Cost effective, Multilingual, Privacy-driven voice-enabled Services

www.compriseh2020.eu

Call: H2020-ICT-2018-2020
Topic: ICT-29-2018
Type of action: RIA
Grant agreement N°: 825081

WP N°2: Privacy-driven voice interaction
Deliverable N°2.2: Improved transformation library and initial privacy guarantees
Lead partner: INRIA
Version N°: 1.0
Date: 30/04/2020

Document information	
Deliverable N° and title	**D2.2– Improved transformation library and initial privacy guarantees**
Version N°	**1.0**
Lead beneficiary	**INRIA**
Author(s)	**David Adelani (USAAR), Ali Davody (USAAR) Brij Srivastava (INRIA), Marc Tommasi (INRIA), Nathalie Vauquier (INRIA)**
Reviewers	**Gerrit Klasen (ASCO), Askars Salimbajevs (TILDE)**
Submission date	**30/04/2020**
Due date	**30/04/2020**
Type[1]	**OTHER**
Dissemination level[2]	**PU**

Document history			
Date	**Version**	**Author(s)**	**Comments**
09/04/2020	0.1	David Adelani, Ali Davody, Brij Srivastava, Marc Tommasi, Nathalie Vauquier	Draft deliverable
20/04/2020	0.2	David Adelani, Ali Davody, Brij Srivastava, Marc Tommasi, Nathalie Vauquier	Revision based on the reviewers' comments
30/04/2020	1.0	Zaineb Chelly & Emmanuel Vincent	Final version integrating feedback and comments from the reviewers

[1] **R**: Report, **DEC:** Websites, patent filling, videos; **DEM:** Demonstrator, pilot, prototype; **ORDP:** Open Research Data Pilot; **ETHICS:** Ethics requirement. **OTHER:** Software Tools

[2] **PU:** Public; **CO:** Confidential, only for members of the consortium (including the Commission Services)

Document summary

This deliverable is devoted to the design, implementation, and evaluation of transformations focusing on deleting the citizen's[3] identity and words carrying critical information, and model learning. It extends the initial work that produced baselines of such transformations; which was delivered in August 2019[4]. All software components of this deliverable are available to the public on the COMPRISE git repository[5]. The proposed privacy-driven speech and text transformations will be integrated in the COMPRISE Software Development Kit (SDK) Client Library, while the proposed differentially private training tool implements a general training approach that can be used to develop specific training components to be integrated in the COMPRISE Cloud Platform. In this report, first, we recall in Section 1 the objectives of the work package, the choices that govern software development and the status reached at the beginning of the period covered by this deliverable. Then, we focus in Section 2 on the evaluation of the privacy obtained by our proposed approach. This is mainly a scientific exposition of the methodology we have followed. The outcomes of this study led to improvements of the baselines. They are presented in Section 3 together with experiments. A presentation of the new software library is given in Section 4.

[3] In this report we will use the terms "user" and "citizen" to refer to the person speaking to the dialogue system. From the GDPR point of view, the citizen is the "data subject".

[4] `https://www.compriseh2020.eu/files/2019/08/D2.1.pdf`

[5] `https://gitlab.inria.fr/comprise`

Contents

1 Context

Modern applications now allow the user's voice to be the main means of interaction with computers or smart objects. Technologies in voice interaction rely on machine learning models trained on user's data. This raises serious privacy concerns because voice is considered as biometric data and carries a lot of private and sensible information. One of the objectives of COMPRISE is to design a framework that enables the training and the use of the voice interaction tool in a private-by-design manner. In COMPRISE, the voice interaction chain consists of two branches: the operating branch and the training branch (see Figure 1). Privacy in the operating branch will be ensured by running all computations on the user's device or on a trusted personal server and sending only the information needed to deliver the required service provided by the app. COMPRISE will also focus on ensuring privacy in the training branch. To do so, in order to match the objective of Work Package 2, we introduce two innovations that complement each other : a new privacy-driven speech transformation and a new privacy-driven text transformation.

1.1 State-of-the-art after Deliverable D2.1

We have delivered in August 2019 a first version of the privacy-driven transformation tools, which was in the form of Deliverable D2.1[6]. Let us recall first the main achievements that were obtained at that stage.

The global architecture is depicted in Figure 1. We have represented the main tasks and the flow of information between them.

In the learning branch, the aim is to produce neutral voice and text data that will be used for further improvement of the speech-to-text (STT), spoken language understanding (SLU), and dialogue management components used in the COMPRISE system.

First the citizen voice is processed by an STT module to obtain the corresponding text marked with temporal positions. The text is then processed by a *text transformer* that removes sensitive words and expressions. The speech signal is also processed by a *voice transformer* to produce a neutral voice. Thanks to the temporal positions and the neutral text, a *secure voice builder* is able to reconstruct a new speech signal where sensitive patterns have been removed. So we provided the following software developments in the previous period:

`VoiceTransformer` objects have two main methods. The first one, `fit`, is used to compute internal parameters to fit the transformer to the user's voice. Indeed, some transformations need to be instantiated with specific features of the user's voice. It needs a few utterances as input and is used only once. Then `transform` performs a transformation to a target voice.

`TextTransformer` objects address two tasks: identifying the parts of the text to be transformed, and performing the actual transformation into a neutral text.

`SecureVoice_Builder` The `mask_words_in_speech` script in `SecureVoice_Builder` receives neutral voice, neutral text and temporal positions of each sensitive word to reconstruct a secured speech signal where sensitive words have been removed.

The temporal position of each word in the speech signal is provided by the STT module. However, for evaluation purposes, ground truth text can be used as input to the `TexTransformer`

[6] `https://www.compriseh2020.eu/files/2019/08/D2.1.pdf`

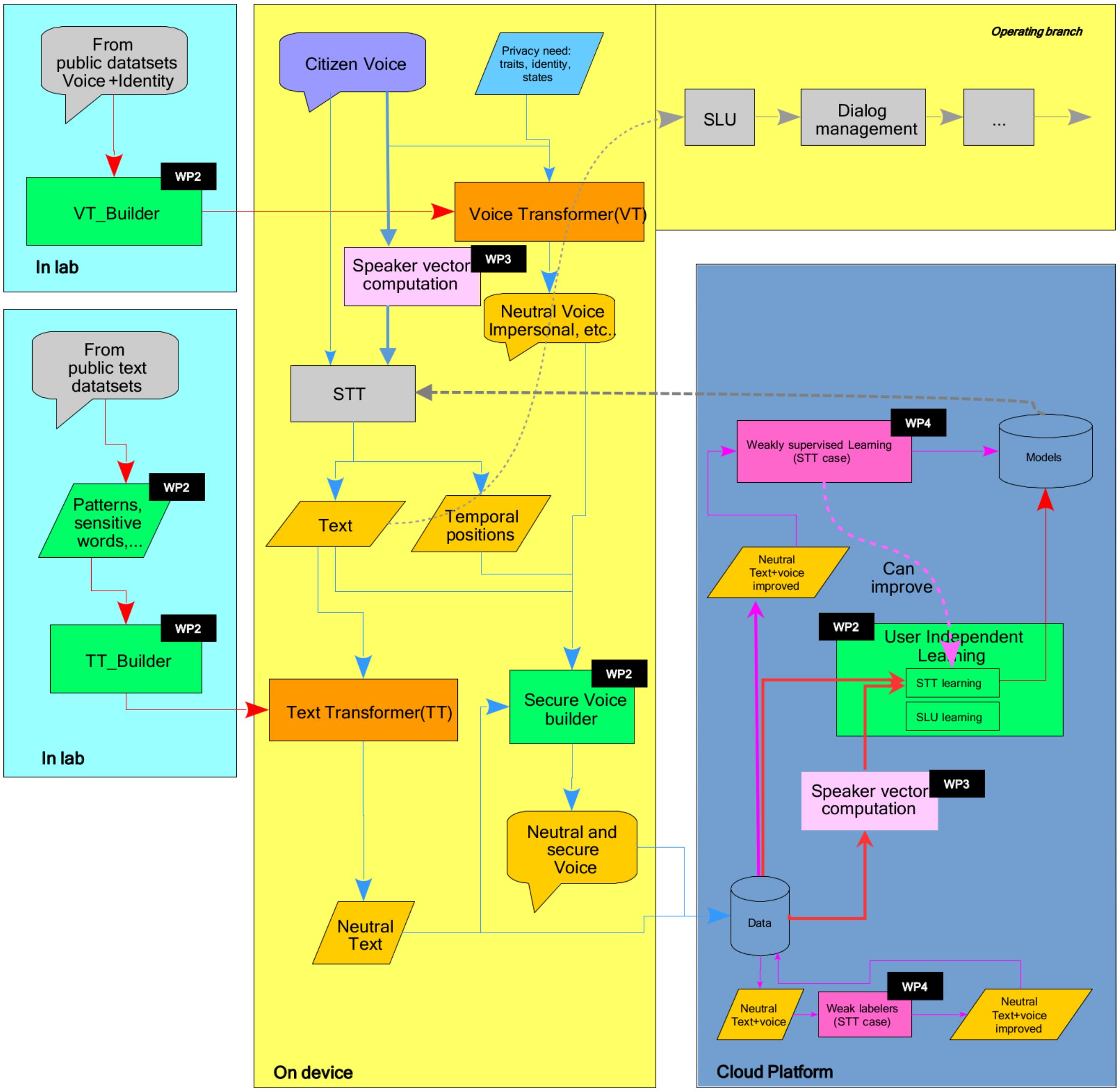

Figure 1: Overview of the global architecture of the learning branch, with interactions with other WPs.

instead of the STT text. In this case, the additional package align-s2t can be used to obtain the temporal positions of the words from the ground truth text in the speech signal.

