A Memory Augmented Architecture for Continuous Speaker Identification in Meetings

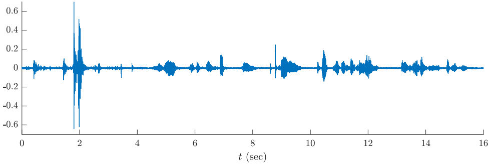
Nikolaos Flemotomos[1], Dimitrios Dimitriadis[2]

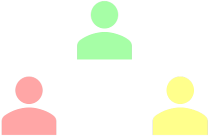
[1]Signal Analysis and Interpretation Laboratory
University of Southern California

[2]Speech and Dialog Research Group
Microsoft

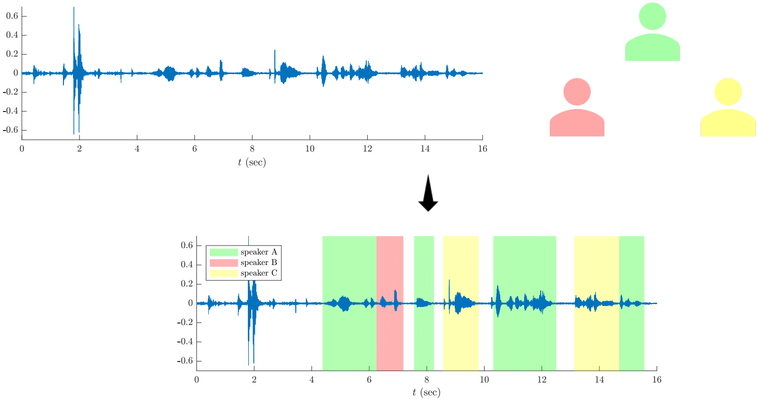
ICASSP 2020

Continuous Speaker Identification

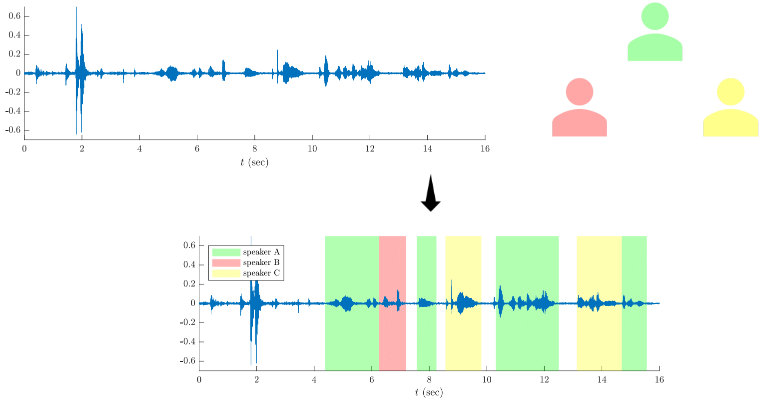




Continuous Speaker Identification



Continuous Speaker Identification

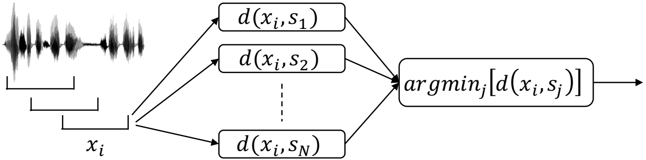


Why?

- rich transcription
- outlier detection
- speaker adaptation (ASR)
- speaker tracking

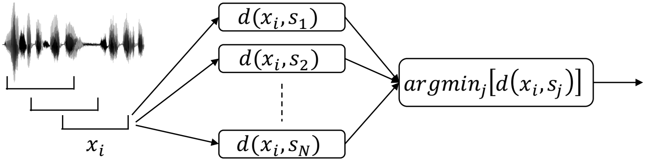
Baseline System

- extract a fixed-dimensional feature vector s_j, $j = 1, \cdots, N$ for each one of the N speakers (*speaker profiles*)
- segment the speech signal
- extract a fixed-dimensional feature vector x_i for each segment
  we 'll be using x-vectors
- pick a distance metric $d(\cdot, \cdot)$
- $\forall x_i$ select the speaker j that minimizes the distance $d(x_i, s_j)$



Baseline System

- extract a fixed-dimensional feature vector s_j, $j = 1, \cdots, N$ for each one of the N speakers (*speaker profiles*)
- segment the speech signal
- extract a fixed-dimensional feature vector x_i for each segment
- pick a distance metric $d(\cdot, \cdot)$ we 'll be using x-vectors
- $\forall x_i$ select the speaker j that minimizes the distance $d(x_i, s_j)$

