MANNs on several problems requiring relational reasoning and specifically using the Relational Memory Core (RMC) [13], we build a memory-based network for the task of

[*] Nikolaos Flemotomos performed this work while at Microsoft.
[1] A single speaker is present.

continuous speaker identification in meeting scenarios. While compared to the baseline approach, we show consistent improvements in performance.

2. METHOD

2.1. MANNs and RMC

The main concept of MANNs is augmenting a recurrent neural network with a memory matrix $\mathbf{M} \in \mathbb{R}^{Q \times P}$ consisting of Q P-dimensional memory slots. The main architecture, called the controller, decides how to update the memory through attention mechanisms using read and write heads (Fig. 2). The entire system is differentiable, consequently it can learn a task-specific organization of the memory in a supervised way through gradient descent [16].

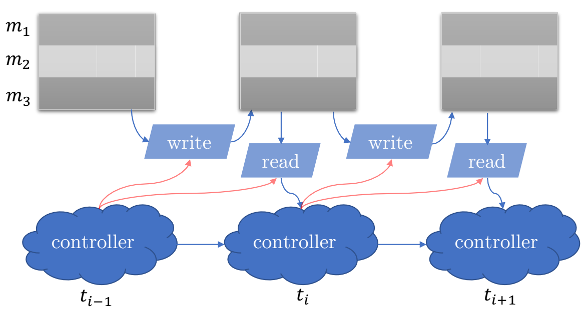


Fig. 2: Graphical illustration of how a MANN operates through time. The memory matrix $\mathbf{M}$ here consists of 3 memory slots $\{m_i\}_{i=1}^{3}$.

Several implementations of this class of networks have been proposed in the literature, e.g., [15, 17]. For our task, the framework introduced in [13] is used, where the controller of the network is embedded into a Long Short-Term Memory (LSTM) cell and is called "Relational Memory Core" (RMC). RMC controls the memory updates through a self-attention mechanism [18] in a way that the memory matrix dimensions remain constant. Each time a new observation is presented, self-attention allows each memory to attend to that observation, as well as to all the other memories, before being updated, so that cross-memory relations are encoded.

2.2. RMC-based architecture for speaker identification

RMC-based networks have shown good performance on several problems [13], including the "n^{th} farthest task", where the goal is to find the n^{th} farthest vector from the m^{th} element in a given sequence of vectors. Assume we are given a particular audio segment of a meeting x_i and a set of profiles $\{s_j\}_{j=1}^{N}$ corresponding to the N participants in the meeting. Under the n^{th} farthest task notation, we construct the sequence $S_i = \{x_i, s_1, s_2, \cdots, s_N\}$ and we view speaker identification as the problem of finding the closest element to x_i in the sequence S_i. In more detail, a sequence of vectors S_i for each audio segment x_i of the input audio signal is passed through an RMC-based recurrent layer, as in Fig. 3. The output of this layer goes through a fully-connected Multilayer Perceptron (MLP) with a softmax inference layer returning the label $l_i \in \{1, 2, \cdots, N\}$; that is the one maximizing the probability $\mathbb{P}\left[l_i = j | x_i, \{s_j\}_{j=1}^{N}\right]$. Intuitively, the RMC projects each element of the input sequence onto the "memory space" and the network learns some local data-driven distance metric, sorts the resulting distances, and finds the profile that yields the minimum distance.

Note that N is a prefixed maximum number of speakers within a meeting that the network can handle. Given a sequence with $\tilde{N} < N$ profiles, the remaining $N - \tilde{N}$ outputs of the softmax layer are expected to be close to zero. To that end, the network is trained with variable length sequences, providing training examples with all the expected numbers of participants.

As shown in Section 4, the proposed architecture needs a large number of training sequences, containing many speaker profiles. The network is prone to overfitting when the in-domain meeting data are limited. Instead, an out-of-domain dataset can be used to construct speaker profiles and generate sequences of speech segments and random profiles[2]. In the case when multiple training sequences contain the same speaker profiles, e.g., long real-world meeting data are used during training, it is necessary to steer the network to learn distance relations and not specific positions of the profiles within the sequences. For that reason, the profiles $\{s_j\}_{j=1}^{N}$ of each sequence are randomly permuted during training.

