Voice Anonymization by Multimodal Adaptive Noise

CSC2518 project report
Zining Zhu

Abstract

Speaker anonymization has gained increasing popularity recently. This project aims at evaluating three methods of voice anonymization based on adding noises. This project quantitatively evaluates the effects of these methods. In addition, the equal error rate (in speaker verification) and the word error rate (in speech recognition) are computed to evaluate the efficacy of the anonymization methods. All analysis codes are open-sourced at `https://github.com/ZiningZhu/CSC2518`.

1 Introduction

Speaker identification is a prevalent task in speech processing. The speaker identification systems are widely applied. In a telephone-based customer service system, speaker identification can help build the profiles of customers, enabling the potential for a user-specific customer service strategy. Speaker identification can also be helpful for security. A speech for as short as 30 seconds can enable high-accuracy speaker identification systems [1]. Phone banking services can use speaker identification as an additional "fingerprint". If the user loses the password by accident, the password could be retrieved through this "voice-based fingerprint".

Recently, as the attention toward privacy is raised, it is increasingly desirable to protect the speaker's identity. For example, a smart assistant located indoors does not need to know who the speaker is. It is sufficient to automatically recognize the speech content and respond accordingly[1]. In another example, a customer might want to avoid the telemarketing companies to make customized phone sales based on the features extracted from their speeches. These scenarios motivate the protection against identification, i.e., voice anonymization.

Researchers have proposed a wide collection of voice anonymization techniques. This project is specifically interested in an avenue of approach: noise injection. I intend to empirically evaluate the efficacy of these approaches in removing the voice-based fingerprints. In addition, I intend to assess the impacts of these anonymization techniques on ASR systems: While it is desirable to anonymize the voice, it is undesirable to erase the useful information. The Equal Error Rate (EER) is used to evaluate the effect of anonymization, and the Word Error Rate (WER) of speech recognition is used to evaluate the amount of remaining portion of information.

This project applies some voice anonymization techniques to a subset of the LibriSpeech data, which is used by the VoicePrivacy initiative evaluation plan [2]. The techniques are based on multimodal Gaussian noises applied to the utterances of the speakers. As baselines, unimodal Gaussian noises and non-adaptive noises are also tested.

[1] One might argue that an assistant located in a smartphone should respond to only the owner of the smartphone. However, for a smart assistant located indoors, only the commands of those people physically in the household can reach that smart assistant.

2 Related Works

There are many voice anonymization attempts using signal processing techniques. Voice Transformation [3] aimed at changing some attributes of voice so that it sounds like a target speaker. They tested approaches including changing the fundamental frequency $F0$, changing the duration, and "transterpolating" (i.e., interpolate to beyond $(0, 1)$ ranges) the Mel-cepstral coefficients (MCEP). VoiceMask [1] applied frequency-warping functions to change the characteristics of the speeches. Vocal tract length normalization (VTLN) [4, 5] was originally proposed to improve the automatic speech recognization performances, especially in the case of small training data. However, since VTLN aims to adjust for some characteristics of individual speakers, VTLN effectively anonymizes the speeches.

[6] considered an approach based on adjusting the McAdams coefficients. In musical synthesis, a popular approach generates the timbre of music through a weighted addition of (co)sinusoidal oscillations. Let $y(t)$ be the music signal, then this synthesis approach can be written as $y(t) = \Sigma_k r_k(t)\cos(2\pi(kf_0)^\alpha)t + \phi_k$, where $r_k(t)$ is the amplitude, f_0 is the fundamental frequency, ϕ_k is the phase shifts, and α is referred to as the McAdams coefficients [7], which adjusts the frequency of each harmonic. Changing the McAdams coefficients adjusts the timbre of the music. Similarly, this approach adjusts the styles of the speech during a "linear predictive coding – resynthesis" pipeline. As a side note, a relevant approach, "automatic speech recognition (ASR) – synthesis" are used in voice transformation [8]. However, one may argue that ASR preserves some aspects of speech (e.g., the choice of wording) that can reveal the identity of the speakers. These aspects may be carried over in the synthesized speech, which does not really result in true anonymization.

