A Speaker De-identification System based on Sound Processing

Mihai-Andrei Costandache
Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi, Romania *andrei97mihai@gmail.com*

Adrian Iftene
Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi, Romania *adiftene@info.uaic.ro*

Daniela Gîfu
Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi, Romania & Institute of Computer Science, Romanian Academy, Iasi Branch *daniela.gifu@uaic.ro*

Abstract

In the context of products employing speech recognition, where the speech signal is sent from the device to centralized servers that process data, or simply products that involve data storage on servers, privacy for audio data is an important issue, just as it is for other types of data. Ignoring privacy has consequences for both, speakers (information leaks) and server administrators (legal issues). In this paper, we propose a speaker de-identification solution based on sound processing, altering voice characteristics, along with an API. Our solution consisting of pitch shift and noise mix (the latter is an optional augmentation method) has a great speaker de-identification performance, without an important loss in terms of word intelligibility. It is worth mentioning that sometimes the recordings may not be easy to understand in the initial (i.e., not de-identified) form, due to the speaker's pronunciation, talking speed, and other related factors.

Keywords: speaker de-identification system, pitch shift, noise mix, sound processing

1. Introduction

Voice privacy, also referred to as speaker anonymization [2] or de-identification [9], is an essential security aspect in different communication contexts. Privacy in general (i.e., for all data types) is the reason for adopting the recent law, General Data Protection Regulation (EU) 2016/679 (GDPR) [12]. In order to preserve the speaker's privacy, some solutions have been proposed. Beyond cryptography-based solutions [14], there are approaches consisting in the removal of personally identifiable information within a speech signal [6, 7], [21].

The main hypothesis of this paper is that the *speaker de-identification can be performed more securely by sound processing*.

We propose a speaker de-identification solution based on sound processing, which modifies the voice in order to remove as much as possible from the particularities that lead to speaker recognition, with the words remaining intelligible. We give some use-cases, sorted in ascending order by the severity of disclosing someone's identity: low - product guide, faculty admission committee guidance, medium - teacher evaluation, vocal command application, support group meeting, and high - business meeting, trial. The rest of the paper is organized as follows: Section 2 summarizes existing speaker de-identification approaches, with applicability in security, and presents available tools with voice processing features, used for entertainment. Section 3 refers to the solution we propose, in terms of de-identification techniques and system architecture. Section 4 presents the evaluation of our solution, involving human participants, before drawing some conclusions in the last section.

2. Related Works

In this section, we introduce existing approaches in the security context of speaker de-identification and also products used for entertainment. Essentially, the speaker-dependent variability is based on parameters related to the speaker (e.g., age [4], gender [8], pathologies [1]) and the environment (e.g., noise, channels).

In [14], there are proposed two privacy-preserving frameworks: rendering secure algorithms by employing techniques such as homomorphic encryption, secure multi-party computation, and oblivious transfer, and modifying voice pattern classification tasks into string comparison operations. [6] presents the modification of a source speaker's voice into a target speaker's voice by using Gaussian Mixture Model (GMM) mappings, and [7] describes an approach consisting of diphone recognition, done mainly by Hidden Markov Models (HMMs) and GMMs, and speech synthesis, HMM-based or diphone-based; The adversarial representation learning technique in [21] has the goal of learning representations that perform well in speech recognition but hide the speaker's identity and it mainly consists of an encoder, a decoder (this component is for speech recognition), and an adversarial branch (this component is for speaker identification).

We believe that the security tools are less known to the general public, but very useful for enterprises or governmental institutions. The products used for entertainment seem to be more available, Table 1 describes some of them. They usually provide voice modification features (custom or preset) and sound effects. The fact that these products seem to be easier to obtain emphasizes that our solution is needed.

Table 1. Existing tools, but used for entertainment

Product	Platforms	Usage	Additional Features
Voxal Voice Changer [23]	Windows & Mac OS X	general	text-to-speech
Skype Voice Changer [19]	Windows	Skype	sound player, text-to-speech, recorder
PyVoiceChanger [16]	Linux	general	-

We can draw the conclusion from the relevant work on this topic that speaker de-identification is a task with multiple approaches, but very difficult.

3. Proposed Solution

In this section, we present our solution, which does not consist just of speaker de-identification techniques, as we also developed a system with multiple components, including an API.

3.1. Technologies

We used Python 3.8.8 [15] and several modules. The most important modules were LibROSA 0.8.0 [10] (sound processing), NumPy 1.20.1 [5] (numerical computing and n-dimensional arrays), Flask 1.1.2 [3] (API building), and Requests 2.25.1 [17] (API requests sending). The audio file codecs our product supports depend on the codecs managed by an important dependency of LibROSA, called SoundFile [20] (e.g., usual variants of WAV, FLAC). Also, it is worth mentioning that we decided to convert the recordings to mono (one channel).