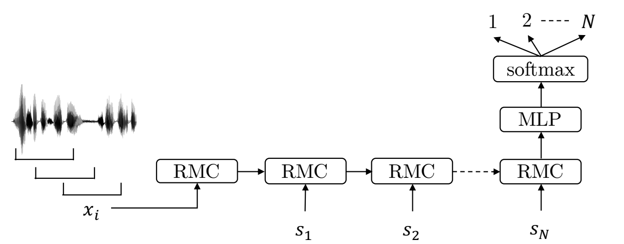


Fig. 3: Unrolled recurrent network for continuous speaker identification. x_i represents a speech segment and $\{s_j\}_{j=1}^{N}$ are the profiles of the N speakers appearing in the recording.

2.3. Incorporating temporal information

As mentioned in the Introduction, one of the challenges of the traditional approaches is the lack of temporal information. However, adding temporal context in the proposed approach is straightforward, by constructing the input sequences while including the information from neighboring segments. For example, to identify the speaker profile that corresponds to x_i with a temporal context of 1, we would use the sequence $\{x_{i-1}, x_i, x_{i+1}, s_1, s_2, \cdots, s_N\}$, where $\{s_j\}_{j=1}^{N}$ are the candidate speaker profiles.

Temporal continuity is a well-known issue at the decision level as well since it is highly improbable that isolated short segments correspond to some speaker in the middle of an utterance assigned to another speaker. Within the diarization community, this is often addressed through Viterbi or Variational Bayes resegmentation [19, 20]. For this work we apply a simple smoothing filter of the trajectory of the predicted speaker labels through median filtering. Similarly, a Hidden Markov Model (HMM) could be introduced for this trajectory smoothing.

3. DATASETS

The publicly available AMI corpus [21] consists of meeting data, either occurring naturally or following a scripted scenario. For our experiments we use the scripted meetings with 4 speakers each, with both close-talk and far-field audio available, giving us 31 scenarios. Each scenario consists of 4 meetings happening throughout a

[2]The only constraint being that the ground truth profile corresponding to a segment should be included in the training sequence.

day: the first set of meetings is used for the speaker profile estimation (totally $8.0\,\mathrm{h}$ ignoring silence), the second and third for training ($35.5\,\mathrm{h}$), and the fourth for evaluation ($17.1\,\mathrm{h}$). We refer to this evaluation set as the `seen` one, since the speakers are seen during training. An additional evaluation set with `unseen` speakers is used, consisting of 6 meetings ($4.1\,\mathrm{h}$) – while 6 more meetings ($3.8\,\mathrm{h}$) with the same speakers are used for their profile extraction. To resemble real-world conditions, far-field audio is used for training and evaluation, while the profile estimation is based on the close-talk microphones, as would normally be done during the enrollment phase.

To introduce more speaker variability during training, we additionally use data from VoxCeleb 1 and 2 [22, 23]. All the speakers with more than 6 utterances each are kept in, resulting in a subset of $6,490$ speakers. For each speaker, 3 utterances are randomly selected for the profile estimation (totalling $383.3\,\mathrm{h}$) and the rest ($2184.9\,\mathrm{h}$) are used for training.

The method is also evaluated on 9 real meetings recorded within Microsoft, with a circular microphone array [24]. The seven channels of the array are combined through a differential beamformer, processing the beam output with the highest energy. The number of speakers in those meetings (total length of $4.6\,\mathrm{h}$) ranges from 4 to 15 and all the speakers had been already enrolled while reading short text excerpts with a close-talk microphone.

4. EXPERIMENTS AND RESULTS

4.1. Experimental setup

The ground truth Voice Activity Detection (VAD) segmentation as provided by the available human-generated transcripts is used, while ignoring the speaker labels. Consecutive speech segments with an in-between silence shorter than $0.5\,\mathrm{sec}$ are now merged. This segmentation scenario will be noted as the `oraclevad`, as opposed to the `oraclespk`, where the initial speaker label info is also included. The main difference between these two scenarios is that segments in `oraclevad` may contain more than one speakers, while each segment in `oraclespk` contains a single speaker. While generating the training and evaluation sequences, a sliding analysis window with length equal to $1.5\,\mathrm{sec}$ is used for the x-vector extraction. The window shift is $0.75\,\mathrm{sec}$ for the AMI and the internal meeting dataset, and $10\,\mathrm{sec}$ for the VoxCeleb dataset. This process creates 160K training examples for AMI and 841K subsegments for VoxCeleb. A 512-dimensional x-vector is generated per window, using the pretrained VoxCeleb model[3] provided by the Kaldi toolkit [25]. The x-vectors are decorrelated via an LDA projection (after which we keep 200 dimensions) and are further mean- and length-normalized [11]. A speaker profile is estimated as the per speaker mean of all the x-vectors estimated on the available speaker-homogeneous segments, in the `oraclespk` scenario.