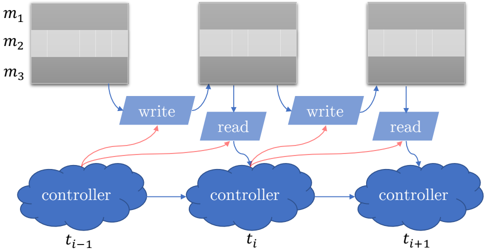


Problems

- Is the distance metric optimal?
- Is the speaker representation appropriate for the task?
- Lack of temporal information.

Memory-Augmented Neural Networks

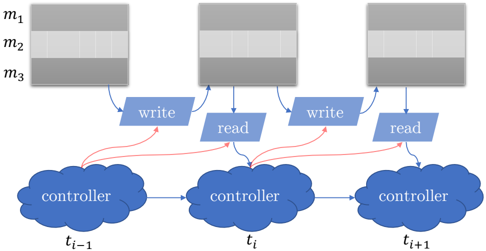
- Idea: Augment a neural architecture with a memory matrix.
- A *controller* decides how to update the memory through attention mechanisms using read and write *heads*.
- The whole system is differentiable ⇒ can learn a task-specific organization of the memory in a supervised manner through gradient descent.



Graves, Alex, *et al.* "Hybrid computing using a neural network with dynamic external memory." *Nature* (2016)
Pham, Trang, *et al.* "Relational dynamic memory networks." *arXiv preprint* (2018)
Santoro, Adam, *et al.* "Relational recurrent neural networks." *NeurIPS* (2018)

Memory-Augmented Neural Networks

- Idea: Augment a neural architecture with a memory matrix.
- A *controller* decides how to update the memory through attention mechanisms using read and write *heads*.
- The whole system is differentiable ⇒ can learn a task-specific organization of the memory in a supervised manner through gradient descent.



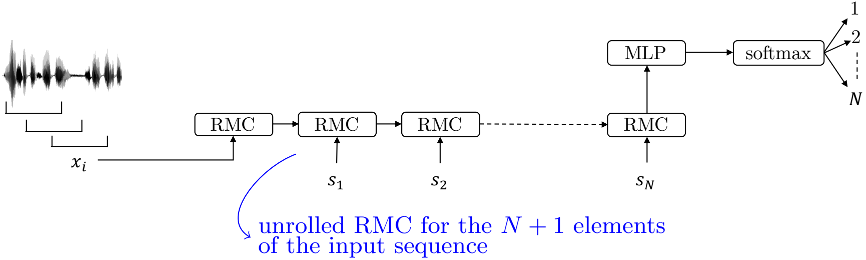
In our implementation: Relational Memory Core (RMC)

- controller [RMC] is embedded into an LSTM
- memory updates are based on a self-attention mechanism

Graves, Alex, *et al.* "Hybrid computing using a neural network with dynamic external memory." *Nature* (2016)
Pham, Trang, *et al.* "Relational dynamic memory networks." *arXiv preprint* (2018)
Santoro, Adam, *et al.* "Relational recurrent neural networks." *NeurIPS* (2018)

Proposed Architecture

- $\forall x_i$ create the sequence $\{x_i, s_1, s_2, \cdots, s_N\}$
- pass the sequence through an RMC-based network and get the label $l_i \in \{1, 2, \cdots, N\}$ corresponding to x_i; this is the one that maximizes the probability $\mathbb{P}\left[l_i = j | x_i, \mathbf{s} = \{s_j\}_{j=1}^{N}\right]$



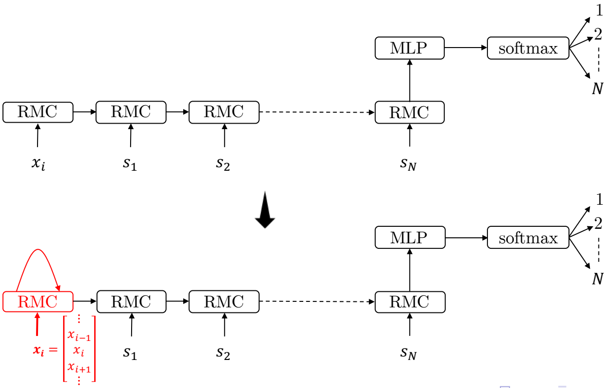
- Each element of the sequence is projected onto the "memory space".
- The RMC learns some *local* distance metric, sorts the distances and finds the s_j that minimizes the distance from x_i.