The mask_words_in_speech script gives fine control over the type of words and named entities (NEs) that can be finally masked in the speech signal. Hence, SecureVoice_Builder also includes an optional helper script mask_words_in_text which produces neutral text with the same type of words and named entities that are masked in the speech signal. Again, this feature is meant for detailed evaluation purposes.

Two specific functions are producing the text and the voice transformers. They are represented in the blue frames on the left of the diagram.

VT_Builder takes as input a dataset containing speech data and speaker identities and produces a voice transformer.

TT_Builder is responsible for selecting the appropriate transformation strategy to use inside the TextTransformer and for continuously improving the models used by the latter. At the current stage, however, the TextTransformer uses fixed, external models only, pushing the need for a TT_Builder to a later stage in the project.

1.2 Contents

The software tools delivered in August were at an early stage and the evaluation of privacy and the quality of the privacy measures had not been studied in depth. During the period covered by this deliverable (M9–M17), we have addressed the problem of assessing the privacy resulting from the different methods proposed in our Voice and Text Transformers. Our study is reported in Section 2. We have identified the context of the usage of voice-enabled technologies as a rich source of information and a mid-term objective to improve our transformers. We have taken into account some background knowledge an attacker may have in the context of COMPRISE. We have rigorously studied different measures in the literature (e.g., in the speaker identification domain) and their relevance for the assessment of privacy. Finally, we have proposed a first differentially private approach to text transformation giving a more comprehensive trade-off between utility and privacy for text transformation.

Based on our observations, we have proposed improvements of our privacy driven transformations (see Section 3). Voice transformers have been improved and a new private training method for deep learning is presented. We have also started to address the problem of gender obfuscation, following our objective to take into account more traits. On the multilingual side, we have started to study the case of the Latvian language.

2 Evaluation of privacy

2.1 Privacy in speech technologies

Following the initial work done in the first period about categorisation of personal data, COMPRISE partner ROOT has proposed in [MJ20] an overview of how personal data recorded by voice-enabled systems can be identified through the categorisation of personal information and the analysis of context. ROOT has shown how these methods can be used to design private-by-design voice-based solutions that intend to neutralise personal data and information/words that may reveal private information.

Different sources of information are analysed: spoken content, speaker's voice, background sounds, and usage metadata (geolocation, application type, etc.). Each source is captured locally on the device and represents either a possible leak of private information or a way to improve the distinction between private or non private data.

The study reveals that the context of use at the moment a user interacts with the system is of major importance to disambiguate and identify personal data. Context can be related to usage, cultural, linguistic and other combinations of different elements. For instance, a hump for the British is a state of depression or annoyance, while for Americans it could mean making a vigorous effort. In that case, knowing either the cultural and linguistic context or even the usage context (e.g., a medical application) is clearly helpful. The paper depicts many situations where *contextualisation* is of major importance with respect to personal data.

In standard applications where data is processed in the Cloud by distrusted third parties, the exploitation of context information is of course a source of privacy leak. Hence, the paper also advocates for more decentralised processing, following the architecture and implementation choices made in COMPRISE.

This work enlightens the advantages of the choices made in the definition of the architecture of COMPRISE. It also gives an important direction of the future development that the project may follow.

2.2 Taking into account the attacker's knowledge

In [Sri+20], we investigate anonymisation methods based on voice conversion (VC). In contrast to prior work, we argue that various *linkage* attacks can be designed depending on the attackers' knowledge about the anonymisation scheme. We compare two frequency warping-based conversion methods presented in Deliverable D2.1 (VTLN and VoiceMask) and a deep learning based method (called *Disentangled Representation*) in three attack scenarios. The utility of converted speech is measured via the Word Error Rate (WER) achieved by STT, while privacy is assessed by the increase in Equal Error Rate (EER) achieved by state-of-the-art i-vector or x-vector based speaker verification. Our results show that voice conversion schemes are unable to effectively protect against an attacker that has extensive knowledge of the type of conversion and how it has been applied, but may provide some protection against less knowledgeable attackers.

As opposed to past studies that only considered weak attack scenarios where the attacker is unaware that an anonymisation method has been applied to the data, we consider different linkage attacks depending on the attacker's knowledge of the anonymisation method. At one end of the spectrum, an *Ignorant* attacker is unaware of the speech transformation being applied, while at the other end an *Informed* attacker can leverage complete knowledge of the transformation algorithm and its parameter values. A *Semi-Informed* attacker may know the voice transformation algorithm but not its parameter values. In our experiments, we evaluate three VC methods with different target speaker selection strategies in various attack scenarios to study unlinkability in the spirit of ISO/IEC 30136 standard [ISO17]. In each scenario, we assess how well each method protects the speaker identity against attackers that leverage state-of-the-art speaker verification techniques based on i-vectors [Deh+10] or x-vectors [Sny+18] to design linkage attacks. We also report the WER achieved by a state-of-the-art end-to-end STT system [Wat+18]. All experiments are conducted on the LIBRISPEECH dataset [Pan+15].

In this study, we consider that the VC function and the sets of possible parameter values are known to all users. Each user captures his/her voice on his/her device and applies a VC scheme locally before sending it to the Cloud. In the threat model we consider, an attacker accesses the converted utterances (called *trial* utterances) and performs a linkage attack to identify which ones are spoken by a particular user. To this end, we assume that the attacker also has access to a small amount of *enrollment* speech from this user (and potentially some additional public resources, such as benchmark speech processing datasets to train generic speaker models).

We apply three VC methods mainly parameterised by the choice of a target speaker.

- VoiceMask ([Qia+18]) is a frequency warping method based on the composition of a log-bilinear function and a quadratic function, expressed by two parameters (called α and β).
- VTLN-based VC ([SN03]) represents each speaker by a set of centroid spectra extracted using the CheapTrick [Mor15] algorithm for k pseudo-phonetic classes. Classes of a source speaker are warped using a power function to the classes of a target speaker.
- *Disentangled Representation* based VC ([CL19; UVL17]) is based on a neural network transformation and uses a *speaker encoder* and a *content encoder* to separate the factors of variation corresponding to speaker and content information.

We consider three possible target selection strategies for the three VC methods above, which can be seen as key ingredients of a "private-by-design" speech processing system (see Figure 2).

- In strategy *const*, the VC function is constant across all users and all utterances. This means choosing a unique target speaker and, in the case of VoiceMask, fixed values for α and β.
- In strategy *perm*, the conversion parameters are chosen at random once by each user. In other words, when a user downloads the VC module on his/her device, he/she selects a personal target speaker and, in the case of VoiceMask, personal random values for α and β.
- Finally, in the *random* strategy, each time a user applies VC to an utterance, a random set of parameters is drawn, i.e., a random target speaker is selected and, in the case of VoiceMask, random values are drawn for α and β.[7]

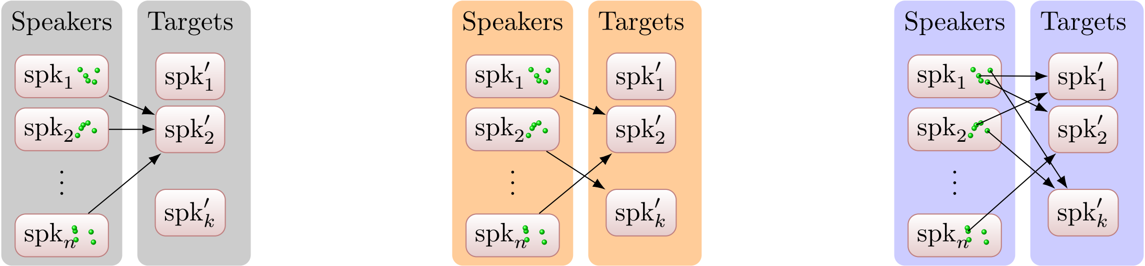

Figure 2: Strategies *const*, *perm* and *random*. Utterances are small green balls, arrows show the mapping to select a target speaker.

[7] The random choice of the parameters is a key component to achieve anonymisation in the VoiceMask method, therefore only *random* strategy was evaluated.

We have implemented several attackers depending on the choice of the VC algorithm and the target selection strategy as well as the extent of the attacker's knowledge (*Ignorant*, *Semi-Informed* or *Informed*, see Figure 3). Our *Ignorant* attacker is unaware of the VC step: he/she simply trains x-vector/i-vector models on the untransformed training set, and applies them to the untransformed enrollment set. Our *Semi-Informed* attacker knows the VC algorithm and the target selection strategy (*const*, *perm* or *random*) but not the particular choices of targets. He/she applies this strategy to the training and enrollment sets by drawing random target speakers from the subset of 100 target speakers used by the VC method (we assume that the value of k in VTLN is known to the attacker). As a result, the training and enrollment data are converted in a similar way as the trial data, but the target speaker associated with every speaker in the enrollment set is typically different from that associated with the same speaker in the converted trial set. Finally, our *Informed* attacker has access to the actual VC models and target choices used to anonymise the trial set, so it converts the training and enrollment sets accordingly.