3.2. System Components

The system we developed has four components:

- **Sound processing component** - audio input/output functionality and speaker de-identification techniques, pitch shift and noise mix;
- **Small framework for performing multiple de-identification runs** - component that ensures experimental repeatability and reproducibility, by separating the input and output data, in a rigorous but easy to understand way;
- **Small framework for evaluation** - data management and metrics computation;
- **API** - the speaker de-identification product the users interact with.

The main components are the sound processing component and the API. We chose to provide services through an API and not through a specialized application, in order to cover multiple, potential use-cases. There are two security degrees, normal and advanced, that differ in terms of the used techniques/parameter values. The users do not have to register and their recordings are not kept on the server. The API provides both de-identification and recording reconstruction, the latter feature being done by inverting the de-identification operations. Due to factors such as the conversion to mono, a file obtained by reconstruction is not 100% the original, but the differences are not relevant. The system performs the following steps:

De-identification

1. Get the security degree from the route the request was sent to (`<de-identification_route>/("normal"|"advanced")`) and the recording;
2. Pick technique(s) and parameter values, according to the security degree;
3. Perform the necessary processing;
4. Generate an API key;
5. Save the operations needed for the reconstruction of the original recording, represented by the inverses of all modification operations in inverse order, in a place that is identified by the API key; A hash is stored, not the key itself;
6. Send to the user the de-identified recording and the API key.

Reconstruction

1. Get the recording and the API key, sent by the user at the specific route (`<reconstruction_route>`);
2. Load the necessary operations to perform the reconstruction according to the key;
3. Perform the necessary processing;
4. Send to the user the reconstructed recording.

We also provided a `<help_route>`. Figure 1 presents the most important components.

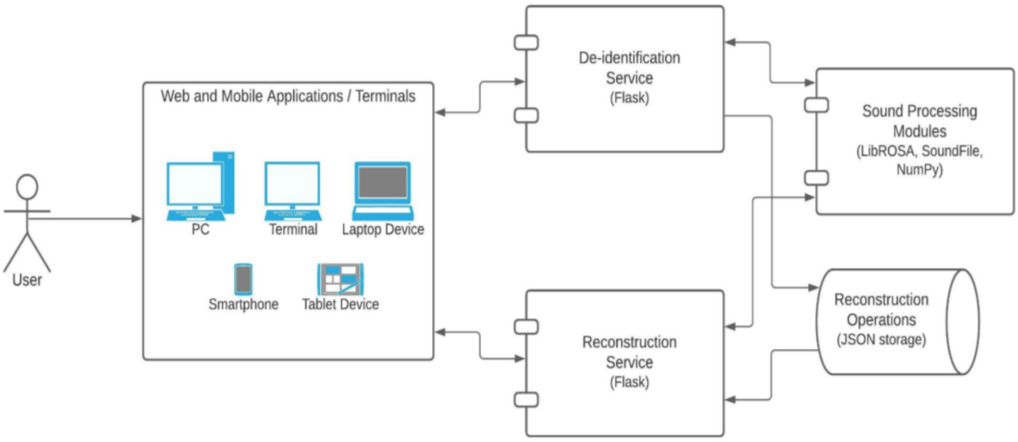
De-identification Service (Flask)
Web and Mobile Applications / Terminals
Sound Processing Modules (LibROSA, SoundFile, NumPy)
PC
Terminal
Laptop Device
User
Smartphone
Tablet Device
Reconstruction Operations (JSON storage)
Reconstruction Service (Flask)

Fig. 1. Main components (Sound processing and API) architecture

3.3. Speaker De-identification Techniques Description

In the context of pitch shift, we explain the first two terms [24], semitone and octave (from music, but important in our context). The semitone is the interval between two adjacent pitches, while the octave (12 semitones) is the interval between two pitches that have a 1:2 (or 2:1) frequency ratio. The usual voice pitch range [13] is 125 Hz (the average of adult male speech) - 500 Hz (the minimum of baby cries). The LibROSA package provides the *pitch_shift* function, that time-stretches, resamples, and crops to the initial dimension (basically, NumPy array size) an input signal. Two of its parameters are *n_steps* (how much to modify the voice) and *bins_per_octave* (how is an octave divided). A positive *n_steps* value makes the pitch higher, while a negative value makes it lower. The value of the parameter *bins_per_octave* is by default 12, but we used 24, in order to control the steps better (they can be considered quarter-tones). To get the inverse operation we multiply by -1 the *n_steps* value.