More recent approaches tackle the voice anonymization problem through the lens of "style". [9] set up neural networks to learn x-vector that represents the styles. For each speaker, swapping out the x-vector changes the styles into those of a "pseudo-speaker". Up till now, x-vector is considered the state-of-the-art in competitions, including the VoicePrivacy Challenge. [10] used CycleGAN to convert the styles of speeches between male and female speakers. Subsequently, they perturbed the pitches and tempo to arrive at anonymized speeches.

3 Methods

3.1 Problem setting

Let $y^{(s)}(t)$ represent the speech signals from speaker s. An utterance of the speech signals is stored as an array $\{y(1), ..., y(T)\}^{(s)}$. Let $x^{(s)}(t)$ represent the text that correspond to the speech signals $y^{(s)}(t)$. We consider an avenue of anonymization: noise injection. This changes the speech signals into $y^{(s)}(t) + \epsilon(t)$, where $\epsilon(t)$ is the adaptive noise. The rest of this project report adopts the notations described here.

3.2 Voice anonymization

Different anonymization approaches is referred to by the natures of the adaptive noise $\epsilon(t)$.

- *Uniform noise* is a simple noise drawn from Gaussian distribution with fixed scale: $\epsilon(t) \sim \mathcal{N}(0, \sigma^2)$
- *Adaptive noise* is a Gaussian noise with the variance proportional to the amplitude of the utterance: $\epsilon(t) \sim \mathcal{N}(0, \gamma\sigma^2)$, where $\sigma^2 = \text{Var}(y^{(s)}(t))$, and γ is a hyperparameter

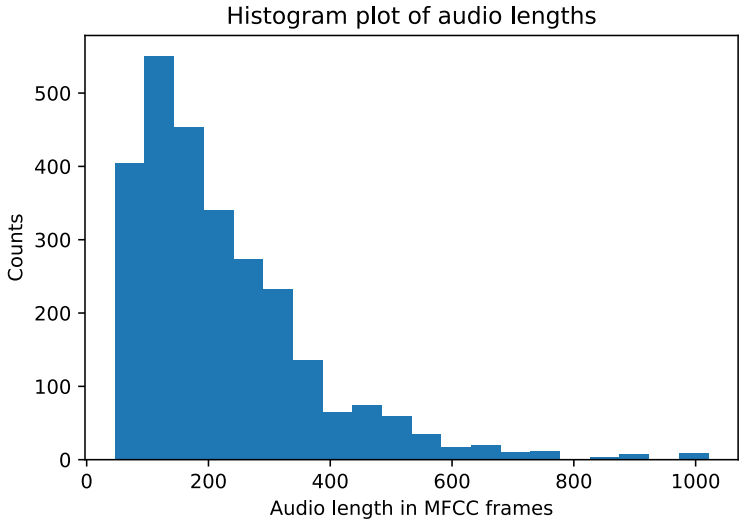

Figure 1: Audio lengths by frames.

that adapts the scales of the noise. In this setting, an utterance with more salient speaker identification features would be added by a noise with a larger scale.

- *Multimodal adaptive noise* follows the intuition that the speech signals have more than one modes. This approach first fits a Gaussian Mixture model with m components to each utterance: $\hat{y}^{(s)} \sim \Sigma_m \omega_m \Gamma(\mu_m, \sigma_m^2)$, where ω_m, μ_m and σ_m^2 are the model parameters, and $\Gamma(\mu_m, \sigma_m^2)$ denotes a Gaussian model. This project uses an off-the-shelf toolkit, scikit-learn [11] to solve for the parameters of the Gaussian Mixture model. Then, this approach adds m adaptive noises (weighted by ω_m) to anonymize the speech signal: $\epsilon(t) \sim \Sigma_m \omega_m \mathcal{N}(\mu_m, \gamma\sigma_m^2)$.

3.3 Speaker identification

This project assumes the following attack model in speaker identification. An adversary $\mathcal{A}$ has access to all utterances $y^{(s)}(t)$, but the adversary only has partial knowledge of the speaker identity s for some frames.

