Interactive Speaker Recognition

Vyacheslav Golovin
Aleksandr Samarin (supervisor)

June 22, 2023

Huawei CBG AI and HSE University

Introduction

Speaker recognition (SR) is the task of recognizing a person using speech audio. There are 2 types of SR tasks:

1. **Identification** — select a speaker from a group.
2. **Verification** — decide whether the selected person is the actual speaker.

In deep learning setting both these tasks boil down to comparing speaker and speech vector representations, which we will refer to as speaker and word embeddings.

Motivation

Application: Speaker verification system which prompts the speaker to say some word or phrase in order to verify their identity.

Requirements:

- few (short) prompts,
- high accuracy,
- diverse prompts to avoid spoofing.

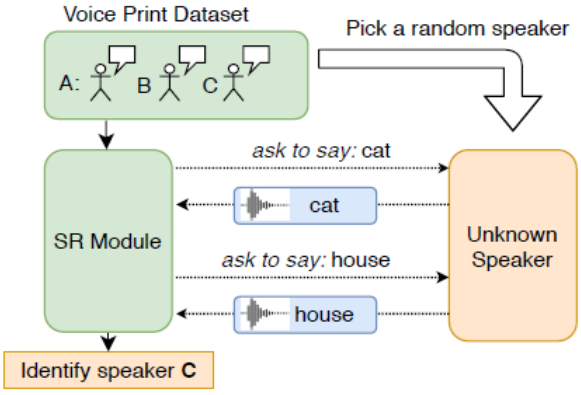
Proposal: use Interactive Speaker Recognition approach introduced in

> *A Machine of Few Words — Interactive Speaker Recognition with Reinforcement Learning*, Mathieu Seurin et al., INTERSPEECH 2020, `arXiv:2008.03127v1`.

Interactive Speaker Recognition

Input data: TIMIT dataset (630 speakers, 20 shared words).

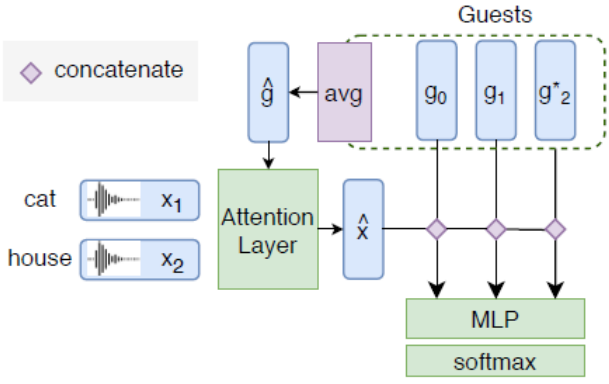
Audio is processed with MFCC and x-vector neural network.



Important notes:

1. Only identification problem is considered.
2. The set of words is fixed.
3. SR Module uses 2 separate neural networks: `Enquirer` and `Guesser`.

Guesser



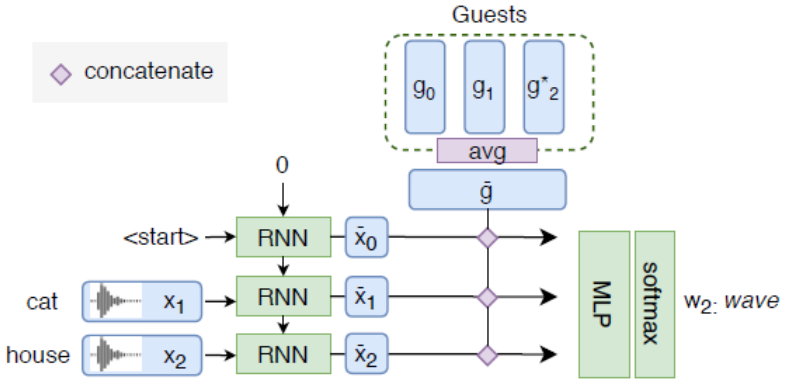
Inputs:

- speaker embeddings g_k
- word embeddings x_t

Outputs:

- probability distribution over speakers $P(k = \text{target})$

Enquirer



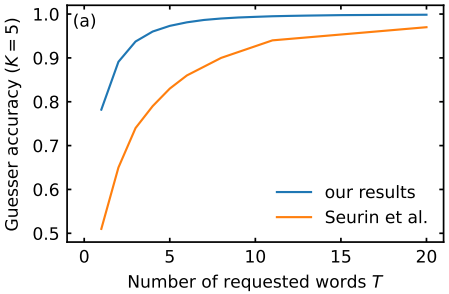
Inputs:

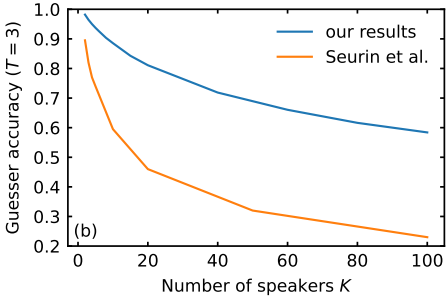
- mean speaker embedding $\hat{g}$
- uttered word embeddings x_t

Outputs:

- probability distribution over vocabulary $P(v = \text{requested word})$

Training Guesser

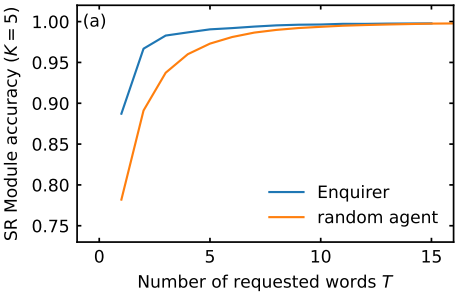


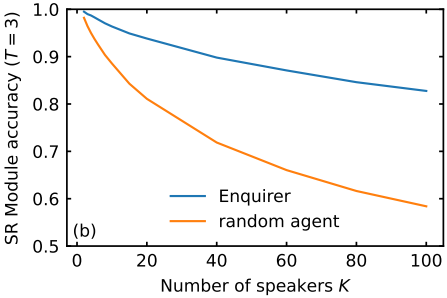


Trained with supervised learning: speakers and words are sampled randomly, model is trained to minimize cross-entropy.

The difference is likely due to the increase in embedding size — 512 vs 128 in the original paper.

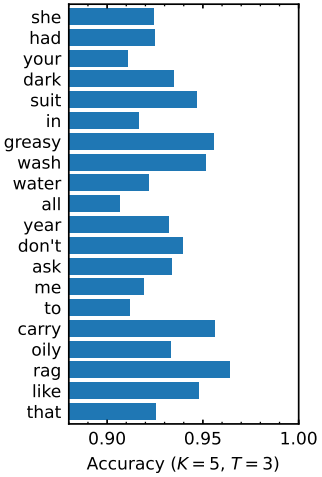
Training Enquirer



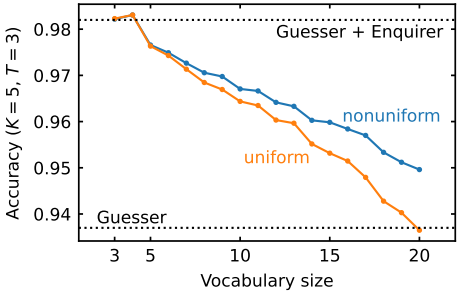


Trained with reinforcement leaning, namely **PPO** algorithm. `Enquirer` selects words for trained `Guesser`, reward of 1.0 is received if `Guesser` correctly chooses target speaker.

