Privacy-Preserving Adversarial Representation Learning in ASR: Reality or Illusion?
Brij Mohan Lal Srivastava, Aurélien Bellet, Marc Tommasi, Emmanuel Vincent

To cite this version:
Brij Mohan Lal Srivastava, Aurélien Bellet, Marc Tommasi, Emmanuel Vincent. Privacy-Preserving Adversarial Representation Learning in ASR: Reality or Illusion?. INTERSPEECH 2019 - 20th Annual Conference of the International Speech Communication Association, Sep 2019, Graz, Austria. hal-02166434

HAL Id: hal-02166434
https://inria.hal.science/hal-02166434
Submitted on 3 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Privacy-Preserving Adversarial Representation Learning in ASR: Reality or Illusion?

Brij Mohan Lal Srivastava\textsuperscript{1}, Aurélien Bellet\textsuperscript{1}, Marc Tommasi\textsuperscript{2}, Emmanuel Vincent\textsuperscript{3}

\textsuperscript{1}INRIA, France \textsuperscript{2}Université de Lille, France \textsuperscript{3}Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
\{brij.srivastava, aurelien.bellet, marc.tommasi, emmanuel.vincent\}@inria.fr

Abstract

Automatic speech recognition (ASR) is a key technology in many services and applications. This typically requires user devices to send their speech data to the cloud for ASR decoding. As the speech signal carries a lot of information about the speaker, this raises serious privacy concerns. As a solution, an encoder may reside on each user device which performs local computations to anonymize the representation. In this paper, we focus on the protection of speaker identity and study the extent to which users can be recognized based on the encoded representation of their speech as obtained by a deep encoder-decoder architecture trained for ASR. Through speaker identification and verification experiments on the Librispeech corpus with open and closed sets of speakers, we show that the representations obtained from a standard architecture still carry a lot of information about speaker identity. We then propose to use adversarial training to learn representations that perform well in ASR while hiding speaker identity. Our results demonstrate that adversarial training dramatically reduces the closed-set classification accuracy, but this does not translate into increased open-set verification error hence into increased protection of the speaker identity in practice. We suggest several possible reasons behind this negative result.

Index Terms: speech recognition, end-to-end system, privacy, adversarial training, speaker recognition

1. Introduction

With the emergence of pervasive voice assistants [1, 2] like Amazon Alexa, Apple’s Siri and Google Home, voice has become one of the most widespread forms of human-machine interaction. In this context, the speech signal is sent from the user device to a cloud-based service, where automatic speech recognition (ASR) and natural language understanding are performed in order to address the user request.\textsuperscript{1} While recent studies have identified security vulnerabilities in these devices [3,4], such studies tend to hide more important privacy risks that can have long-term impact. Indeed, state-of-the-art speech processing algorithms can infer not only the spoken contents from the speech signal, but also the speaker’s identity [5], intention [6–9], gender [10,11], emotional state [12–14], pathological condition [15–17], personality [18,19] and cultural [20,21] attributes to a great extent. These algorithms require just a few tens of hours of training data to achieve reasonable accuracy, which is easier than ever to collect via virtual assistants. The dissemination of voice signals in large data centers thereby poses severe privacy threats to the users in the long run.

These privacy issues have little been investigated so far. The most prominent studies use homomorphic encryption and bit string comparison [22, 23]. While these methods provide strong cryptographic guarantees, they come at a large computational overhead and can hardly be applied to state-of-the-art end-to-end deep neural network based systems.

An alternative software architecture is to pre-process voice data on the device to remove some personal information before sending it to web services. Although this does not rule out all possible risks, a change of representation of the voice signal can contribute to limiting unsolicited uses of data. In this paper, we investigate how much of a user’s identity is encoded in speech representations built for ASR. To this end, we conduct closed- and open-set speaker recognition experiments. The closed-set experiment refers to a classification setting where all test speakers are known at training time. In contrast, the open-set experiment (a.k.a. speaker verification) aims to measure the capability of an attacker to discriminate between speakers in a more realistic setting where the test speakers are not known beforehand. We implement the attacker with the state-of-the-art x-vector speaker recognition technique [24].