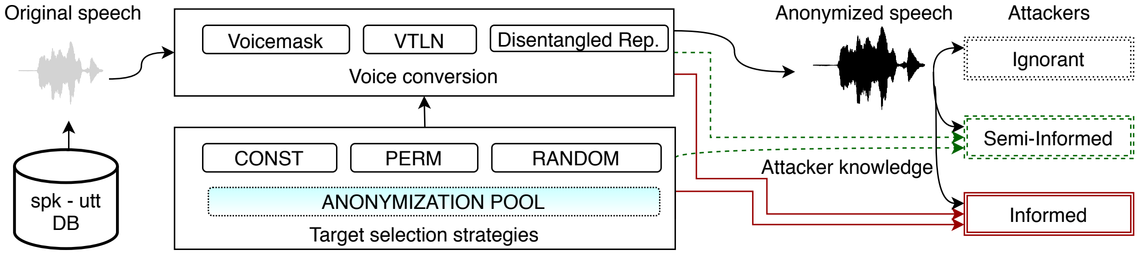

Figure 3: Schema for speaker anonymisation and three types of linkage functions designed as attackers based on increasing degrees of knowledge. The anonymised speech is observed by the attackers (black arrows). The green dotted arrows indicate partial knowledge while red solid arrows indicates full knowledge.

In the Comprise setting, we can think that an *Ignorant* attacker is unaware that speech has undergone VC based transformation, the *Semi-Informed* attacker has studied the code of the VC method employed by the user, and the *Informed* attacker has hacked the user's device and can get internal values at any time.

Results We first train and apply the STT and speaker verification systems on the original (untransformed) data for baseline performance. The EER is used to measure privacy[8] and the WER is used to measure the utility of the data after transformation. We obtain an EER of 4.61% and 4.31% for i-vector and x-vector based speaker verification, respectively, and a WER of 9.4% for STT. Table 1 gives the WER obtained for each VC method, which we use as a proxy for the usefulness of the converted speech. Note that there is no difference between converted data in different attack scenarios, hence the WER does not depend on the attacker. VoiceMask and VTLN-based VC achieve reasonable WER compared to the untransformed data, while the disentangled representation based VC yields an unreasonably high WER.

Tables 2 and 3 present the EER for x-vector and i-vector based speaker verification for the three attackers and the various VC methods and target selection strategies. The *Informed* attacker

[8] Other measures of privacy are evaluated in the following section.

Table 1: WER (%) achieved using end-to-end STT.

	VoiceMask	**VTLN-based VC**			**Disentangl.-based VC**		
Subset ↓ / Strategies →	*random*	*const*	*perm*	*random*	*const*	*perm*	*random*
test-clean	18.1	19.8	18.4	15.9	41.5	23.7	115.1

Table 2: EER (%) achieved using x-vector based speaker verification.

	VoiceMask	**VTLN-based VC**			**Disentangl.-based VC**		
Attackers ↓ / Strategies →	*random*	*const*	*perm*	*random*	*const*	*perm*	*random*
Informed	5.01	4.71	3.91	6.32	4.71	0.20	5.52
Semi-Informed	-	12.84	23.37	6.32	13.64	43.03	5.42
Ignorant	28.69	24.27	30.99	27.38	27.68	32.20	30.59

Table 3: EER (%) achieved using i-vector based speaker verification.

	VoiceMask	**VTLN-based VC**			**Disentangl.-based VC**		
Attackers ↓ / Strategies →	*random*	*const*	*perm*	*random*	*const*	*perm*	*random*
Informed	8.22	6.22	10.23	9.84	4.71	0.20	11.03
Semi-Informed	-	18.25	31.49	18.76	15.65	43.93	10.53
Ignorant	50.55	26.08	49.15	49.15	49.95	47.74	49.85

achieves similar or even slightly lower EER than the baseline. This indicates that, when the attacker has complete knowledge of the VC scheme and target speaker mapping, none of the VC methods is able to protect the speaker identity. Fortunately, an attacker with such complete knowledge is unrealistic in practical cases, including in Comprise.

For the more realistic *Semi-Informed* attacker, we observe that strategy *perm* is quite effective in protecting privacy and shows the highest gains in EER. This is due to the fact that the target speaker in the enrollment data may not be the same as the one in trial, hence greater confusion is induced during inference. We also notice that strategy *random* is not much affected by the change of speaker mapping, which is intuitive because in this case the utterances are already being mapped randomly to different speakers. Such mapping would be ineffective due to averaging of randomness. Strategy *const* is also slightly affected by the change of mapping but the effect is not as significant as the *perm* strategy.

Conclusion We investigated the use of VC methods to protect the privacy of speakers by concealing their identity. We formally defined target speaker selection strategies and linkage attack scenarios based on the knowledge of an attacker. Our experimental results indicate that both aspects play an important role in the strength of the protection. Simple methods such as VTLN-based VC with appropriate target selection strategy can provide reasonable (but not complete) protection against linkage attacks with partial knowledge. Section 3 reports new contributions to further improve the level of protection.

2.3 Linkability and other privacy measures

The classical measure used in many studies in speaker verification is the EER. A speaker verification system performs a binary decision to detect whether a given speaker in the enrollment set is the same (genuine) or not (impostor) as the one who uttered a given trial utterance. The genuine vs. impostor decision is achieved by comparing a similarity score with a decision threshold. Modifying this threshold modifies the false positive and false negative rates, and the EER corresponds to the value where those two rates are equal. But is EER a good measure for privacy? This question has motivated a study where we have analysed and compared several alternative measures that could be more relevant in the context of privacy.

We have considered a representation of speech data or a representation of speakers by means of deep neural network based speaker embeddings called x-vectors [Sny+18]. It is known that x-vectors perform very well for speaker identification and verification, but also for many other speech and speaker features [Raj+19]. Following the study in Section 2.2, we have performed our analysis using the *perm*, *const*, *random* strategies for the three voice conversion methods VTLN, VoiceMask and *Disentangled Representation* and a fourth method detailed in Section 3.1, according to the different levels of knowledge of the attacker.

Indistinguishability and re-identification A first approach was to study the embedding space of x-vectors. We started with the distinguishability of the embeddings of utterances of the same speaker. We measured the proportion of utterances that have at most a certain precision at top-k, where the top-k represents the k-closest utterances according to the Euclidean distance. In summary, we observed a larger indistinguishability in the *perm* case, however, we noticed that in any case almost all utterances have a precision at 100% at top-1. The same observation was made in re-identification experiments where essentially, we measured the anonymity size, that is the rank of a correct identification. In that case, identification is based on the probability of being the same speaker, using a notion of similarity computed by a normalised Gaussian kernel.

Linkability A second approach was to study linkability. By definition, linkability means that an attacker can sufficiently distinguish whether two items (in our case utterances) are related or not. Here, the relation is formally given by a linkage function that produces a score s that informs on the strength of the relation between the two items (in our case, it informs on the similarity between the speakers for those utterances). We study $D_{\leftrightarrow}(s)$ as the measure of how much a score s tells us that two items are linkable (i.e., belong to the same speaker). The computation follows [Gom+17] and $D_{\leftrightarrow}(s)$ is close to 0 when the score cannot distinguish mated (i.e., same) and non-mated (i.e., distinct) speakers. Also, rather than providing a single measure, this approach offers a full range of evaluations for all score distributions. We conducted experiments to study the linkability between trial and enrollment utterances on the one hand and among trial utterances on the other hand, where the score s was computed by Probabilistic Linear Discriminant Analysis (PLDA) [9] of x-vectors. In summary, the results are consistent with the above.

Cost sensitive measure In certain applications, the cost of a false negative differs from the cost of a false positive. The $C_{\text{llr}}^{\text{min}}$ measure provides an application-independent measure by integrating over all costs. The decision is based on the log-likelihood ratio of the posterior probability of having

[9] We have also shown that the PLDA score is more relevant as a linkage function than Euclidean or cosine distance when considering distances between x-vectors.