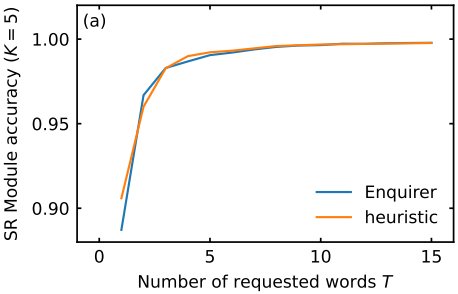
Heuristic agent

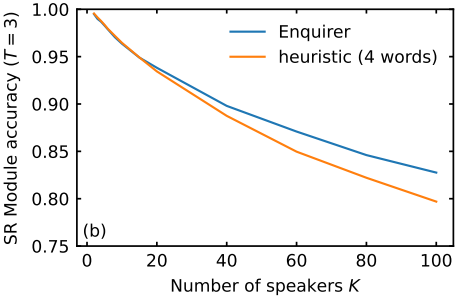


1. Compute word accuracies on validation subset.
2. Sample only from a subset of words with the highest accuracies.



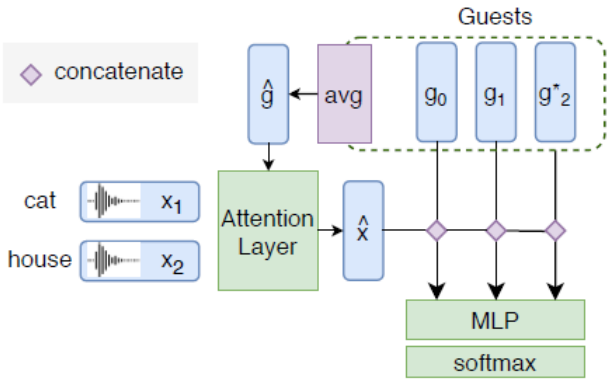
`Enquirer` vs heuristic agent





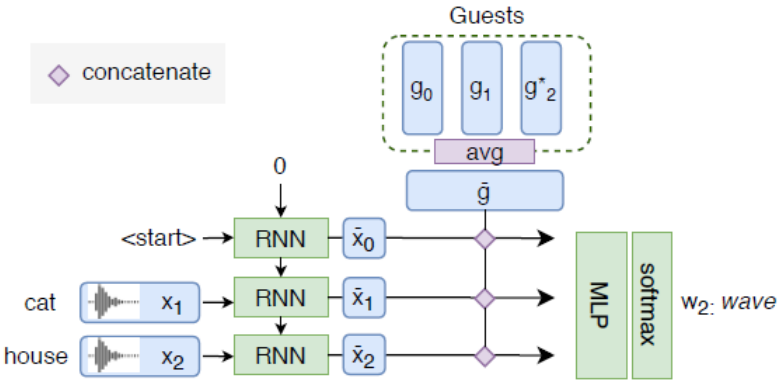
The two agents' policies are very similar, i.e., `Enquirer` mostly selects the same words irrespective of guest composition, more diverse policies typically perform worse.

From identification to verification



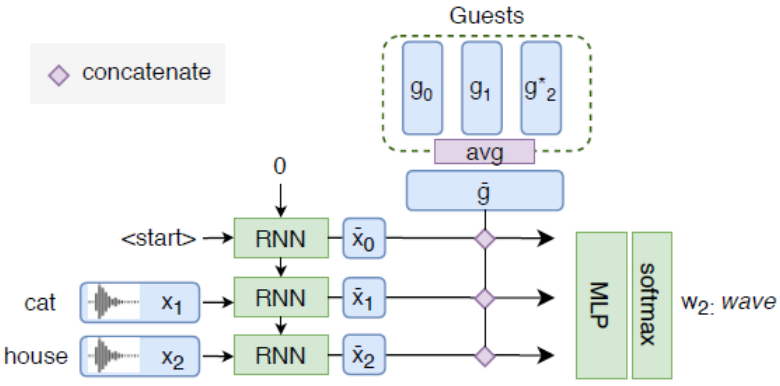
- `Guesser`: replace softmax with sigmoid

From identification to verification



- `Guesser`: replace softmax with sigmoid
- `Enquirer`: no changes required

From identification to verification



- `Guesser`: replace softmax with sigmoid
- `Enquirer`: no changes required

Agent	Accuracy
random	0.895
Enquirer	0.933
heuristic	0.917

Selecting the training mode

Agent	Training mode	Accuracy
random		0.895
`Enquirer`	$T = 3$	0.933
heuristic		0.917
random		0.913
`Enquirer`	$T = 2$	0.947
heuristic		0.945

Verification accuracy, $T = 3$ requested words

Selecting the training mode

Agent	Training mode	Accuracy
random	$K = 5$	0.937
`Enquirer`	$T = 3$	0.982
heuristic		0.984
random	$K = 20$	0.951
`Enquirer`	$T = 2$	0.989
heuristic		0.988

Identification accuracy, $K = 5$ speakers, $T = 3$ requested words

CodebookEnquirer

Problem: Current `Enquirer` implementation uses a fixed set of words. Adding a new one would require retraining or fine-tuning.

Proposed solution:

- Change last layers of `Enquirer`, so that it returns requested word embedding instead of probability distribution.
- Construct `Codebook` — a tensor of word embeddings averaged over training set speakers.
- Select next requested word based on distances between output and codebook vectors.

Result: Little to none accuracy penalty, even if different sets of words are used during training and testing.

Other experiments

1. Background noise
   - 6 noise samples from MUSAN (rain, car, crowd, typing, hum, white) added to word audio with 3 dB SNR. Noise type is consistent throughout the ISR game episode.
   - No significant change in results, only a small drop in accuracy for every word selection algorithm. Notably, no noise adaptation of `Enquirer`.
2. Different embeddings

Embeddings	Identification		Verification	
	random	Enquirer	random	Enquirer
x-vector	0.75	0.91	0.89	0.94
CPC	0.95	0.99	0.95	0.97

Speaker recognition accuracy, $K = 20$ speakers, $T = 2$ requested words

Conclusions

- The interactive speaker recognition method works — selecting requested words with a neural agent allows for a significant increase in speaker recognition accuracy.
- The approach is rather flexible — we were able to easily transition from identification to verification and perform other modifications that improve model usability and performance.
- Performance is very similar to a simple baseline — not clear if the use of complicated reinforcement learning algorithms is actually justified.