The speaker identification is done in the following procedure. First, MFCCs are computed using the default configurations of the *librosa* toolkit[2]. Each utterance is converted into varying frames (mean 224.76, std 146.71). Figure 1 shows a histogram of the length distributions. We take the first $N = 50$ frames, and evenly split into a "train set" vs. a "dev set" with stratified sampling. Those audios that are shorter than 50 frames (there are 4, accounting for 0.15%) are skipped.

The adversary learns to predict the speaker from the MFCC features of the frames of speech. A `MLPClassifier` from scikit-learn is used, together with the default configuration in training. To evaluate the performance of classification, I follow the VoicePrivacy recipe and compute the equal error rate (EER), which is the false positive rate P_{fp} when the prediction threshold θ is adjusted so that P_{fp} equals the false negative rate P_{fn}:

$$EER = P_{fp}(\theta) = P_{fn}(\theta)$$

[2] `https://librosa.org`

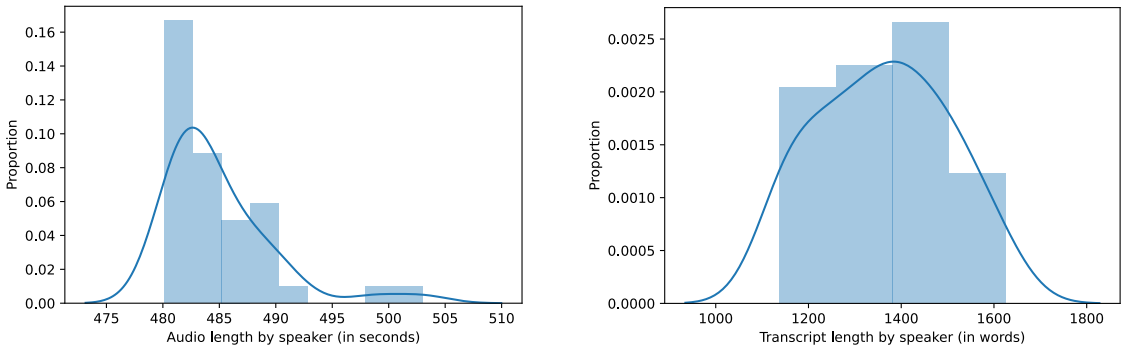

Figure 2: Distribution plots of audio length (left) and transcript length (right) by speaker.

3.4 ASR evaluation

To evaluate the informativeness of the anonymized audios, automatic speech recognization (ASR) is used. Here I use a pre-trained model, `asr-crdnn-rnnlm-librispeech`, implemented by SpeechBrain [12]. Following the convention, word-error-rate (WER) is used to quantify the quality of the transcription.

4 Data

The *dev-clean-100* subset of LibriSpeech is used. There are 2,703 utterances from 40 speakers, totaling 5.39 hours of speech. On average, each utterance is 7.18 seconds in length.

For each speaker, there are 484.90 seconds (std=4.78) of audio, with 1360 words (std=139) of transcript. Figure 2 shows the distribution of the audio and transcript lengths of the speakers.

5 Experiments

5.1 A case study for anonymized spectrograms

Figure 3 illustrates the effects of the anonymization mechanisms through spectrograms of an utterance (Speaker 84, session 121123, id 0007). The transcript is "WHAT DO YOU MEAN SIR", and the hyperparameters for the added noise follow the default configurations ($\sigma = 0.01$ for uniform noise, $\gamma = 0.1$ for both adaptive and multimodal noises, and $n = 3$ for the multimodal noise).

The first observation is perhaps the reduction of variation between the pattern and the background. Adding noises to the audio makes the spectrogram patterns less salient. Especially, the high-frequency bands in the middle of the plot become barely visible in the adaptive and multimodal noise spectrograms.

On the other hand, the low-frequency patterns pertaining to the voiced words are still visible, indicating that the contents that are recognizable are largely preserved during the anonymization.