Incorporating Temporal Information

Segment length: a trade-off decision

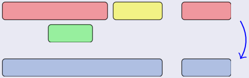
- short segments ⇒ unstable speaker representation
- long segments ⇒ multiple speakers in a single segment

<u>Solution</u>: reasonably short segments while keeping information from neigboring ones



Data Pre-processing

Segmentation



- `oraclespk:` (metric: classification accuracy)
- `oraclevad:` (metric: Speaker Error Rate)

Subsegmentation

Each available segment is further uniformly subsegmented into 1.5*sec* windows (best trade-off for baseline system).

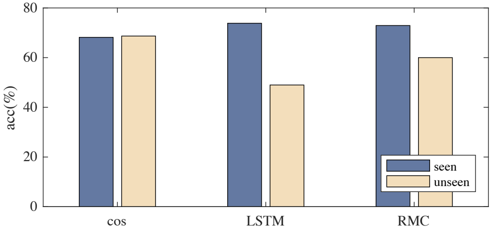
Profile estimation

An *x-vector* is extracted ∀ available segment in the "oracle speakers" scenario and a mean *x-vector* per speaker is calculated.

Snyder, David, *et al.* "X-vectors: Robust DNN embeddings for speaker recognition." *IEEE International Conference on Acoustics, Speech and Signal Processing* (2018)

Results on AMI

Simulated business meetings: 4 speakers per meeting

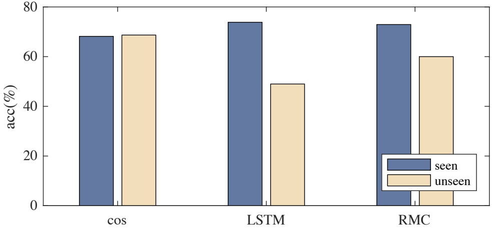


`oraclespk` segmentation, trained on AMI

- RMC captures distance information better than LSTM
- both networks fail to beat the baseline on unseen speakers (limited training speakers? $\Rightarrow$ switch to VoxCeleb for training)

Results on AMI

Simulated business meetings: 4 speakers per meeting



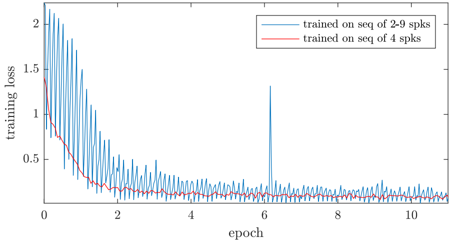
`oraclespk` segmentation, trained on AMI

- RMC captures distance information better than LSTM
- both networks fail to beat the baseline on unseen speakers (limited training speakers? $\Rightarrow$ switch to VoxCeleb for training)

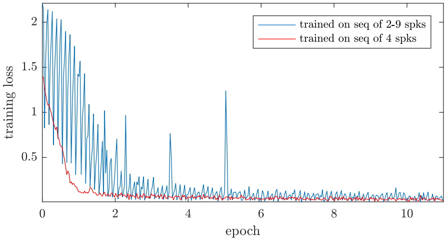
system	training set	acc (%)
cos	–	68.68
RMC	AMI	60.00
	VoxCeleb clean	68.15
	VoxCeleb reverb	70.25
	VoxCeleb reverb+noise	71.90
RMC & context (±1)	VoxCeleb reverb+noise	**73.86**

`oraclespk` segmentation, evaluation on `unseen` AMI

Training with variable-length sequences



VoxCeleb reverb+noise, no context

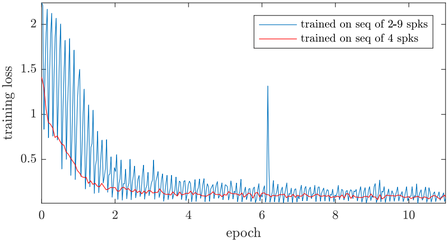


VoxCeleb reverb+noise, with context

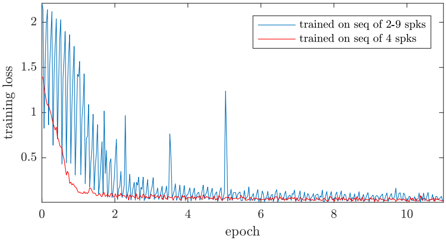
training seq length	4 spks	4-6 spks	2-9 spks	4-15 spks
w/o context	71.90	71.94	70.84	69.66
with context	73.86	73.77	72.67	73.42

System accuracy on `unseen` AMI set when trained with different ranges of sequence lengths.
(always testing on sequences of 4 speakers)

Training with variable-length sequences



VoxCeleb reverb+noise, no context



VoxCeleb reverb+noise, with context

Adding context improves robustness when training with variable-length sequences.