Regarding noise mix, we consider it an augmentation method for pitch shift, which is the main speaker de-identification technique. Thus, we only use noise mix as an optional addition to pitch shift, not independently. An important aspect is the signal-to-noise ratio (SNR) [18], a metric computed by dividing the level of a signal by the level of noise and usually expressed in decibels (dB). There are two main types of noise [11]: Additive White Gaussian Noise (AWGN) and real world noise. We only used the former, which consists in noise generated using a Gaussian distribution with the mean value of 0. AWGN is mixed with a signal by addition and to get the inverse operation we multiply by -1 all noise samples.

4. Evaluation

In this section, we describe how we measured the quality of our solution. Our approach implies human evaluators and it can be used to test similar systems, with few customizations. We simulated the teacher evaluation process that is performed at the end of each term at our faculty. Basically, the students provide feedback on the teachers, taking into consideration aspects such as the ability to explain the discussed topics well, the objectivity in grading, and the organizational skills. We chose to simulate this scenario because of its academically character.

4.1. Overview

We approached the speaker identification and word intelligibility problems. We used accuracy for speaker identification and word error rate (WER) [22] for speech recognition. Both metrics need to be low. WER is obtained by dividing the number of all editing operations (i.e., substitutions, insertions, and deletions), necessary in order to get the hypothesis (assumed) text from the reference (real) text, by the number of words in the reference text.

4.2. Participants

The test participants were four teachers and six students. The students were 23-25 years old. They are colleagues and the teachers work/worked with them, so the listeners recognize the speakers in normal conditions. We also consider two fictional faculty teachers. We chose to add fictional persons in order to make the students more comfortable giving reviews.

4.3. Stages

There were three stages: (1) tasks (preceded by an introduction), (2) a post-test Questionnaire for User Interaction Satisfaction (QUIS), with integer scores between 1 ("never") and 9 ("always"), and (3) post-test open-ended questions, asking the participants what they (dis)liked, and what recommendations did they have. In the following, we present the tasks. The students recorded themselves giving a grade (mandatory item, required a 0-4 integer value), positive and negative comments, and suggestions for the fictional teachers (distinct recordings), and provided the transcriptions of what they said. All the participants then tried to recognize the speakers and to understand the words in de-identified versions of the recordings. The students did not perform the last tasks for their own recordings and also all the participants were imposed a maximum of eight plays per recording.

4.4. Data

The recordings were WAV files, 21-35 seconds long (only the former was a requirement). We applied pitch shift with the *bins_per_octave* value of 24, without noise mix, for the recordings we gave back to the participants. We noticed that the word intelligibility decreases quicker for the low-pitched voices, so we used less steps (in absolute values) for them. On average, for the high-pitched voices *n_steps* is 24 and for the low-pitched ones it is -15.

4.5. Results

Separately provided for teachers/students, the accuracy of the speaker identity assumptions and the WER on average for high/low-pitched voices can be seen in Table 2.

Table 2. Teachers/students average accuracy and WER for high/low-pitched voices

High Pitch Category		
Metrics	**Teachers**	**Students**
Accuracy (%)	16.67	83.33
WER (%)	33.33	16.67
Low Pitch Category		
Metrics	**Teachers**	**Students**
Accuracy (%)	66.67	83.33
WER (%)	47.5	16.67

Separately provided for teachers/students, the QUIS scores (1-9 scale) on average can be seen in Table 3.

Table 3. Teachers/students QUIS average scores

Opinions	Teachers	Students
The speakers' identities cannot easily be detected.	7	8.33
Words can be understood without too much effort.	5	7
The recordings, although they may sound weird, are tolerable.	7	7
A task done in an audio-based manner, provided with a de-identification service, can replace the same task performed in a text-based manner.	7	6.67
Generally speaking, this service works well.	7	7.33

From the participants' answers to the open-ended questions, we mainly found out that they considered that de-identification is achieved and they had the following ideas: make all recordings have approximately the same volume, add real-time processing, and add sentiment analysis in order to use high pitch for a positive review and low pitch for a negative one.

We provide the following experimental conclusions, which depend on the goal of obtaining a balance between the speaker's identity protection and the word intelligibility:

- The speakers/listeners were not English natives, therefore for the original recordings we could have had approximately a 10% WER;
- According to Table 2, the high-pitched voices are better for speaker de-identification and easier to understand than the low-pitched ones;
- The pitch should be modified between half an octave and an octave, up or down;
- The identity of the speaker is difficult to determine, usually, the participants had to consider speech particularities (e.g., talking speed, English speaking skills), analyze the words, make correlations, use the process of elimination, or guess;
- From our experience, as we did not test noise mix with the participants, the best values for the signal-to-noise ratio (SNR) were approximately in the [-10dB, 20dB] interval. These values depended on the recordings, so it is difficult to generalize.

