A Comprehensive Evaluation Framework for Speaker Anonymization Systems

Carlos Franzreb (DFKI), Tim Polzehl (DFKI), Sebastian Möller (TU Berlin)
carlos.franzreb@dfki.de

1. Evaluation Framework

- **100% Python**: easy to understand and modify, perfect for PyTorch models
- **Configurable**: easily change which components and datasets to use
- **Modular**: add your own components
- **Fast**: Evaluate the utility and privacy of your model in less than 4 hours.

2. Evaluation Datasets

We have used four datasets for our experiments, chosen for their diverse recording environments and speaker characteristics:

Dataset	Type of speech	Recording quality	Speaker diversity	Num. of speakers	Num. of hours
LibriSpeech	Read	High	Low	40	5.3
EdAcc	Spontaneous	Low	High	60	10.8
Common Voice	Read	Low	High	654	3.3
RAVDESS	Emotional	High	Low	24	1.5

3. Anonymization Pipelines

Pipelines may include four components:

1. **Feature extraction**: extract information of any kind from the waveforms: speaker embeddings, text, etc.
2. **Feature processing**: anonymize the extracted features.
3. **Feature fusion**: Merge the different features, e.g. by concatenation.
4. **Synthesis**: transform the fused features into waveforms.

Here are the two pipelines we have evaluated with our framework. Neither requires a fusion component. Both are trained on 20 targets: we randomly select a target for each source speaker:

Module	StarGANv2-VC	SoftVC	STT-TTS (NeMo)
Feat. extraction	Spectrogram	HuBERT	Whisper-small
Feat. processing	StarGANv2-VC	Acoustic model	FastPitch
Synthesis	Parallel WaveGAN	HiFi-GAN	HiFi-GAN
Num. of targets	20	1	20

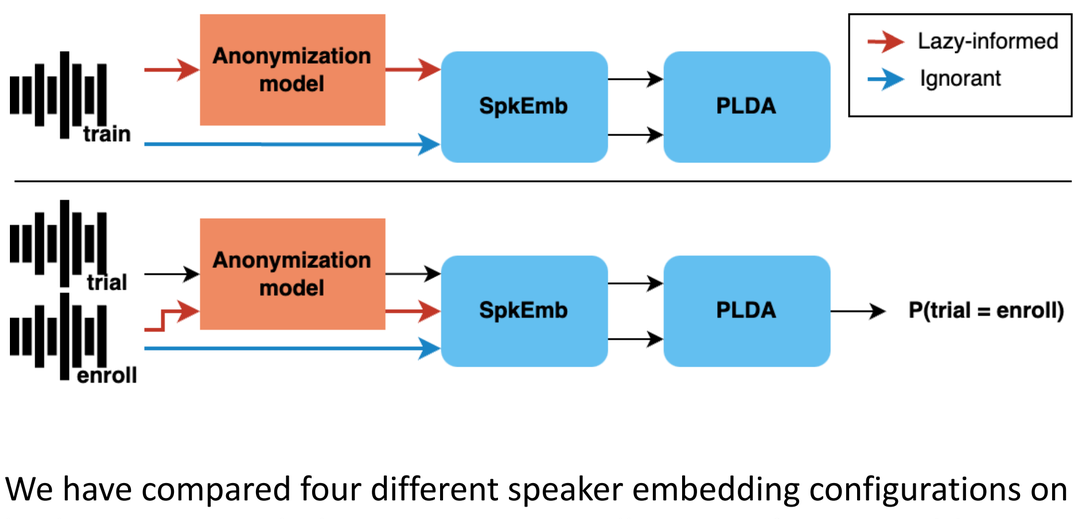
4. Utility evaluation

Different applications have different requirements, and they should all be considered in the utility evaluation (ideally). Preferably, **evaluation components should not be trained**, as this is time-consuming. Until now, we have added the following:

1. **ASR:** measures the intelligibility of the anonymized speech with the small and large Whisper models, as the size influences the results.
2. **Emotion Preservation:** with a wav2vec2.0 fine-tuned for SER, compares the emotion embeddings of the original and the anonymized speech.
3. **Naturalness**: measured by the MOS scores from the NISQA-TTS model, which quantify the clarity and naturalness of the anonymized speech.
4. **Performance:** inference speed on CPU & GPU, as well as throughput on GPU, which is important for training.

5. Privacy Evaluation

The framework includes the **ASV evaluation from the VoicePrivacy Challenge.** It includes two scenarios, which differ in whether the anonymization pipeline is available to the attacker or not:



We have compared four different speaker embedding configurations on both scenarios with the StarGANv2-VC. All use **speaker recognition models from SpeechBrain**, and are **trained on subsets of LibriSpeech and EdAcc**. Surprisingly, x-vectors work better than ECAPA-TDNN embeddings. Here are the EERs (from 0 to 100, lower is better) for the ignorant and lazy-informed scenarios, respectively:

Dataset	ECAPA	ECAPA+LDA	Xvect+LDA	All
LibriSpeech	**32.2** / 36.5	34.3 / 36.7	32.5 / 26.8	38.8 / **26.2**
EdAcc	39.9 / 41.4	39.8 / 41.2	**37.9** / **28.4**	40 / 30.1
CV	34.2 / 33.3	34.2 / 33.3	27.2 / **21.7**	33.5 / 22
RAVDESS	**43.4** / 43.8	46.8 / 42.8	43.7 / 35.8	46.6 / **34.5**
Avg.	37.4 / 38.8	39.4 / 38.5	**35.3** / **28.2**	39.7 / 28.2

6. Results

Here are the results for the StarGANv2-VC and the TTS pipeline, compared with a baseline where speech is not anonymized.

LibriSpeech

Component	Baseline	StarGANv2-VC	SoftVC	STT-TTS
Ignorant ASV	0.96	32.46	43.72	**48.63**
Lazy-informed ASV		26.77	19.17	**43.16**
Small ASR	0.06	0.14	**0.07**	0.08
Large ASR	0.07	0.09	**0.06**	0.08
SER		**1.0**	**1.0**	0.99
Naturalness	3.88	3.12	**3.95**	3.7
CPU inference		8.71	8.06	**1.74**
GPU inference		0.16	0.67	**0.09**

EdAcc

Component	Baseline	StarGANv2-VC	SoftVC	STT-TTS
Ignorant ASV	4.69	37.89	44.49	**48.43**
Lazy-informed ASV		28.4	30.31	**46.75**
Small ASR	0.32	0.75	0.52	**0.34**
Large ASR	0.29	0.6	0.47	**0.34**
SER		0.99	**1.0**	0.99
Naturalness	2.63	2.95	**3.6**	3.58

Take-aways

1. **StarGAN has better EERs in the lazy-informed scenario than SoftVC** due to its multiple targets; but its utility is very low for noisy speech.
2. **StarGAN is faster than SoftVC on GPU**, and slower on CPU, due to its convolutional architecture against SoftVC's attention.
3. **SoftVC offers great utility** in both and ignorant EER.
4. **STT-TTS offers the most privacy and is the fastest**, but its emotion preservation should be lowest (SER must be improved).