The representations of speech we consider in our work are given by the encoder output of end-to-end deep encoder-decoder architectures trained for ASR. Such architectures are natural in our privacy-aware context, as they correspond to encoding speech on the user device and decoding in the cloud. Our baseline network follows the ESPnet architecture [25], with one encoder and two decoders: one based on connectionist temporal classification (CTC) and the other on an attention mechanism. Inspired by [26], we further propose to extend the network with a speaker-adversarial branch so as to learn representations that perform well in ASR while hiding the speaker identity.

Several papers have recently proposed to use adversarial training for the goal of improving ASR performance by making the learned representations invariant to various conditions. While general form of acoustic variabilities have been studied [27], there is some work specifically on speaker invariance [28, 29]. Interestingly, there is no general consensus on whether it is more appropriate to use speaker classification in an adversarial or a multi-task manner, despite the fact that these two strategies implement opposite means (i.e., encouraging representations to be speaker-invariant or speaker-specific). This question was studied in [30], in which the authors conclude that both approaches only provide minor improvements in terms of ASR performance. Their speaker classification experiments also show that the baseline system already tends to learn speaker-invariant features. However, they did not run speaker verification experiments and hence did not assess the suitability of these features for the goal of anonymization.

In contrast to these studies which aim to increase ASR performance, our goal is to assess the potential benefit of adversarial training for concealing speaker identity in the con-
text of privacy-friendly ASR. Our contributions are the following. First, we combine CTC, attention and adversarial learning within an end-to-end ASR framework. Second, we design a rigorous protocol to quantify speaker identity in ASR representations through a series of closed-set classification and open-set verification experiments. Third, we run these experiments on the Librispeech corpus [31] and show that this framework dramatically reduces speaker classification accuracy, but does not increase speaker verification error. We suggest several possible reasons behind this disparity.

The structure of the rest of the paper is as follows. In Section 2, we describe the baseline ASR model and our proposed adversarial model. Section 3 explains the experimental setup and presents our results. Finally, we conclude and discuss future work in Section 4.

2. Proposed model

We start by describing the ASR model we use as a baseline, before introducing our speaker-adversarial network.

2.1. Baseline ASR model

We use the end-to-end ASR framework presented in [32] as the baseline architecture. It is composed of three sub-networks: an encoder which transforms the input sequence of speech feature frames, a decoder branches: CTC and attention decoders respectively. The model is trained in an end-to-end manner by optimizing the following min-max objective:

$$\min_{\theta_e} \max_{\theta_a} L_{\text{loss}}(\theta_e; \theta_a) = \lambda L_c(\theta_e; \theta_a) + (1 - \lambda) L_a(\theta_e; \theta_a),$$

with $\lambda \in [0, 1]$ a trade-off parameter between the two decoders. We denote each sample in the dataset as $S_i = (X_i, Y_i, z_i)$, where $X_i = \{x_1, ..., x_T\}$ is the sequence of $T$ acoustic feature frames, $Y_i = \{y_1, ..., y_M\}$ is the sequence of $M$ character labels in the transcription, and $z_i$ is the speaker label.

In the case of CTC, several intermediate label sequences of length $T$ are created by repeating characters and inserting a special `blank` label to mark character boundaries. Let $\Psi(Y_i)$ be the set of all such intermediate label sequences. The CTC loss $L_c(\theta_e; \theta_a)$ is computed as

$$L_c(\theta_e; \theta_a) = -\ln \prod_{i=1}^{T} P(y_i \mid X_i; \theta_e; \theta_a),$$

where $y_i$ is the character generated every $\tau$ timesteps, $\tau$ is the CTC blank insertion rate, $m$ is number of characters, $\lambda$ is the trade-off parameter between the two decoders.

The attention branch does not require intermediate label representation and conditional independence is not assumed. Hence, the loss is simply computed as

$$L_a(\theta_e; \theta_a) = -\sum_{m \in M} \ln P(y_m \mid X_i; y_i; m-1; \theta_e; \theta_a).$$