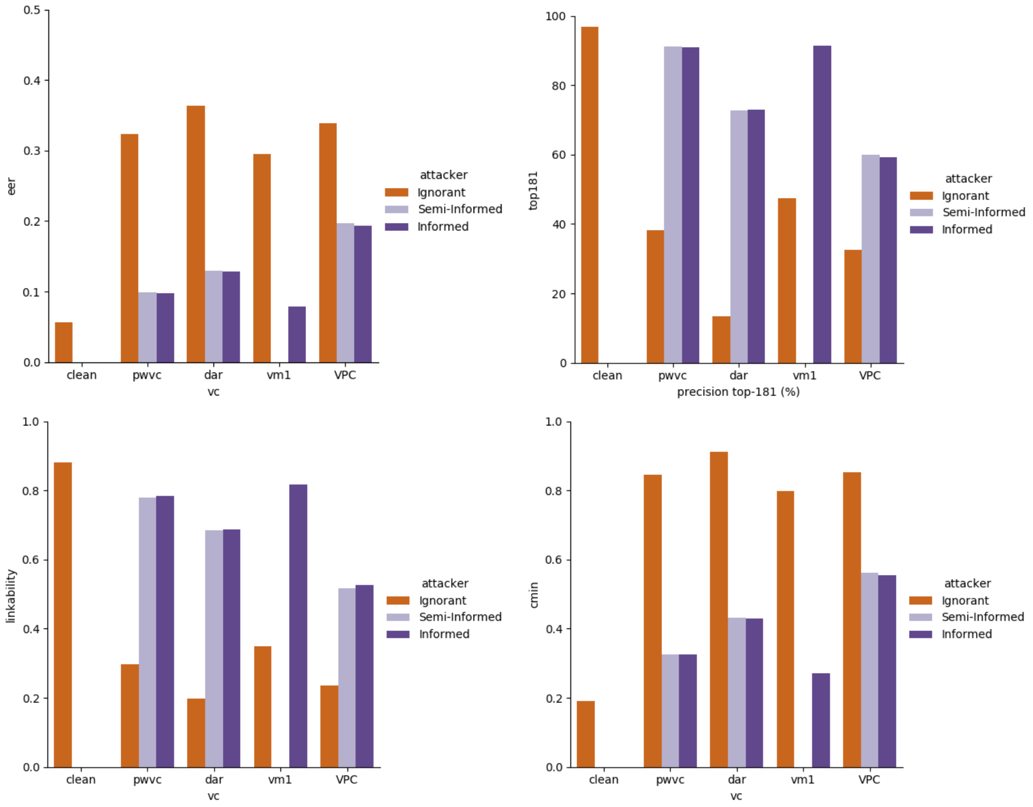

Figure 4: Experimental comparison of four privacy measures: EER (top left), top-k ($k = 180$, top right), linkability (bottom left) and $C_{\text{llr}}^{\text{min}}$ (bottom right) on the different scenarios and transformers. The four transformation methods are denoted as pwvc (VTLN), dar (disentangled representation), vm1 (VoiceMask), and VPC (see Section 3.1).

a score given a mated or non-mated couple of items and the decision threshold takes the costs into account.

Summary All the measures we have tested give consistent observations. This is illustrated in Figure 4. Privacy is well preserved in the ignorant case. In the semi-informed case, the considered VTLN, disentangled representation based VC, and VoiceMask transformations do increase the level of privacy compared to original untransformed data, but the results do not allow us to claim complete protection in the considered experimental setting. The disentangled representation based VC approach provides the best level of protection but it seems unusable because the resulting speech intelligibility and WER are poor. To address this issue, we developed an improved transformation, which we present in more detail in Section 3. As an initial preview of the results of this new transformation, the reader can observe its performance in the last bars (labelled VPC) of the histograms in Figure 4.

Crucially, all experiments were done on a small set of speakers, which implies a very hard anonymization problem. Of course, the more speakers in the trial set, the greater the chance of reaching an acceptable level of privacy. The results should therefore be considered as *relative* results allowing us to compare different transformation schemes and not as absolute privacy guarantees.

Additionally, this large set of experiments on different measures gave us some insights on measuring privacy in real cases. We advocate for the use of $C_{\mathrm{llr}}^{\mathrm{min}}$ and Linkability. Indeed, even though the results in our case are consistent with EER, we observe that they are more grounded, and provide more information by taking into account a large spectrum of calibrations and applications. The conducted experiments also allowed us to conclude that PLDA is more meaningful than the Euclidean or the cosine distance to compute distances in the x-vector space.

2.4 Evaluation of differentially private text transformations

In Deliverable D2.1, we presented different ways of hiding private information in dialogue transcripts by replacing user sensitive words with placeholders or their surrogates of the same type [THG04]. We already showed that same-type replacements are more appropriate for Spoken Language Understanding (SLU) tasks because they led to the least drop in performance after text transformation. Here, we formulate different hiding strategies in terms of *differential privacy* and compare their performance-privacy trade-off. We verify this by applying the state-of-the-art deep learning model to train an SLU task where the word features are obtained from BERT embeddings [Dev+18] before they are passed into a Bidirectional Long Short-Term Memory neural network with a Conditional Random Field layer (BiLSTM-CRF) [HXY15] for named entity recognition (NER).

Definition 2.1. *(Differential Privacy). A randomised algorithm $\mathcal{M}$ is (ε, δ) private with domain $\mathcal{D}$ if for any measurable set $S \in Range(\mathcal{M})$ and for all neighbouring datasets D_1 and D_2 differing on at most one data point, we have*

$$\Pr[\mathcal{M}(D_1) \in S] \leq \exp(\varepsilon) \Pr[\mathcal{M}(D_2) \in S] + \delta \tag{1}$$

An (ε, δ) differentially private algorithm guarantees that the absolute value of information leakage is bounded by ε with a probability of at least $1 - \delta$. Therefore ε controls the level of privacy protection and so is called privacy loss. The lower ε, the greater the privacy.

We proceed by introducing the privacy loss ε into text replacement methods. To do so, we employ a similar technique to the *randomised response* [War65]. It was developed in social science to guarantee a degree of *plausible deniability* to any individual while collecting useful statistics from a population. We consider a text replacement strategy $\pi : \mathcal{T} \longrightarrow \mathcal{T}$, where $\pi(t|t')$ is the probability of replacing t' in the original text with t. Let p be the probability that we apply the text replacement strategy to a given text. We randomise π further as shown in this algorithm.

Algorithm 1: Randomised response text replacement.

```
Input: dataset $\mathcal{D}$, text replacement policy $\pi$, probability parameter $p$.
for $t'$ in sensitive data do
    $r \sim U(0,1)$ if $r \leq p$ then
        replace $t'$ with $t \sim \pi(t|t')$
    end
end
```

Lemma 2.1. *If text replacement strategy π in Algorithm 1 is independent from source words, i.e., $\pi(t|t') = \pi(t)$, then Algorithm 1 is $(\varepsilon, 0)$ differentially private where:*

$$\varepsilon = \max_t \log \frac{1 - p + p\,\pi(t)}{p\,\pi(t)}. \tag{2}$$

To prove this lemma, we consider two neighbouring datasets D_1 and D_2 — for instance D_2 is the dataset where one token t' has been replaced by t while D_1 has a token t' that can either be replaced or unchanged, which implies that the datasets differ by at most one element — and we can compute the privacy loss as

$$\varepsilon = \log \frac{\Pr[\mathcal{M}(D_1) = t]}{\Pr[\mathcal{M}(D_2) = t]} \tag{3}$$

where t is the observed word in the replaced text. Note that replacement policies are independent from original words. So, if the original word in the two datasets is not equal to t then we have:

$$\frac{\Pr[\mathcal{M}(D_1) = t]}{\Pr[\mathcal{M}(D_2) = t]} = \frac{p\,\pi(t|D_1)}{p\,\pi(t|D_2)} = 1. \tag{4}$$

On the other hand if the original word in one of the datasets say D_1 is equal to t then:

$$\frac{\Pr[\mathcal{M}(D_1) = t]}{\Pr[\mathcal{M}(D_2) = t]} = \frac{1 - p + p\,\pi(t)}{p\,\pi(t)}. \tag{5}$$

Combining the two cases, we obtain Equation (2). It is worth mentioning that naive text replacement corresponds to $p = 1$ and therefore has a zero privacy loss. Since the text transformation depends on the automatic identification of privacy sensitive entities, the quality of the identification can be quantified by p. If all sensitive information are correctly identified, $p = 1$ and $\varepsilon = 0$, otherwise, $p < 1$ and $\varepsilon > 0$, and in both cases, we can estimate the privacy loss due to the imperfect identification of privacy sensitive entities. Also, we can tune the parameter p to get a better performance-privacy trade-off depending on the SLU task.

Experimental Setup We evaluated the performance of the proposed text transformation on the Verbmobil [Wah00] NER dataset (introduced in Deliverable D2.1) using the Flair [ABV18] implementation[10]. The NER dataset consists of 19,151 training, 2,846 development, and 5,230 test sentences with 5 private token categories or named entities: PER (Personal names), ORG (organisation), LOC (location), DATE, and TIME. The hyper-parameters of the model are: a 768-dimensional embedding layer (for BERT), a BiLSTM layer size of 256 in each direction, an adaptive learning rate of 0.1 (the learning is halved when development data loss does not improve), a mini-batch size of 32, and a maximum number of epochs of 50.

Results We show that the performance of replacing tokens by surrogates of the same type can be improved by tuning the parameter p described in Equation (2). For example, by setting $p = 0.9$, we can improve the F1-score by around 2% for the same-type replacement as shown in Table 4, thus, we can control the privacy-utility trade-off. For the same-type replacement experiments, we replace a named entity t' by another named entity t of the same class based on their relative frequency distribution $\pi(t)$ in the corpus. In Table 4, we evaluate the performance of the same-type replacement strategy as a function of p based on the differential privacy loss ε computed according to Equation (2) which is the maximum of the privacy losses of all named entity classes.