The memory matrix has $(N + 1)$ 2048-dimensional memory slots, where N is the maximum number of speakers the network expects to see. The MLP component consists of 4 fully connected layers of 256 neurons each. The network is built using TensorFlow [26] and the Sonnet library[4].

In the case of `oraclespk` segmentation, the evaluation metric is based on the window-level (subsegment-level) classification accuracy. For the `oraclevad` segmentation scenario, we report the Speaker Error Rate (SER) as estimated by the NIST `md-eval.pl` tool, with a $0.25\,\mathrm{sec}$-long collar, while ignoring overlapping speech segments. The system described in Fig. 1 using the cosine distance provides the baseline performance.

4.2. Results on AMI

First, the system is trained and evaluated on the AMI corpus using the `oraclespk` segmentation. Besides the baseline and the RMC-based systems, a third one, replacing the RMC-based layer of Fig. 3 with an LSTM layer, is also evaluated. The RMC appears capturing the desired distance information better than the LSTM, but both networks are outperformed by the baseline on the `unseen`-speakers scenario, as shown in Fig. 4. In particular, the performance of the baseline system is practically the same for all testing conditions (i.e., `seen` vs. `unseen`). This is expected as the system is based solely on the cosine distances without any supervision. On the contrary, the behavior of the recurrent networks appears quite different with a big performance gap between the `seen` and `unseen` evaluation sets. A possible explanation is that the networks overfit to the training speakers and fail to generalize well. Since the LSTM-based network performs substantially worse in the `unseen` conditions than the RMC-based one, we henceforth use only the latter.

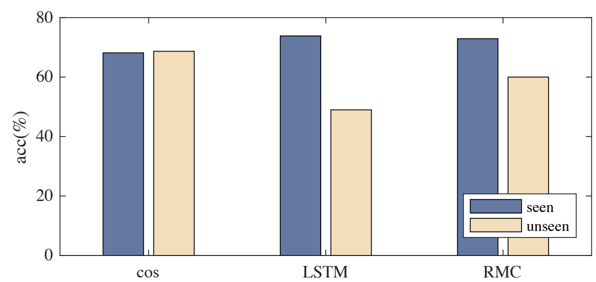


Fig. 4: Classification accuracy on AMI for the `seen` and `unseen` evaluation sets with `oraclespk` segmentation. Both the LSTM- and RMC-based networks are trained on AMI.

To avoid overfitting during training because of the limited number of training speakers in AMI, we decided to use randomly generated training sequences from VoxCeleb, as described in Section 2.2. Note here that the speaker profiles are always estimated on clean audio, even when the VoxCeleb training recordings are distorted by reverberation and noise to simulate the meeting environment. As observed in Table 1, by including the VoxCeleb dataset for training, thus incorporating greater variability in the speaker acoustic characteristics, the performance becomes comparable to the baseline system for the `unseen` evaluation set. Further distortion, simulating real-world conditions, yields additional improvements in performance. Finally, incorporating temporal context (last row of Table 1) gives further substantial improvements.

Up to this point all sequences for training and evaluating have a fixed number of speakers (equal to 4). A more realistic system, used especially on meeting data, should support a variable number of participants. Thus, we now train the system with variable-length sequences, with or without context, and in Table 2 we report the results when it is still evaluated on AMI (4 speakers). As expected from the analysis and results of the previous experiments, adding context leads to consistent performance improvements. In the case when there is not temporal information, an additional observation is that, as the range of the number of speakers in the training sequences increases, there is a decreasing trend in the classification accuracy. However, comparing the columns 1 and 4 in Table 2, it seems that

[3]`https://kaldi-asr.org/models/m7`
[4]`https://github.com/deepmind/sonnet`

system	training set	acc (%)
cos	–	68.68
RMC	AMI	60.00
	VoxCeleb clean	68.15
	VoxCeleb reverb	70.25
	VoxCeleb reverb+noise	71.90
RMC & context (±1)	VoxCeleb reverb+noise	**73.86**

Table 1: Classification accuracy on AMI for the `unseen` eval set with `oraclespk` segmentation, when the system is trained on different training sets, with or without context.

adding context not only improves the performance, but also makes the system more robust, when this is trained on wider ranges of sequence lengths.

training seq length	4 spks	4-6 spks	2-9 spks	4-15 spks
w/o context	71.90	71.94	70.84	69.66
with context	73.86	73.77	72.67	73.42

Table 2: Classification accuracy (%) on AMI for the `unseen` evaluation set with `oraclespk` segmentation, when the system is trained on VoxCeleb (with added noise and reverbaration), with diff-ferent ranges of sequence lengths.