2.2. Speaker-adversarial model

In order to encourage the network to learn representations that are not only good at ASR but also hide speaker identity, we propose to extend the above architecture with what we call a speaker-adversarial branch. This branch models an adversary which attempts to infer the speaker identity from the encoded representation $\phi$. We denote by $\theta_s$ the parameters of the speaker-adversarial branch. Given the encoder parameters $\theta_e$, the goal of the adversary is to find $\theta_s$ that minimizes the loss $L_{\text{spk}}(\theta_s; \theta_a) = -\ln P(z_i \mid X_i; \theta_e; \theta_s)$. Our new model is then trained in an end-to-end manner by optimizing the following min-max objective:

$$\min_{\theta_e} \max_{\theta_s} L_{\text{loss}}(\theta_e; \theta_s; \theta_a) - \alpha L_{\text{spk}}(\theta_e; \theta_s),$$

where $\alpha \geq 0$ is a trade-off parameter between the ASR objective and the speaker-adversarial objective. The baseline network can be recovered by setting $\alpha = 0$. Note that the max part of the objective corresponds to the adversary, which controls only the speaker-adversarial parameters $\theta_s$. The goal of the speaker-adversarial branch is to act as a "good adversary" and produce useful gradients to remove the speaker identity information from the encoded representation $\phi$. In practice, we use a gradient reversal layer [33] between the encoder and the speaker-adversarial branch so that the whole network can be trained end-to-end via backpropagation. We refer to Fig. 1 for an illustration of the full architecture.

3. Experimental evaluation

3.1. Datasets

We use the Librispeech corpus [31] for all the experiments. We use different subsets for ASR training, adversarial training, and speaker verification. For the sake of clarity we refer to them as data-full, data-adv, and data-spkv, respectively (see Table 1).
The data-full set is almost the original Librispeech corpus, including train-960 for training, dev-clean and dev-other for validation, and test-clean and test-other for test, except that utterances with more than 3,000 frames or more than 400 characters have been removed from train-960 for faster training.

The data-adv set is a 100 h subset of train-960, which is obtained by removing long utterances from the original Librispeech train-100 set similarly to above. It is split into three subsets in order to perform closed-set speaker identification experiments, since the speakers in the original train/dev/test splits are disjoint. There are 251 speakers in data-adv: we assign 2 utterances per speaker to each test-adv and dev-adv. The remaining utterances are used for training and referred to as train-adv.

For speaker verification with x-vectors [24], we use data-spkv, which is again derived from data-full. The train-960 subset was augmented using room impulse responses, isotropic and point-source noises [34] as well as music and speech [35] as per the standard sre16 recipe for training x-vectors [24] from the Kaldi toolkit [36], which we adapted to Librispeech. This increased the amount of data by a factor of 4. A subset of the augmentation set (train-plda and sre16 point-source noises [34] as well as music and speech [35] as per the standard sre16 recipe for training x-vectors [24] from the Kaldi toolkit [36], which we adapted to Librispeech. This increased the amount of data by a factor of 4. A subset of the augmented data containing 373,985 utterances was used to train the x-vector representation and another subset containing 422,491 utterances to train the probabilistic linear discriminant analysis (PLDA) backend. These subsets are referred to as train-spkv and train-plda, respectively. For evaluation, we build an enrollment set (test-clean-enroll) and a trial set (test-clean-trial) from the test-clean data. Out of 40, 29 speakers were selected from test-clean based on sufficient data availability. For each speaker, we selected a 1 min subset after speech activity detection for enrollment and used the rest for trials. The details of the trials are given in Table 2.

| Table 1: Splits of Librispeech used in our experiments. |
|----------------|----------------|----------------|
| dataset       | data split     | # utts         | duration (h) |
| data-full      | train-960      | 281,231        | 960.98       |
|                | test-clean     | 2,620          | 5.40         |
|                | dev-clean      | 2,703          | 5.39         |
|                | test-other     | 2,939          | 5.34         |
|                | dev-other      | 2,864          | 5.12         |
| data-adv       | train-adv      | 27,535         | 97.05        |
|                | dev-adv        | 502            | 1.77         |
|                | test-adv       | 502            | 1.77         |
| data-spkv      | train-spkv     | 373,985        | 1,388.79     |
|                | train-plda     | 422,491        | 1,443.96     |
|                | test-clean-enroll | 438 | 0.75         |
|                | test-clean-trial | 1496 | 3.60         |

| Table 2: Detailed description of the trial set (test-clean-trial) for speaker verification experiments. |
|----------------|----------------|
|                | Male | Female |
| # Speakers     | 13   | 16     |
| # Genuine trials | 449  | 548    |
| # Impostor trials | 9,457 | 11,196 |