[10] `https://github.com/zalandoresearch/flair`

Table 4: Differential privacy guarantees and utility achieved on the Verbmobil corpus by the text replacement strategy in Algorithm 1. With a probability $p = 0$ of replacing a sensitive attribute, the level of privacy protection is identical to the original data. When $p = 1$, we apply the word-by-word replacement strategy, i.e., same-type replacement.

p	0	0.5	0.9	1.0
ε	∞	8.95	6.75	0.0
F1-score	89.1	86.4	81.7	79.8

3 Improvement of privacy driven transformations

3.1 New privacy driven speech transformation

Comprise members are involved in the organisation of the Voice Privacy Challenge (VPC). [11] The challenge aims to develop anonymisation solutions which suppress personally identifiable information contained within speech signals. At the same time, solutions should preserve linguistic content and speech quality/naturalness. The challenge will conclude with a session/event held in conjunction with Interspeech 2020 at which challenge results will be made publicly available.

We have joined efforts with Avignon Université, EURECOM, and NII to produce a clear definition of an attacker scenario, evaluation metrics and baselines for speech privacy. In this section, we present a baseline that is now also part of the Comprise library as a new voice transformer method. The VPC system is inspired from the speaker anonymisation method proposed in [Fan+19] and shown in Figure 5. Anonymisation is performed in three steps:

- **Step 1: Feature extraction:** extract the speaker x-vector [Sny+18], and the fundamental frequency (F0) and bottleneck (BN) features from the original audio waveform.
- **Step 2: X-vector anonymisation:** anonymise the x-vector of the source speaker using an external pool of speakers.

[11] https://www.voiceprivacychallenge.org/.

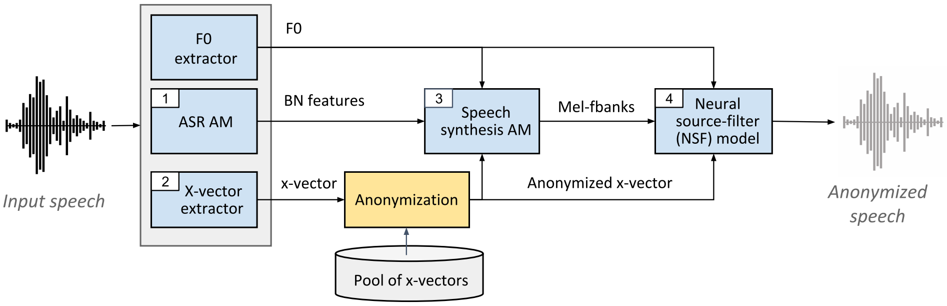

Figure 5: VPC anonymisation system.

- **Step 3: Speech synthesis:** synthesise the speech waveform from the anonymised x-vector and the original BN and F0 features using an acoustic model and a neural waveform model.

In order to implement these steps, four different models are required as shown in Figure 5. Details for training these components are presented in Table 5. More details about the implementation can be found in the COMPRISE git repository and in the challenge repository[12].

In *Step 1*, to extract BN features, an STT acoustic model (AM) is trained. We assume that these BN features represent the linguistic content of the speech signal. The STT AM has a factorised time delay neural network (TDNN-F) model architecture [Pov+18; PPK15] and is trained using the Kaldi toolkit [Pov+11]. To encode speaker information, an x-vector extractor with a TDNN model topology is also trained using Kaldi.

In *Step 2*, for a given source speaker, a new anonymised x-vector is computed by averaging a set of candidate x-vectors from the speaker pool. The candidate x-vectors for averaging are chosen in two steps. First, for a given x-vector, the N farthest candidates in the speaker pool are selected. Second, a smaller subset of N^* x-vector candidates from this set are chosen randomly[13].

In *Step 3*, two modules are used to generate the speech waveform: a speech synthesis AM that generates Mel-filterbank features given the F0, the BN features, and the anonymised x-vector, and a neural source-filter (NSF) waveform model [WTY19] that produces a speech waveform given the

[12] `https://github.com/Voice-Privacy-Challenge/Voice-Privacy-Challenge-2020`

[13] By default, the following parameter values are used: $N = 200$ and $N^* = 100$; and PLDA is used as the distance between x-vectors.

Table 5: VPC anonymisation system: models and corpora. The model indexes are the same as in Figure 5. Superscript numbers represent feature dimensions.

#	Model	Description	Output features	Training dataset
1	ASR (STT) AM	TDNN-F Input: MFCC^{40} + i-vectors^{100} 17 TDNN-F hidden layers Output: 6032 triphone ids LF-MMI and CE criteria	BN^{256} features extracted from the final hidden layer	LIBRISPEECH: train-clean-100 train-other-500
2	X-vector extractor	TDNN Input: MFCC^{30} 7 hidden layers + 1 stats pooling layer Output: 7232 speaker ids CE criterion	speaker x-vectors^{512}	VOXCELEB: 1, 2
3	Speech synthesis AM	Autoregressive (AR) network Input: $F0^1$+ BN^{256}+x-vectors^{512} FF * 2 + BLSTM + AR + LSTM * 2 + highway-postnet MSE criterion	Mel-filterbanks^{80}	LIBRITTS: train-clean-100
4	NSF model	sinc1-h-NSF in [WTY19] Input: $F0^1$ + Mel-fbanks^{80} + x-vectors^{512} STFT criterion	speech waveform	LIBRITTS: train-clean-100
Pool of speaker x-vectors				LIBRITTS: train-other-500

Table 6: Speaker verification performance achieved by the VPC transformation.

Dataset	Gender	Anonymization		Development		Test	
		Enroll	Trial	EER (%)	$\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$	EER (%)	$\mathrm{C}_{\mathrm{llr}}^{\mathrm{min}}$
LibriSpeech	Female	original	original	8.67	0.304	7.66	0.183
			anonymized	50.28	0.997	48.54	0.996
		anonymized		35.09	0.876	29.74	0.797
	Male	original	original	1.24	0.034	1.11	0.041
			anonymized	58.39	0.998	53.23	0.999
		anonymized		29.66	0.806	32.52	0.835
VCTK	Female	original	original	2.86	0.100	4.89	0.169
			anonymized	50.03	0.988	48.87	0.999
		anonymized		29.48	0.814	34.21	0.884
	Male	original	original	1.44	0.052	2.07	0.072
			anonymized	55.33	1.000	53.73	1.000
		anonymized		26.10	0.756	25.83	0.743

Table 7: STT performance achieved by the VPC transformation.

Dataset	Anonymization	Dev. WER (%)	Test WER (%)
LibriSpeech	original	3.83	4.14
	anonymized	6.50	6.77
VCTK	original	10.79	12.81
	anonymized	15.50	15.53

F0, the anonymised x-vector, and the generated Mel-filterbanks. Both models are trained on the same corpus (*LibriTTS-train-clean-100*).

The resulting privacy-utility trade-off is evaluated on two different datasets: LibriSpeech and VCTK [VYM19]. Speaker verification results are provided in Table 6, and STT results in Table 7. Comparing the LibriSpeech results in Table 6 with those in Table 2 and looking at Figure 4, we see that the level of privacy protection resulting from the VPC transformation is comparable to that achieved by the VTLN, disentangled representation based VC, and VoiceMask transformations in the case of an ignorant attacker, but it is much better in the case of a semi-informed attacker, which translates into greater privacy guarantees for users in practice. Also, the VPC transformation results in smaller distortion of the speech signal, and therefore better STT performance.

Here also, all experiments have been conducted on a small set of speakers. The obtained EER values should be considered as relative results allowing us to assess the superiority of the VPC transformation, rather than as absolute privacy guarantees.

3.2 Differentially private training for deep learning

Deep neural networks trained with a massive amount of data have enjoyed great success in a wide variety of domains. Often, these datasets contain sensitive and highly private information such as medical records of patients. Differential privacy [DR+14] is a well-known mechanism for training machine learning models to prevent exposure of private information in the training dataset. The core idea is randomising the non-private training algorithms by injecting calibrated noise. Differential privacy has been integrated into deep learning in [SS15; Aba+16] to tackle privacy issues. The proposed method in [Aba+16] is based on clipping and adding random noise to the gradient at every iteration of stochastic gradient descent (SGD). This differentially private SGD (DPSGD) technique along with the *moments accountant* method for tracing the privacy loss has enabled training of deep neural network under modest privacy losses with a manageable reduction in the model's test accuracy. Nevertheless, DPSGD results in a large drop in accuracy under a low privacy loss.

To address this problem, we first study how injecting random noise degrades a non-differentially private deep learning model and augment the model with normalisation layers (e.g., batch normalisation or layer normalisation) to increase its robustness to noise. We hypothesise that this increased robustness is a consequence of the scale invariance property of normalisation operators. Building on this observation, we propose a new algorithmic technique for training deep neural networks under very low privacy losses by sampling weights from Gaussian distributions and utilising batch or layer normalisation to limit the drop in performance. We refer to this new algorithm as *Scale Invariant DPSGD (SI-DPSGD).*

Noise and Normalisation A crucial part of any privacy-preserving learning mechanism is adding noise to the training procedure. We investigate how random Gaussian noise affects the performance of a neural network in the presence of normalisation layers. More specifically, we sample the weights from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with learnable mean parameters μ and constant variance σ^2. This approach is very similar to variational Bayesian learning of neural networks [Blu+15], where weights are represented by probability distributions rather than a deterministic value. Unlike the Bayesian approach, whose goal is to learn the true posterior distribution of the weights given the training data, here the noise is introduced via an ad-hoc distribution.