5. Conclusions

This paper highlights that providing privacy in audio recordings is just as important as in textual or other types of data. There are multiple use-cases, therefore we had a general approach to the problem, leaving to other programmers the task of building customized products, using our API. Our solution using pitch shift (optionally augmented with noise mix) performs speaker de-identification very well, without altering too much the word intelligibility. In many cases of recordings that cannot be understood properly, the problem is mainly due to issues already present in the original files (e.g., speaker's pronunciation). The evaluation section presented our findings about the implemented speaker de-identification solution, supporting the previous claims. The test was made possible by human evaluators.

Acknowledgments

This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/H2020-SC1-2019-Two-Stage-RTD.

References

1. Botelho, C., Teixeira, F., Rolland, T., Abad, A., Trancoso, I.: Pathological Speech Detection Using X-Vector Embeddings. arXiv:2003.00864 (2020)
2. Fang, F., Wang, X., Yamagishi, J., Echizen, I., Todisco, M., Evans, N., Bonastre, J.-F.: Speaker Anonymization Using X-vector and Neural Waveform Models. In Proceedings of the 10th ISCA Speech Synthesis Workshop, Vienna, 155-160 (2019)
3. Flask, https://flask.palletsprojects.com/en/1.1.x/. Accessed June, 2021
4. Ghahremani, P., Nidadavolu, P.S., Chen, N., Villalba,J., Povey, D., Khudanpur, S., and Dehak, N.: End-to-end Deep Neural Network Age Estimation. In: Proceedings of the 19th Annual Conference of the International Speech Communication Association Interspeech 2018, Hyderabad, 277-281 (2018)
5. Harris, C. R., Millman, K. J., van der Walt, S. J. et al.: Array programming with NumPy. Journal Nature 585, 357-362 (2020)
6. Jin, Q., Toth, A.R., Schultz, T., Black, A.W.: Speaker de identification via voice transformation. In: Proceedings of the 11th biannual IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2009, Merano, 529-533 (2009)
7. Justin, T., Struc, V., Dobrisek, S., Vesnicer, B., Ipsic, I., Mihelic, F.: Speaker De-identification using Diphone Recognition and Speech Synthesis. In: Proceedings of the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG). Ljubljana (2015)
8. Kotti, M., Kotropoulos, C.: Gender Classification in Two Emotional Speech Databases. In: Proceedings of the 19th International Conference on Pattern Recognition. Tampa (2008)
9. Magariños, C., Lopez-Otero, P., Docio-Fernandez, L., Rodriguez-Banga, E., Erro, D., Garcia-Mateo, C.: Reversible speaker de-identification using pre-trained transformation functions. Computer Speech & Language, 46, 36-52 (2017)
10. McFee, B., Lostanlen, V., Metsai, A., McVicar, M. et al.: librosa/librosa: 0.8.0. DOI: 10.5281/zenodo.3955228.
11. Medium, https://medium.com/analytics-vidhya/adding-noise-to-audio-clips-5d8cee24ccb8. Accessed June, 2021
12. Nautsch, A., Jasserand, C., Kindt, E., Todisco, M., Trancoso, I., Evans, N.: The GDPR & Speech Data: Reflections of Legal and Technology Communities, First Steps Towards a Common Understanding. In Proceedings of the 20th Annual Conference of the International Speech Communication Association Interspeech 2019, Graz, 3695-3699 (2019)
13. NCVS, http://www.ncvs.org/ncvs/tutorials/voiceprod/tutorial/influence.html. Accessed June, 2021
14. Pathak, M. A., Raj, B., Rane, S., Smaragdis, P.: Privacy-Preserving Speech Processing: Cryptographic and String-Matching Frameworks Show Promise. International Journal IEEE Signal Processing Magazine, 30(2), 62-74 (2013)
15. Python, https://docs.python.org/3.8/. Accessed June, 2021
16. PyVoiceChanger, https://github.com/juancarlospaco/pyvoicechanger. Accessed June, 2021
17. Requests, https://docs.python-requests.org/en/master/. Accessed June, 2021
18. Rode, https://www.rode.com/blog/all/what-is-signal-to-noise-ratio. Accessed June, 2021
19. Skype Voice Changer, https://skypevoicechanger.net/. Accessed June, 2021
20. SoundFile, https://pypi.org/project/SoundFile/. Accessed June, 2021
21. Srivastava, B. M. L., Bellet, A., Tommasi, M., Vincent, E.: Privacy-Preserving Adversarial Representation Learning in ASR: Reality or Illusion?. In: Proceedings of the 20th Annual Conference of the International Speech Communication Association Interspeech 2019, Graz, 3700-3704 (2019).
22. Vocapia Research, https://www.vocapia.com/glossary.html. Accessed June, 2021
23. Voxal Voice Changer, https://www.nchsoftware.com/voicechanger/index.html. Accessed June, 2021
24. Young Scientists Journal, https://ysjournal.com/mathematics-in-music/. Accessed June, 2021