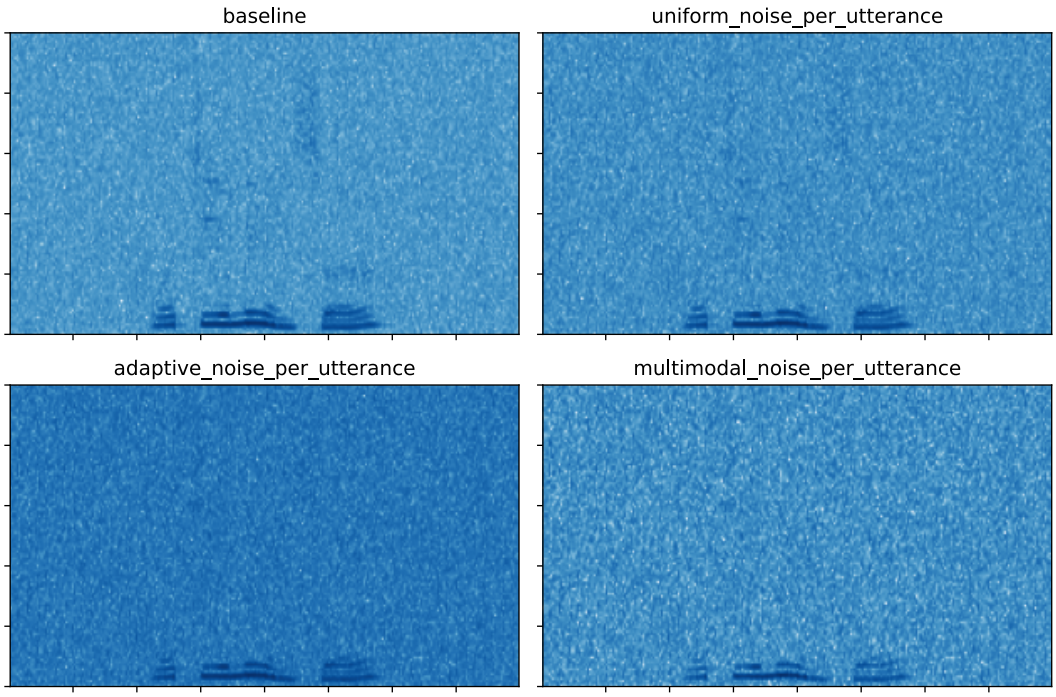

Figure 3: Spectrograms of an example utterance ("what do you mean sir"), processed by different mechanisms.

5.2 A case study for anonymized transcripts

Table 1 shows some examples of ASR transcription errors of one sample. The ASR transcription errors mostly focus on some "hard-to-transcribe" words, which are highlighted by colors.

Some transcription errors are relevant to the context. For example, used to is frequently transcribed into mister – a term occurring twice in the surrounding context. I do not observe evidence that this type of transcription error is affected by choice of anonymization mechanisms.

Another type of transcription error involves frequent tokens. For example, his is sometimes transcribed into the 's or es markers at the end of the preceding nouns. I hypothesize that the language model in the ASR system plays an important role in making these adjustments.

As a side note, both in the adaptive noise and the multimodal noise settings, a larger noise scale is accompanied by more transcription errors – for the settings with larger noises, a part of the sentence ("and Mister John Collier gives his sitter a cheerful slap on the back") is not recognized at all. Apparently, the ASR system stops translation when the probabilities for words appear relatively small compared to that of an "end-of-sentence" token.

5.3 Comparison of results

To account for the difference in random seeds in generating noises, for each configuration of experiments, three runs (with random seeds 1234, 42, 0, respectively) are proceeded. The EER and WER scores of the three runs are reported in Table 2.