Batch normalisation [IS15] and layer normalisation [BKH16] are introduced to speed up training by regularising neurons dynamics via mean and variance statistics and reducing variance in the input to each node. Normalisation techniques in combination with other architecture innovations like residual connections [He+16] make training of very deep networks feasible. Batch and layer normalisation both ensure zero mean and unit variance in the output of a layer but using different statistics. Batch normalisation (BN) calculates the mean and variance statistics across samples in a mini-batch for each neuron independently, while layer normalisation (LN) standardises each summed input to a node utilising the statistics over all hidden units.

To test our hypothesis, we add normalisation layers to various deep neural netork models and compare them against the unnormalised baselines. We train some standard fully-connected as well as convolutional neural networks with noisy weights on MNIST [LeC+98a]. MNIST is a common dataset for the assessment of differential privacy properties of deep learning models [Aba+16]. In particular, we examine LeNet-300-100 and LeNet-5 [LeC+98b] and variants of ResNet [He+16] and VGG [SZ14]. All models are implemented in Pytorch [Pas+19] and trained with the Adam optimiser [KB14]. We train each model with different levels of noise $\sigma = \{0, 0.01, 0.1, 1, 2\}$. Hyperparameters are tuned by grid-search for each value of noise separately using 8% of training data as the held-out validation set.

Table 8 shows the accuracy of augmented models on the MNIST test dataset averaged on ten runs against unnormalised baselines, trained with the reparameterisation trick. The BN/LN prefixes in this table denote models that are obtained by adding batch normalisation or layer normalisation layers to the original architectures, respectively. It is evident from this experiment

Table 8: MNIST test-set accuracy (%) ± standard error (%) achieved by different models when injecting noise to the weights.

Model	**Noise Level (σ)**				
	0	0.01	0.1	1	2
LeNet-300-100	98.20±0.07	97.70±0.30	96.98±0.12	Random	Random
BN-LeNet-300-100	98.20±0.10	98.10±0.10	98.07±0.11	98.07±0.12	98.13±0.08
LN-LeNet-300-100	98.04±0.16	98.00±0.10	98.08±0.14	98.04±0.09	98.03±0.17
LeNet-5	99.20±0.02	98.94±0.07	98.40±0.03	Random	Random
BN-LeNet-5	99.20±0.08	99.21±0.05	99.18±0.06	99.24±0.04	99.25±0.07
LN-LeNet-5	99.16±0.08	99.14±0.06	99.13±0.07	99.21±0.05	99.19±0.07

that all augmented models are tolerant to noise while the baselines are not. Indeed, the accuracy of augmented models does not change for a large range of noise levels thanks to the scale invariance property of the networks. That is, scale invariance keeps the output of the model intact when noise is added to the weights and protects the network. On the other hand, baseline models are very sensitive to small weight perturbations. Notably, disturbing the weights by a small noise in the order of $\sigma = 0.3$ results in a complete drop in performance down to that of random prediction.

We observe a similar result on a natural language text classification task. We trained a BiLSTM model with one dense layer (DL) with/without layer normalisation (LN) on the AG News corpus [14], a popular text classification dataset with 4 categories of news: *World*, *Sports*, *Business*, and *Sci/Tech*. Each class has 30,000 training examples and 1,900 test examples. In total, the dataset consists of 120,000 training examples and 7,600 test examples. We further split the training examples into training/validation sets where we use 96,000 examples for training the models and 24,000 for validation (e.g., early stopping and learning rate tuning). We trained the models with different noise levels σ for the same number of epochs (i.e., 25). The larger the noise, the more epochs are needed to maintain the accuracy of the baseline model. Our results on the language data in Table 9 suggest that layer normalisation makes the BiLSTM-DL model robust to noise with minimal drop in accuracy ($1-7\%$), unlike the BiLSTM-DL model without layer normalisation that exhibits a large drop in accuracy (65%) with a noise level $\sigma = 2.0$. Our findings enable us to conclude that the LSTM and CNN architectures are robust to noise when equipped with layer and batch normalisation.

DPSGD Differential privacy has been integrated into deep learning in [SS15] and subsequently in [Aba+16] for the setting where the adversary has access to the network architecture and the learned weights, $f(\boldsymbol{\theta}^*, .)$. In particular, [Aba+16] preserves privacy by adding noise to the SGD updates, leading to differentially private SGD (DPSGD). More precisely, to make the learning private, the authors of [Aba+16] update the weights at each training iteration as

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \frac{\eta}{L}\left(\sum_{i=1}^{L} \nabla\mathcal{L}(\boldsymbol{\theta}_t, x_i) + r\right), \tag{6}$$

[14] `http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`

Table 9: AG News test accuracy with noisy weights for a variety of models. We see the same pattern as the vision dataset where models augmented with layer normalisation are very robust against noise.

Model	**Noise Level (σ)**				
	0	0.01	0.1	1	2
BiLSTM-DL	89.34%	89.57%	89.01%	66.32%	24.76%
LN-BiLSTM-DL	89.34%	88.87%	88.62%	85.74%	82.41%

where $\mathcal{L}$ is the loss function, η is the learning rate and r is sampled according to Gaussian distribution $\mathcal{N}(0, \sigma^2)$. To control the influence of training samples on the parameters, the gradients are clipped by l_2-norm:

$$\pi(g_i) = g_i . \min\left(1, \frac{C}{\|g_i\|_2}\right), \tag{7}$$

where g_i is the gradient corresponding to the i-th sample and C is the clipping factor. It has been shown in [Aba+16] that each step of DPSGD is (ε, δ)-differential private once we tune the noise $\sigma = C\, z$ as $z = \frac{\sqrt{2 \ln \frac{1.25}{\delta}}}{\varepsilon}$.

SI-DPSGD Here, we develop a different approach for training deep neural networks while preserving data privacy. Our approach deviates from DPSGD by making a major change in the representation of weights, i.e., we sample the weights from a normal distribution with learnable mean parameters $\boldsymbol{\mu}_t$ and a fixed variance σ^2 given by the desired privacy loss. Furthermore, we utilise the normalisation layers to retain the model performance. Algorithm 2 outlines the steps of SI-DPSGD for differentially private training of networks augmented with layer normalisation layers. We have also implemented the SI-DPSGD algorithm for batch normalisation for vision datasets (MNIST and CIFAR-10) with slightly better results than layer normalisation.

Consider Algorithm 2, the first step is to initialise the mean parameters and weights randomly. Then, at each iteration, we bound the influence of each individual sample on the gradients. To do so, we follow the same strategy as in DPSGD [Aba+16], i.e., we clip the gradients in l_2 norm using a clipping threshold C as in Equation (7). We, then, update the mean parameters $\boldsymbol{\mu}_t$ by SGD using truncated gradients. Finally, to preserve privacy, we sample the weights from the normal distribution $\mathcal{N}(\boldsymbol{\mu}_{t+1}, \frac{\eta_t^2}{L^2} C^2 z^2)$, with updated mean values and the variance corresponding to the privacy loss for each iteration. These sampled weights, in turn, will be used in the next forward and backward pass to compute the loss and gradients. Using standard arguments, it can be shown that each iteration of Algorithm 2 is (ε, δ)-differentially private. It is important to note that the mean parameters $\boldsymbol{\mu}_t$ are not protected by this mechanism and should not be revealed to the adversary as we do not add noise to the gradients.

Result We report the results of our method and compare them with existing differentially private mechanisms. All models, as well as DPSGD, have been implemented in PyTorch [Pas+17]. To track the privacy loss over the whole training procedure, we employ the Rnyi differential privacy technique developed in [Mir17]. It provides a tighter bound on the privacy loss comparing with the

Algorithm 2: Scale invariant DPSGD with layer normalisation

```
Input: dataset $\mathcal{D} = \{(x_1, y_1), \cdots\}$ of size $N$, loss function $\mathcal{L}(\boldsymbol{\theta}, .)$, learning rate $\eta_t$, noise multiplier $z$, sample size $L$, gradient norm bound $C$ and $T$ iterations.
• Initialise mean parameters $\boldsymbol{\mu}_0$ randomly.
• Set weights as $\boldsymbol{\theta}_0 = \boldsymbol{\mu}_0$.
for $t = 0$ to $T - 1$ do
    • Take a random sample with size $L$ and selection probability $\frac{L}{N}$.
    • Compute gradient
      $\mathbf{g}_t(x_i) \leftarrow \nabla_{\boldsymbol{\mu}_t} \mathcal{L}(\boldsymbol{\theta}_t, x_i)$.
    • Clip gradient
      $\mathbf{g}_t(x_i) \leftarrow \mathbf{g}_t(x_i)\ .\ \min(1, C/\|\mathbf{g}_t(x_i)\|_2)$
    • Compute averaged gradient
      $\mathbf{g}_t \leftarrow \frac{1}{L} \sum_i \mathbf{g}_t(x_i)$
    • Update mean parameters:
      $\boldsymbol{\mu}_{t+1} \leftarrow \boldsymbol{\mu}_t - \eta_t\ \mathbf{g}_t$
    • Sample weights
      $\boldsymbol{\theta}_{t+1} \sim \mathcal{N}(\boldsymbol{\mu}_{t+1}, \frac{\eta_t^2}{L^2} C^2 z^2)$
end for
Return $\boldsymbol{\theta}_T \sim \mathcal{N}(\boldsymbol{\mu}_T, \frac{\eta_T^2}{L^2} C^2 z^2)$.
```

strong composition theorem [DRV10]. We use the open-source implementation of Rnyi DP from the TensorFlow Privacy package[15]. The total privacy loss ε is computed as a function of the noise multiplier z, the dataset size N, the lot size L, the number of iterations T, and δ.