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

System accuracy on **unseen** AMI set when trained with different ranges of sequence lengths. (always testing on sequences of 4 speakers)

Results on Internal Meetings

9 real-world business meetings (4.6h): 4-15 speakers per meeting

	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**

System evaluation with different segmentation approaches on internal meetings.

Results on Internal Meetings

9 real-world business meetings (4.6h): 4-15 speakers per meeting

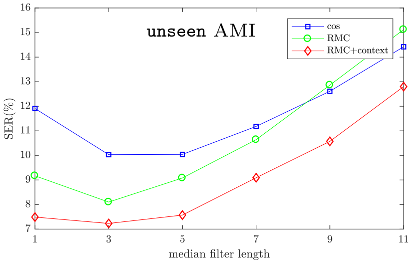
	cos	RMC	RMC & context
`oraclevad` – SER (%) lower is better	20.95	18.56	**11.69**
`oraclespk` – acc (%) higher is better	70.66	72.51	**79.97**

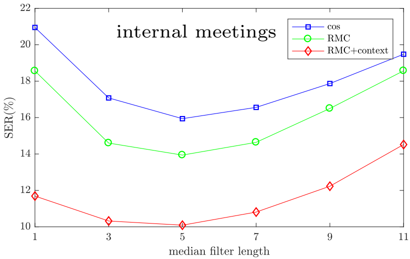
System evaluation with different segmentation approaches on internal meetings.

Adding temporal context substantially improves the performance.
Can we do even better by incorporating temporal context at the decision level?

Smoothing at the Decision Level

<u>Assumption</u>: highly improbable that isolated short segments correspond to some speaker in the middle of an utterance assigned to another speaker
$\Rightarrow$ Smooth the trajectory of the predicted speaker labels via median filtering.

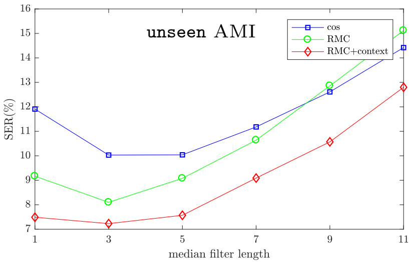


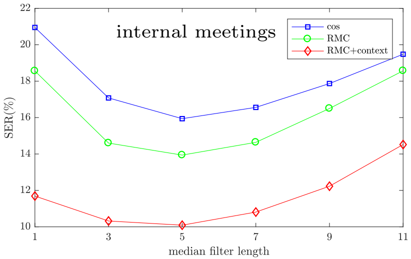


System evaluation for the two datasets using different lengths of median filter for post-processing with the `oraclevad` segmentation. The RMC-based system is trained on sequences of 4-15 speakers.

Smoothing at the Decision Level

Assumption: highly improbable that isolated short segments correspond to some speaker in the middle of an utterance assigned to another speaker
⇒ Smooth the trajectory of the predicted speaker labels via median filtering.





System evaluation for the two datasets using different lengths of median filter for post-processing with the `oraclevad` segmentation. The RMC-based system is trained on sequences of 4-15 speakers.

- A short median filter improves the performance for both datasets.
- Adding temporal context to the network partially acts like a data-driven smoothing filter.

Conclusion

- Introduced a novel architecture for continuous speaker identification.
- Showed the importance of incorporating temporal context information both at the feature and the decision level.
- Demonstrated a SER relative reduction of 39.29% for the AMI corpus and 51.84% for the internal Microsoft meetings, compared to the baseline when using oracle VAD information.

Appendix - Relational Recurrent MANNs: Controller

Let a memory matrix M with memory slots $m_1, m_2, \cdots$

- Updates are based on a self-attention mechanism:
  - assume no new observations

$$\tilde{M} = softmax\left(\frac{\overset{\text{queries}}{(MW^q)}\overset{\text{keys}}{(MW^k)^T}}{\underset{\text{key dimensionality}}{\sqrt{d_k}}}\right)\overset{\text{values}}{MW^v}$$

  - incorporate a new observation x

$$\tilde{M} = softmax\left(\frac{(MW^q)([M;x]W^k)^T}{\sqrt{d_k}}\right)[M;x]W^v$$

Note that memory matrix dimensions do not change.