The EER values in most settings appear larger than the baseline, indicating that the injected noise indeed protects the identity of the speaker to varying extents. However, in an ideal scenario where it is impossible to identify the speakers, the EER should be 1.00, which none of the experimented anonymization methods achieve. Note that in 1-sample t-tests ($dof = 3$) against

Setting	Transcription
Ground truth	IN THE SAME WAY THAT MISTER CARKER USED TO FLASH HIS TEETH AND MISTER JOHN COLLIER GIVES HIS SITTER A CHEERFUL SLAP ON THE BACK
Baseline	IN THE SAME WAY THAT MISTER CARKER USED TO FLASH HIS TEETH AND MISTER JOHN CALLED HERE GIVES HIS SITTER A CHEERFUL SLAP ON THE BACK
Uniform (std=0.01)	IN THE SAME WAY THAT MISTER CARKER MISTER FLASH'S TEETH AND MISTER JOHN COLLIER GIVES HIS CIGAR A CHEERFUL SLAP ON THE BACK
Adaptive ($\gamma = 0.1$)	IN THE SAME WAY THAT MISTER CARKER MISTER FLASH'S TEETH AND MISTER JOHN COLLIER GIVE HIS SITTER A CHEERFUL SLAP ON THE BACK
Adaptive ($\gamma = 0.3$)	IN THE SAME WAY THAT MISTER CARKER MISTER FLASH'S TEETH
Multimodal ($n = 3, \gamma = 0.03$)	IN THE SAME WAY THAT MISTER CARKER USED TO FLASH HIS TEETH AND MISTER JOHN COLLIER GIVES HIS SITTER A CHEERFUL SLAP ON THE BACK
Multimodal ($n = 3, \gamma = 0.1$)	IN THE SAME WAY THAT MISTER CARKER MISTER FLASHES TEETH

Table 1: An excerpt of some transcriptions in different configurations.

the baseline, none turned out to have $p < 0.05$ due to small sample sizes. Additional launches of experiments with other random seeds can make the finding more solid.

The WER values in most settings appear larger than that of the baseline, indicating that the anonymization methods sacrifice some informativeness.

5.4 Ablating on the scaling factor

It is hypothesized that in all of the uniform noise, adaptive noise, and the multimodal noise settings, the EER and WER both increase with the scale of the noise. This trend is supported in the uniform and the adaptive noise settings but less so in the multimodal noise settings.

5.5 Ablating on the GMM number of components

Another important hyperparameter in the multimodal noise setting is the number of components in the Mixture of Gaussian model. If we increase the number of components, the distribution of the noise (which is a scaled version of the Gaussian mixture) would have the capacity to mask a larger portion of the transmitted information, resulting in a higher EER and a larger WER. As shown in the last a few rows in Table 2, the contrary might be more likely, while the differences between $n = 3$, $n = 5$, and $n = 8$ are not statistically significant.

6 Discussion

Do the noise addition mechanisms work? Compared to the baseline, adding noise to the voice audio files indeed improves the EER while compromising the WER slightly. The actual extent

Method	Hyperparameter setting	EER	WER
Baseline (no noise)	N/A	0.4128	0.0315
Uniform noise per utterance	$\sigma = 0.003$	0.4966 ± 0.0628	$^{*}0.0352 \pm 0.0004$
	$\sigma = 0.01$	0.5077 ± 0.0738	0.0430 ± 0.0133
	$\sigma = 0.03$	0.5478 ± 0.0808	$^{**}0.1458 \pm 0.0023$
Adaptive noise per utterance	$\gamma = 0.03$	0.5521 ± 0.0883	$^{*}0.0332 \pm 0.0003$
	$\gamma = 0.1$	0.4897 ± 0.1129	$^{**}0.0371 \pm 0.0006$
	$\gamma = 0.3$	0.4855 ± 0.0282	$^{**}0.0650 \pm 0.0002$
Multimodal noise per utterance	$n = 3, \gamma = 0.03$	0.4359 ± 0.0622	$^{**}0.1064 \pm 0.0024$
	$n = 3, \gamma = 0.1$	0.5769 ± 0.1764	$^{**}0.1084 \pm 0.0034$
	$n = 3, \gamma = 0.3$	0.5068 ± 0.1745	$^{**}0.1072 \pm 0.0027$
	$n = 5, \gamma = 0.1$	0.5821 ± 0.1336	$^{**}0.0626 \pm 0.0031$
	$n = 8, \gamma = 0.1$	0.5180 ± 0.0468	$^{**}0.0409 \pm 0.0010$

Table 2: Comparison of results, with mean $\pm$ std of the trials with different random seeds. The * and ** markers refer to $p < 0.05$ and $p < 0.01$, respectively, in a one-sample t-test (two-tailed, $dof = 3$) against the baseline.

of the improvement and compromise should be further scrutinized via statistical tests with higher degrees of freedom.