Table 10 depicts the accuracy of various models on the MNIST test set for a range of values of ε from high to very low privacy losses. In addition to [Aba+16], we have included the TensorFlow Privacy benchmark on MNIST and LATENT which is a local-privacy based mechanism proposed recently in [Ara+19]. Additionally, we trained LeNet-5 with and without normalisation layers with DPSGD and SI-DPSGD. However, since DPSGD is not directly applicable to batch normalisation, we just show the results of DPSGD with layer normalisation. Also, for training the model augmented with batch normalisation using SI-DPSGD, we employed 30 images of KMNIST [Cla+18] as the public dataset and we added batch normalisation after each learnable layer except the last one before softmax. The probability δ is set to 10^{-5} in all our experiments.

As it is evident from Table 10, our method consistently outperforms other mechanisms for all values of privacy loss. Remarkably, the performance of LeNet-5 trained with SI-DPSGD is almost identical for various privacy losses and the largest gap between private and non-private models is just about 0.6%. In particular, using SI-DPSGD along with batch normalisation, we are able to train the LeNet-5 model under an extremely low privacy loss $\varepsilon = 0.025$ with 98.58% accuracy, which is very close to the non-private 99.2% accuracy. Training with DPSGD results in 50% reduction in accuracy for the same model.

Table 11 shows the result of our experiment on text classification. We compare the test accuracies of a BiLSTM model trained by DPSGD with or without LN layers and the same model trained by SI-DPSGD on the AG News corpus. SI-DPSGD clearly outperforms DPSGD with or without

[15] `https://github.com/tensorflow/privacy`

Table 10: Testing accuracy of various differentially private training methods on MNIST as a function of the privacy loss with $\delta = 10^{-5}$.

DP Algorithm	**privacy loss (ε)**							
	∞	7	3	1	0.5	0.1	0.05	0.025
DPSGD [Aba+16]	98.30%	96.90%	95.80%	93.10%	90.00%	NA	NA	NA
DPSGD [TensorflowPrivacy]	99.00%	97.16%	96.89%	95.00%	91.12%	84.10%	72.77%	29.40%
LATENT [Cha+19]	98.16%	97.10%	96.05%	97.10%	96.26%	NA	NA	NA
DPSGD (LeNet-5)	99.20%	97.01%	96.34%	94.11%	91.10%	83.00%	78.96%	31.56%
DPSGD (LN-LeNet-5)	99.20%	97.35%	97.05%	96.68%	94.81%	87.45%	75.76%	49.53%
SI-DPSGD (LeNet-5)	99.20%	98.90%	98.90%	98.72%	99.10%	99.00%	98.84%	90.82%
SI-DPSGD (LN-LeNet-5)	99.20%	98.85%	98.30%	98.41%	98.56%	97.92%	97.54%	98.36%
SI-DPSGD (BN-LeNet-5)	99.20%	**99.17%**	**99.17%**	**99.15%**	**99.18%**	**99.14%**	**99.12%**	**98.58%**

Table 11: Testing accuracy of various differentially private training methods on AG News as a function of the privacy loss with $\delta = 10^{-5}$.

DP Algorithm	**privacy loss (ε)**						
	∞	7	3	1	0.5	0.1	0.05
DPSGD (BiLSTM-DL)	88.47%	83.86%	80.00%	81.14%	77.88%	37.49%	31.78%
DPSGD (LN-BiLSTM-DL)	88.18%	83.54%	82.43%	82.03%	78.87%	50.09%	31.59%
SI-DPSGD (BiLSTM-DL)	88.47%	85.93%	85.70%	83.29%	81.17%	77.88%	56.72%
SI-DPSGD (LN-BiLSTM-DL)	88.18%	**87.80%**	**87.58%**	**85.74%**	**85.36%**	**84.32%**	**80.10%**

LN for all the considered privacy losses $\varepsilon = \{0.05, 0.1, 0.5, 1, 3, 7\}$, and with a large margin of over 30% improvement in accuracy for the lowest privacy losses $\varepsilon = 0.1$ or 0.05. Also, the best result we obtained is by training SI-DPSGD with LN: even at a very low privacy loss $\varepsilon = 0.05$, we only faced an 8% drop in accuracy. These experiments reveal that our proposed SI-DPSGD algorithm and layer normalisation are the two factors for attaining the highest accuracy, although SI-DPSGD contributes most to reaching this high accuracy.

The proposed method is very general and can be used to train other SLU models in a differentially private manner in the future.

3.3 Addressing more traits in speech

In Comprise, we have first focused our research on the speaker identification problem. Indeed, as already shown in our experiments, speech is a kind of biometric data that can identify speakers with very high probability. X-vectors are the most powerful speaker representation technique and they are efficient for speaker identification even with very short utterances. When such an identifier is available, all the information (that is the entire spoken message) linked to the identifier is considered as personal data. The privacy-driven speech transformation techniques we introduced in Comprise significantly lower the risk that attackers succeed to identify speakers. But the results are mitigated when the set of possible speakers is small. A possible strategy for an attacker is then to try to reduce the possible set of candidates to increase the chance of identifying someone. To do so, other traits like gender, age, or accent may be used.

In this section, we present the first results about gender identification. In gender identification, pitch is a very important feature. VoiceMask and VTLN can be parameterised so that a target speaker, and hence a target gender can be selected to adapt the pitch of the speaker. Disentangled representation based VC and VPC are also parameterised with a target speaker and implement different strategies including cross-gender transformations.

The x-vector space, when represented in a TSNE-plot, clearly shows gender clusters on original data. After transformation with the previous methods (VTLN, VoiceMask, and disentangled representation based VC), we still visually obtain very distinguishable gender clusters (see Figure 6). This is confirmed by a series of experiments where we learn gender classifiers based on x-vectors (see Table 12). We took simple k-NN classifiers with $k = 3$. The experiments are based on a small dataset of 21,650 utterances from 29 speakers (13 male / 16 female). In the (semi-)informed case, the classifier is trained and applied on data transformed with the same method. In the ignorant

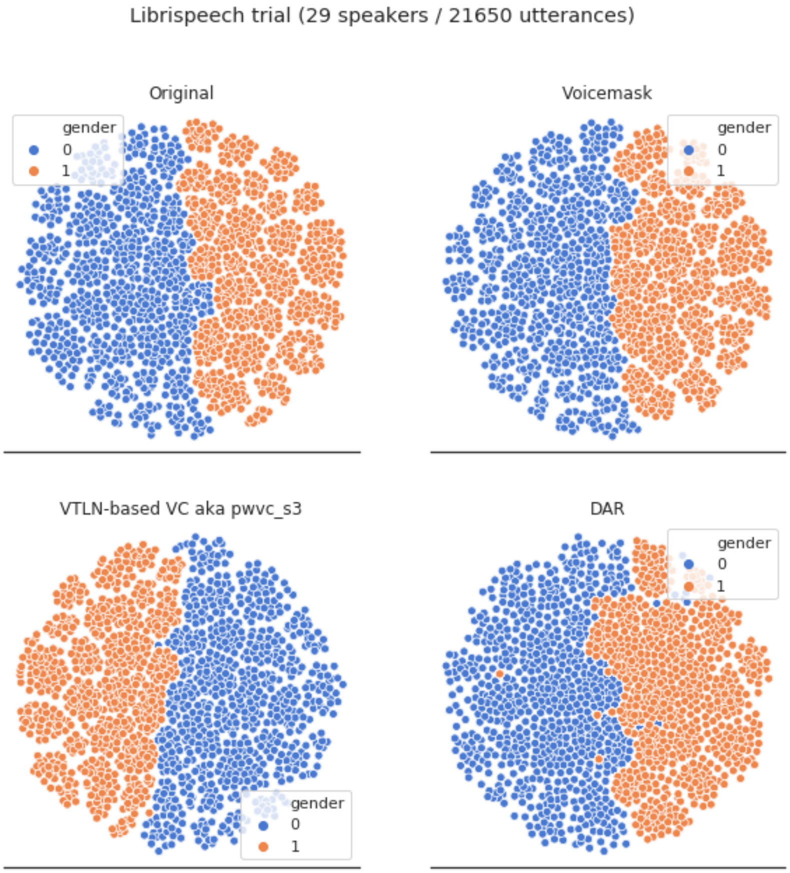

Figure 6: TSNE plots of x-vectors built on the LibriSpeech dataset. The top left subplot corresponds to the original data. The other plots are built after transformation with VoiceMask, VTLN, and disentangled representation based VC, respectively.