4.3. Results on internal meetings

The system trained on VoxCeleb sequences of 4-15 speakers (last column of Table 2) is also evaluated on the 9 internal meetings, where the number of participants varies. Both the `oraclevad` and the `oraclespk` segmentation scenarios are investigated. Once again, the memory-based network yields superior performance compared to the baseline (Table 3) especially when temporal information from the neighboring subsegments is added.

	cos	RMC	RMC & context
`oraclevad` – SER (%) lower is better	20.95	18.56	**11.69**
`oraclespk` – acc (%) higher is better	70.66	72.51	**79.97**

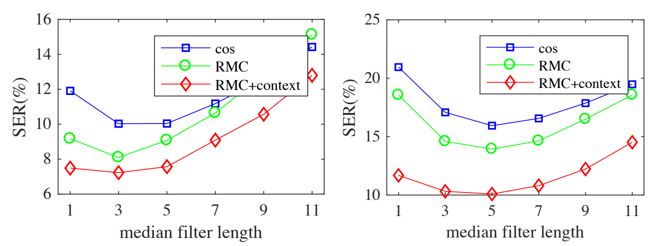
Table 3: System evaluation on the internal Microsoft meetings with different initial segmentation approaches.

The relative performance gain when adding the temporal context to the system is substantially larger, compared to the results on the AMI meetings (Table 2). This behavior can be partially explained by the inherent differences in the acoustic conditions between the two datasets. For instance, about 16% of the speaking time in the `unseen` AMI evaluation set is overlapping speech, while the corresponding percentage for the internal meetings is about 7%. Similar discrepancies were observed with regards to the frequency of speaker change points. As a consequence, it is more probable that neighboring segments of the internal meeting recordings contain information about the same speaker, thus boosting the performance when jointly provided to the network.

4.4. Smoothing at the decision level

Finally, we investigate the effect of temporal continuity on the output decisions by introducing median filtering. It is shown (Fig. 5) that a short median filter of a few taps can improve the overall performance for both datasets. Similar patterns are observed both for the baseline and for the memory-based approach. Even though the results are also improved for the RMC-based network, trained with temporal context from neighboring segments, the relative improvements are substantially smaller. We have concluded that adding temporal context to the network partially acts like a data-driven smoothing filter. Such a smoothing seems to be more effective than applying a post-processing filter (in our case median filter). This becomes apparent from the fact that the RMC-based network without added context is always outperformed by the network with added temporal information, even if no post-processing trajectory smoothing is applied to the latter (median filter length = 1 tap).

Overall, the minimum SER achieved with the RMC-based architecture including context is 7.23% for the AMI data (median filter length = 3 taps) and 10.09% for the internal meetings (median filter length = 5 taps). The cosine-based system without median filtering yields a SER equal to 11.91% and 20.95%, respectively, which translates to a relative SER reduction of 39.29% and 51.84%.



(a) AMI, `unseen`. (b) Internal meetings.

Fig. 5: SER as a function of the post-processing median filter length for the two evaluation datasets with `oraclevad` segmentation. The RMC-based network is trained on sequences of 4-15 speakers.

5. DISCUSSION AND FUTURE WORK

Herein, we have proposed a system suitable for the task of continuous speaker identification in meeting scenarios. This is based on a recurrent memory network which is capable of modeling the distance relations between observations; that is between speaker profiles and speech segments. We evaluated our approach on two corpora featuring conversational business interactions under different conditions. The proposed system yields consistent improvements in performance, when compared to a baseline system depending on the cosine distance metric. We have additionally emphasized the importance of incorporating temporal context both at the feature and the decision level, as well as the beneficial effects of using a training dataset with a large variety of speakers, and with environmental conditions matching the testing conditions, even if artificially. Following our best configuration, we achieved a SER relative reduction of 39.29% for the AMI corpus and 51.84% for the internal Microsoft meetings, when using oracle VAD information.

A potential extension of the current work will focus on better context modeling – e.g. incorporating transition probabilities between the various speakers – and on alternative memory-based architectures, which can generalize to the problem of diarization, capturing the profile information on the fly.