- Each memory attends to all the other memories to be updated $\Rightarrow$ cross-memory relations are encoded

Santoro, Adam, *et al.* "Relational recurrent neural networks." *Advances in Neural Information Processing Systems* (2018)
Vaswani, Ashish, *et al.* "Attention is all you need." *Advances in Neural Information Processing Systems* (2017)

Each memory m_i is embedded into an LSTM. The resulting controller is called *Relational Memory Core (RMC)*.

$$
\begin{aligned}
s_{i,t} &= (h_{i,t-1}, m_{i,t-1}) \\
f_{i,t} &= W^f x_t + U^f h_{i,t-1} + b^f \\
i_{i,t} &= W^i x_t + U^i h_{i,t-1} + b^i \\
o_{i,t} &= W^o x_t + U^o h_{i,t-1} + b^o \\
m_{i,t} &= \sigma\left(f_{i,t} + \tilde{b}^f\right) * m_{i,t-1} + \sigma\left(i_{i,t}\right) * g(\tilde{m}_{i,t}) \\
h_{i,t} &= \sigma\left(o_{i,t}\right) * \tanh\left(m_{i,t}\right) \\
s_{i,t+1} &= (h_{i,t}, m_{i,t})
\end{aligned}
$$

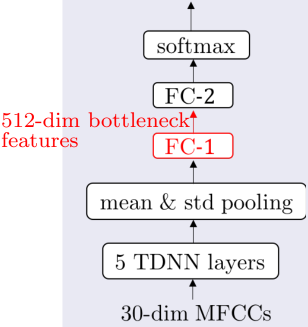
denotes element-wise multiplication

- $g(\cdot)$ is some non-linear function. In practice, modelled by a 2-layer fully connected MLP.
- all the weights and biases are shared across the memories m_i $\forall i$ more memory slots $\not\Rightarrow$ more trainable parameters

Appendix - Speaker Embeddings: x-vectors

golden standard for tasks including → speaker recognition
→ diarization
→ language identification

Architecture



Training procedure

- speaker recognition on VoxCeleb 1, 2
- $2 - 4sec$ long speech segments
- $16kHz$ audio

Feature normalization

- LDA projection on a d-dimesnional space ($d = 200$)
- mean- and length-normalization to $l = \sqrt{d}$

Snyder, David, *et al.* "X-vectors: Robust DNN embeddings for speaker recognition." *IEEE International Conference on Acoustics, Speech and Signal Processing* (2018)

Appendix - Datasets

AMI Meeting Corpus

- 4 speakers per meeting
- 31 scenarios with 4 business meetings each (124 speakers):
  - 62 meetings for training (35.5h) $\rightarrow$ distant mic
  - 31 for evaluation [`seen`] (17.1h) $\rightarrow$ distant mic
  - 31 for profile extraction (8.0h) $\rightarrow$ close-talk mic
- 6 additional meetings with **`unseen`** speakers for evaluation (4.1h) plus 6 for profile estimation (3.8h)
- subsegments every 0.75sec both for training and evaluation (160K subsegments for training)

Appendix - Datasets

VoxCeleb

- 6490 speakers with > 6 utterances
- select 3 utts per speaker for profile estimation (totally $383.3h$)
- subsegment the rest every $10sec$ (840.6K training subsegments)
- randomly create sequences of subsegments and speaker profiles for training

Internal Microsoft Meetings

- 9 business meetings ($4.6h$)
- $4 - 15$ speakers per meeting
- subsegments every $0.75sec$
- speakers already enrolled

Appendix - Experimental Setup

Training details

- Handling speaker ordering: Randomly permute the speaker profiles in every training sequence.
- When training with variable-length sequences, fix the length in each mini-batch.

Network parametrization

- #memories = max #speakers + 1
- memory size = 2048
- MLP: 4 FC layers, 256 neurons

Evaluation metrics

- `oraclespk`: classification accuracy
- `oraclevad`: Speaker Error Rate (collar=0.25*sec*)

Tools

- *Kaldi* for feature and embedding extraction
- *Tensorflow* with *Sonnet* library to build the network
- *NIST md-eval.pl* for SER estimation