3.2. Evaluation metrics

For all tested systems, we measure ASR performance in terms of the word error rate (WER) and we assess the amount of information about speaker identity in the encoded speech representation in terms of both speaker classification accuracy (ACC) and speaker verification equal error rate (EER). The WER is reported on the test-clean set. The ACC measures how well speakers can be discriminated in a closed-set setting, i.e., speakers are known at training time. It is evaluated over the test-adv set using the same classifier architecture as the speaker-adversarial branch of the proposed model (see Section 2). As opposed to the ACC, the EER measures how well the representations hide the speaker identity for unknown speakers, in an open-set scenario. It reflects the process of confirming whether a person is actually who the attacker thinks it might be. It is evaluated over the trial set (see Table 2) using x-vector-PLDA.

The ACC and the EER will be computed for different representations: the baseline filterbank features, the representations encoded by the network trained for ASR only (corresponding to $\phi_0$) as well as those obtained with the speaker-adversarial approach (corresponding to $\phi_\alpha$, for some values of $\alpha > 0$).

3.3. Network architecture and training

For all experiments, we use the ESPnet [25] toolkit which implements the hybrid CTC/attention architecture [32]. The input features are 80-dimensional mel-scale filterbank coefficients with pitch and energy features, totalling 84 features per frame. The encoder is composed of a VGG-like convolutional neural network (CNN) layer followed by 5 bidirectional long short-term memory (LSTM) layers with 1,024 units. The VGG layer contains 4 convolutional layers followed by max pooling. The feature maps used in the convolution layers are of dimensions $(1 \times 64), (64 \times 64), (64 \times 128)$ and $(128 \times 128)$. The attention-based decoder consists of location-aware attention [37] with 10 convolutional channels of size 100 each followed by 2 LSTM layers with 1,024 units. The CTC loss is computed over several possible label sequences using dynamic programming. In all experiments, the trade-off parameter $\lambda$ between the two decoder losses is set to 0.5. We train a single-layer recurrent neural network language model (RNNLM) with 1,024 hidden units over the train-960 transcriptions and use it to rescore the ASR hypotheses. The resulting WER is very close to the state of the art [38] when trained on train-960. Finally, we implemented the speaker-adversarial branch via a 3 bidirectional LSTM layers with 512 units followed by a softmax layer with 251 outputs corresponding to the 251 speakers in data-adv. The adversarial loss $L_{adv}$ is summed across all vectors in the sequence. The speaker label $z_i$ is duplicated to match the length of the sequence, which is smaller than $T$ due to the subsampling performed within the encoder. Due to this subsampling as well as to the use of bidirectional LSTM layers within the encoder and the speaker-adversarial branch, the frame-level adversarial loss approximates well a utterance-level speaker loss that would be computed from a fixed-sized utterance-level representation, while being easier to train.

In all experiments, we start by pre-training the ASR branch for 10 epochs over data-full and then the speaker-adversarial branch for 15 epochs on data-adv in order to get a strong adversary on the pre-trained encoded representations. Then, due to time constraints, all networks are fine-tuned on data-adv: we run 15 epochs of adversarial training (which corresponds to simple ASR training when $\alpha = 0$). Due to this, the WER is comparable to that typically achieved by end-to-end methods when trained on the train-100 subset of Librispeech rather than the full train-960 set. Finally, freezing the resulting encoder, we further fine-tune the speaker-adversarial branch only for 5 epochs to make sure that the reported ACC reflects the performance of a well-trained adversary.

The encoder network contains 133.5M parameters. To encode a 10s audio file, it perform 1.1e12 arithmetic operations which can be executed in-parallel on a 40 core CPU in 17.6s
and on a single Tesla P100 GPU in 149ms.

3.4. Results and discussion

We train our speaker-adversarial network for $\alpha \in \{0, 0.5, 2.0\}$, leading to three encoded representations $\phi_\alpha(X)$. Recall that $\alpha = 0$ corresponds to the baseline ASR system as it ignores the speaker-adversarial branch. Table 3 summarizes the results.

The first column presents the ACC and EER obtained with the input filerbank features, which are consistent with the numbers reported in the literature. As expected, speaker identification and verification can be addressed to very high accuracy on those features. Using the encoded representation $\phi_0(X)$ trained for ASR only already provides a significant privacy gain: the ACC is divided by 2 and the EER is multiplied by 4, which suggests that a reasonable amount of speaker information is removed during ASR training. Nevertheless, $\phi_0(X)$ still contains some speaker identity information.