Following this avenue of research, several other approaches can be evaluated as well. For example, the adaptive noise can be expanded to speaker-level. In LibriSpeech, each speaker participates in multiple sessions, each of which contains multiple utterances. This project only considers utterance-level noises, but speaker-level modeling might introduce anonymization mechanisms that can capture more speaker-specific features.

In addition, the baselines could be made stronger. Two popular benchmark systems, x-vector, and McAdams coefficient adjustment should be compared against using the codebase of the VoicePrivacy challenge when the time and computation resource allows.

Moreover, the route toward speaker anonymization can be equipped with the theoretical bases of differential privacy (DP). For a wide variety of noises, DP provides a theoretical guarantee that the outputs of a system do not change much with a high probability, when the, e.g., added noise is sufficiently large. The system could be a speaker verification or speech recognition one. However, how to reconcile the two systems in a theoretical-driven method requires further exploration.

7 Conclusion

Speaker anonymization is a direction that has become increasingly popular. In this project, I evaluate three methods of adding utterance-level noises to the speaker's voice: uniform, adaptive Gaussian, and multimodal Gaussian. In addition to some qualitative case studies, the equal error rate (in speaker verification) and the word error rate (in speech recognition) are computed to evaluate the efficacy of the anonymization methods.

References

[1] J. Qian, H. Du, J. Hou, L. Chen, T. Jung, and X. Li, "Speech sanitizer: Speech content desensitization and voice anonymization," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[2] N. Tomashenko, X. Wang, X. Miao, H. Nourtel, P. Champion, M. Todisco, E. Vincent, N. Evans, J. Yamagishi, and J. F. Bonastre, "The voiceprivacy 2022 challenge evaluation plan," *arXiv preprint arXiv:2203.12468*, 2022.

[3] Q. Jin, A. R. Toth, T. Schultz, and A. W. Black, "Speaker de-identification via voice transformation," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2009, pp. 529–533.

[4] J. Cohen, T. Kamm, and A. G. Andreou, "Vocal tract normalization in speech recognition: Compensating for systematic speaker variability," *The Journal of the Acoustical Society of America*, vol. 97, no. 5, pp. 3246–3247, 1995.

[5] E. Eide and H. Gish, "A parametric approach to vocal tract length normalization," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 346–348.

[6] J. Patino, N. Tomashenko, M. Todisco, A. Nautsch, and N. Evans, "Speaker anonymisation using the mcadams coefficient," *arXiv preprint arXiv:2011.01130*, 2020.

[7] S. E. McAdams, *Spectral fusion, spectral parsing and the formation of auditory images*. Stanford university, 1984.

[8] W.-C. Huang, T. Hayashi, X. Li, S. Watanabe, and T. Toda, "On prosody modeling for asr+ tts based voice conversion," *arXiv preprint arXiv:2107.09477*, 2021.

[9] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker anonymization using x-vector and neural waveform models," *arXiv preprint arXiv:1905.13561*, 2019.

[10] G. P. Prajapati, D. K. Singh, P. P. Amin, and H. A. Patil, "Voice privacy using cyclegan and time-scale modification," *Computer Speech & Language*, vol. 74, p. 101353, 2022.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[12] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

A Computation budget

The computation resources used in this project are listed as follows:

- Noise addition: within seconds for uniform noise and adaptive noise. The procedure takes between 30 and 60 minutes for the multimiodal Gaussian mixture noises, for all utterances.
- ASR transcription: between 5.5 and 7 hours for 2,703 utterances.

Other computations are finished within seconds. All operations are done on a cloud-based CPU cluster. Overall, there are 34 sets of configurations, each taking around 6 hours. In total, 204 hours of CPU time is consumed.