6. REFERENCES

[1] John HL Hansen and Taufiq Hasan, "Speaker Recognition by Machines and Humans: A Tutorial Review," *Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[2] Jonathan G Fiscus, Jerome Ajot, and John S Garofolo, "The Rich Transcription 2007 Meeting Recognition Evaluation," in *Multimodal Technologies for Perception of Humans*, pp. 373–389. Springer, 2007.

[3] Takuya Yoshioka, Dimitrios Dimitriadis, Andreas Stolcke, William Hinthorn, Zhuo Chen, Michael Zeng, and Huang Xuedong, "Meeting Transcription Using Asynchronous Distant Microphones," in *Proc. Interspeech*, 2019, pp. 2968–2972.

[4] J-F Bonastre, Perrine Delacourt, Corinne Fredouille, Teva Merlin, and Christian Wellekens, "A Speaker Tracking System Based on Speaker Turn Detection for NIST Evaluation," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2000, vol. 2, pp. II1177–II1180.

[5] Moira McGregor and John C Tang, "More to Meetings: Challenges in Using Speech-Based Technology to Support Meetings," in *Proc. Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 2208–2220.

[6] Masato Mimura and Tatsuya Kawahara, "Fast Speaker Normalization and Adaptation Based on BIC for Meeting Speech Recognition," *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2011.

[7] Giorgio Biagetti, Paolo Crippa, Laura Falaschetti, Simone Orcioni, and Claudio Turchetti, "Robust Speaker Identification in a Meeting with Short Audio Segments," in *Intelligent Decision Technologies*, pp. 465–477. Springer, 2016.

[8] Zbyněk Zajíc, Marie Kunešová, and Vlasta Radová, "Investigation of Segmentation in i-vector Based Speaker Diarization of Telephone Speech," in *International Conference on Speech and Computer*, 2016, pp. 411–418.

[9] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-End Factor Analysis for Speaker Verification," *Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[10] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN Embeddings for Speaker Recognition," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5329–5333.

[11] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector Length Normalization in Speaker Recognition Systems," in *Proc. Interspeech*, 2011.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[13] Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap, "Relational Recurrent Neural Networks," in *Proc. Advances in Neural Information Processing Systems*, 2018, pp. 7299–7310.

[14] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al., "End-to-End Memory Networks," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 2440–2448.

[15] Alex Graves, Greg Wayne, and Ivo Danihelka, "Neural Turing Machines," *arXiv preprint arXiv:1410.5401*, 2014.

[16] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al., "Hybrid Computing Using a Neural Network with Dynamic External Memory," *Nature*, vol. 538, no. 7626, pp. 471, 2016.

[17] Trang Pham, Truyen Tran, and Svetha Venkatesh, "Relational Dynamic Memory Networks," *arXiv preprint arXiv:1808.04247*, 2018.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention Is All You Need," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[19] Patrick Kenny, Douglas Reynolds, and Fabio Castaldo, "Diarization of Telephone Conversations Using Factor Analysis," *Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1059–1070, 2010.

[20] Gregory Sell and Daniel Garcia-Romero, "Diarization Resegmentation in the Factor Analysis Subspace," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4794–4798.

[21] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al., "The AMI Meeting Corpus: A pre-announcement," in *Proc. International Workshop on Machine Learning for Multimodal Interaction*, 2005, pp. 28–39.

[22] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: A Large-Scale Speaker Identification Dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[23] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep Speaker Recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[24] Takuya Yoshioka, Igor Abramovski, Cem Aksoylar, Zhuo Chen, Moshe David, Dimitrios Dimitriadis, Yifan Gong, Ilya Gurvich, Xuedong Huang, Yan Huang, Aviv Hurvitz, Li Jiang, Sharon Koubi, Eyal Krupka, Ido Leichter, Changliang Liu, Partha Parthasarathy, Alon Vinnikov, Lingfeng Wu, Xiong Xiao, Wayne Xiong, Huaming Wang, Zhenghao Wang, Jun Zhang, Yong Zhao, and Tianyan Zhou, "Advances in Online Audio-Visual Meeting Transcription," in *Proc. Workshop on Automatic Speech Recognition and Understanding*, 2019.

[25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. Workshop on Automatic Speech Recognition and Understanding*, 2011.

[26] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A System for Large-Scale Machine Learning," in *Proc. USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.