More interestingly, our results clearly show that adversarial training drastically reduces the performance in speaker identification but not in verification. On the other hand, and counterintuitive to the speaker-invariance claims by several previous studies, we observe that the verification performance actually improves after adversarial training. This exhibits a possible limitation in the generalization of adversarial training to unseen speakers and hence establishes the need for further investigation.

The reason for the disparity between classification and verification performance might be that the speaker-adversarial branch does not inherently perform verification and hence is not optimized for that task. It might also be attributed to the representation capacity of that branch, to the number of speakers presented during adversarial training, and/or to the exact range of $\alpha$ needed for generalizable anonymization. These factors of variation open several venues for future experiments.

Table 3: ASR and speaker recognition results with different representations. WER (%) is reported on test-clean set, ACC (%) on test-adv set and EER (%) on test-clean-trial.

<table>
<thead>
<tr>
<th>Filterbank</th>
<th>$\phi_0$</th>
<th>$\phi_{0.5}$</th>
<th>$\phi_{2.0}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER</td>
<td>12.50</td>
<td>12.5</td>
<td>12.5</td>
</tr>
<tr>
<td>ACC</td>
<td>93.1</td>
<td>46.3</td>
<td>6.4</td>
</tr>
<tr>
<td>EER Pooled</td>
<td>5.72</td>
<td>23.07</td>
<td>21.97</td>
</tr>
<tr>
<td>EER Male</td>
<td>3.34</td>
<td>19.38</td>
<td>18.26</td>
</tr>
<tr>
<td>EER Female</td>
<td>7.48</td>
<td>26.46</td>
<td>24.45</td>
</tr>
</tbody>
</table>

We also notice that the WER stays reasonably low and stabilizes to the value of 12.5% after increasing $\alpha$ from 0.5 to 2. In particular, for $\alpha = 2$ the WER is just 1.6% absolute more than the baseline ($\alpha = 0$).

We evaluate whether utterances from the same speaker stay in the same neighborhood or are scattered in the representation space. We compute t-SNE embeddings on the x-vector representations of 20 utterances for 10 speakers (5 male, 5 female), shown in Figure 2. When using filterbanks, we can observe well-clustered utterances. The clusters break down when training the x-vectors on $\phi_0$. For the x-vectors trained on $\phi_{0.5}$ and $\phi_{2.0}$, the clusters start to re-emerge. The silhouette scores for x-vectors extracted from filterbank, $\phi_0$, $\phi_{0.5}$ and $\phi_{2.0}$ representations are 0.14, −0.17, −0.05 and −0.09 respectively, are consistent with the observed EER values.

4. Conclusions and future work

We proposed to combine CTC and attention losses with a speaker-adversarial loss within an end-to-end framework with the goal of learning privacy-preserving representations for ASR. Such representations could be safely transmitted to cloud services for decoding. We investigate the level of speaker identity anonymization achieved by adversarial training through closed-set speaker classification and open-set speaker verification measures. Adversarial training appears to dramatically reduce the closed-set classification accuracy, seemingly indicating a high-level of anonymization. However, this observation does not match with the open-set verification results, which correspond to the real scenario of an adversary trying to confirm the identity of a suspected speaker. Hence we conclude that the adversarial training does not immediately generalize to produce anonymous representations in speech. We hypothesize that this disparity might be attributed to the representation capacity of the adversarial branch, the size of the training set, the formulation of the adversarial loss, and/or the value of the trade-off parameter with the ASR loss. As a future work, we plan to modify the speaker adversarial branch to inherently optimize for verification instead of classification and ascertain the impact of these experimental choices over different datasets, including for languages not seen in training.

5. Acknowledgements

This work was supported in part by the European Union’s Horizon 2020 Research and Innovation Program under Grant Agreement No. 825081 COMPRIS (https://project.inria.fr/comprise/) and by the French National Research Agency under project DEEP-PRIVACY (ANR-18-CE23-0018). Experiments were carried out using the Grid’5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organisations. The authors would like to thank Md Sahidullah for providing the speaker verification data split.
References