Table 12: Gender classification results.

Anonymization	Attack model	Accuracy
original		0.972
VoiceMask	ignorant	0.784
	informed	0.972
VTLN	ignorant	0.556
	semi-informed	0.932
	informed	0.939

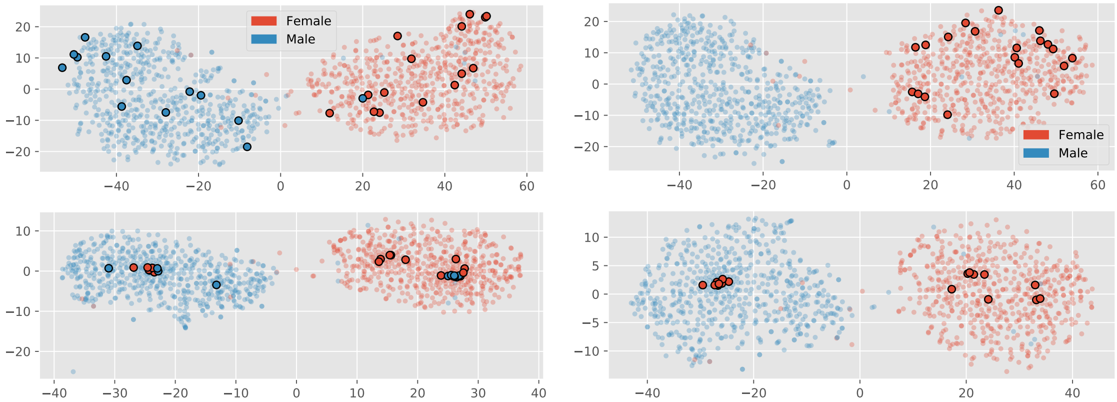

Figure 7: TSNE plots of x-vectors built on the LibriSpeech dataset. The top left subplot corresponds to the original data. The selected speakers to be transformed appear as bigger colored dots. The bottom left subplot shows where the selected speakers have been mapped when a random target speaker is used. Right: same figure for females only.

case, the classifier is trained on clean data and applied on transformed data.

With the new VPC transformer, results are much more promising, as shown in Figure 7. Indeed, speaker representations are mixed among the set of possible representations and genders are also mixed.

3.4 Hiding user attributes in speech transcripts

In Section 3.3, we described different approaches to prevent the identification of users by gender or accent on the speech signal. However, if STT is applied to obtain a speech transcript, we hypothesise that some user attributes such as identity, gender, and age may still be linked to the speaker. For instane, the speaker may use unique filler words or a specific speaking style that are captured in the text. Anonymising the speaking style of users in the text transcript is therefore important to hide these attributes and protect their privacy. A few studies have been conducting on changing the writing style of users in blogs and political speeches [SSF18; GA19] to hide traits like identity, age, and gender. These approaches have been evaluated on large blog and political speech datasets with over 3 million and 65,000 sentences, respectively. Therefore, we focus on approaches that work well in low-resource settings, i.e., small in-domain corpora of, e.g., a few hundred sentences, and which may benefit from large out-of-domain datasets.

As a first step, we verify our hypothesis that users' traits can be detected from speech transcripts by training a classifier to determine whether utterance transcripts can be used to predict gender and age. We confirm this using the Verbmobil dialogue corpus [Wah00]. Specifically, we check whether combining all the utterances of a speaker can be used to correctly classify their gender or age.

Dataset We only consider the 726 English dialogues in the Verbmobil corpus. Each dialogue involves two speakers, which brings the total number of speakers to 1,452. Out of these, we found age and gender attributes for 1,192 speakers, including 913 males and 279 females. Table 13 shows the distribution of these attributes. We chose an age threshold of 20 to split the speakers into two age groups: teenagers (below 20) and adults (above 20), with 484 and 600 speakers, respectively.

Table 13: Number of speakers per gender and age group in the Verbmobil dialogue corpus.

Gender	Number of speakers
Female	279
Male	913

Age group	Number of speakers
Teenager (< 20)	484
Adult (> 20)	600

Experimental setup and results We consider two popular text classification models: Support Vector Machines (SVM) and Convolutional Neural Networks (CNN). The features used for each of the models are word embeddings from Word2Vec [Mik+13]. We trained a 3-layer CNN word-level model with an embedding layer and one fully-connected linear layer. For the SVM, we make use of 75% of the data as training and 25% as test data. However, for the CNN model, the data splits consist of 70% training sentences, 10% validation sentences and 20% test sentences. Since the data is imbalanced, especially in terms of gender, we sample sentences from the training data that belongs to the minority class (i.e., female in the gender classification) with replacement (i.e., *upsampling*) so that the number of examples matches the majority class (i.e., male).

Table 14 shows the results of age and gender classification. We use the accuracy and the F1-score for evaluation, where the F1-score is more appropriate when the classes are imbalanced. With the CNN model, we reached an F1-score of 75% for gender classification and 74% for age classification, which is better than the values obtained by the SVM classifier. Upsampling further improves the F1-score by more than 16% and 19% for gender and age, respectively. Our results show that we can easily identify the gender and age of users by their dialogue utterances. As future work, we plan to investigate what kind of words or part of utterances are more likely to be predictive of gender and age using Layerwise Relevance Propagation [Arr+17] — an approach for explaining the features that contribute to the prediction of a class in neural networks. Also, we will consider utterance-level anonymisation of dialogue speech transcripts. Furthermore, we will evaluate our anonymisation technique on other languages like Latvian where word endings directly indicate the gender of the speaker.

Model	**Gender**		**Age**	
	Accuracy	**F1-score**	**Accuracy**	**F1-score**
SVM	72.1%	63.0%	57.5%	57.0%
CNN	83.7%	75.0%	76.0%	74.0%
SVM (upsampled)	73.7%	73.7%	77.0%	77.0%
CNN (upsampled)	**89.3%**	**89.3%**	**96.8%**	**96.8%**

Table 14: Gender classification accuracy on the Verbmobil test set for different models.

3.5 Experiments with Latvian data

It is worth noting that the training and the evaluation of our speech anonymisation transformers were done for English. But we think that our methods are language-independent and can be run on other languages without any effort. Hence, we have been working on evaluating the voice transformation procedures on Latvian. The dataset is a property of TILDE and therefore the experiments were done directly by TILDE. With respect to this initiative, we first confirmed that the instructions were sufficiently clear to run the software by users that were not involved in its development. The characteristics of the dataset are reported in Table 15. Trials are only between speakers of the same gender and hence the gender-related experiments in Section 3.3 have not been conducted.

Total trials	164068
Impostor	157156
total speakers	50 (28 - male, 22 - female)
Total number of utterances	7400

Table 15: Characteristics of the Latvian dataset.

Anonymisation results are presented in Table 16. They are very promising because a $C_{\text{llr}}^{\text{min}}$ value of 0.952 with a semi-ignorant attacker is a good indicator of anonymisation. On the utility side, after training an STT system, a WER of 11.59% was obtained on the test data. We observed an increase of the WER to 25.84% when the test data was anonymised. The numbers are in fact quite good although we were using English pre-trained voice conversion models over Latvian data. Subjectively, the anonymised speech is far more intelligible than when transformed by the VoiceMask method. But some distortions still remain.

Enrollment	**test**	**EER %**	$\mathbf{C}_{\text{llr}}^{\text{min}}$
clean	clean	19.82	0.636
clean	anonymised	49.77	1.000
anonymised	anonymised	40.27	0.952

Table 16: Anonymisation results for the Lavian data set.

4 Software library

The objective of the Work Package is to provide scripts to perform voice and text transformation. Each method has:

- a `Builder` to compute some parameters, which are specific to each method. The builder would be used to pre-build some parameters to be embedded in the app. The Builder is executed (possibly just once) in the Cloud or in a lab, and uses public datasets such as, e.g., Verbmobil, LibriSpeech, etc., in the case of speech.
- a `Transformer` to transform the data (speech or text utterances of a given speaker). The Transformer is the part that would be run on the device. This Transformer must be:
    - initialised with the pre-built parameters (and possibly additional parameters) depending on the method,
    - fed with local data in case of local adaptation (for instance, with the speaker's voice in the case of speech processing).

In Deliverable D2.1, two voice conversion techniques, VoiceMask and VTLN, and a text transformer tool were made available. The text transformer identifies sensitive named entities in a dialogue conversation and replaces them by named entities of the same type or by placeholders. Scripts were also delivered to train, transform, and evaluate these transformations. We also provided a speech and text alignment tool and a word masking tool to remove sensitive words identified by the text transformer tool in the speech signal.

In the current deliverable, we improve voice transformations with the VPC method introduced in Section 3.1, and we propose new software to train Scale Invariant Differentially Private models with linear, CNN and LSTM architectures as presented in Section 3.2.

We provide Docker containers for the voice transformer and builder and for the text transformers on the client side. Detailed information is available in the README of the Comprise git repository of the deliverable and its subtrees.[16] In this report we enlighten the ease of use of the available software by some examples using the Docker files.

4.1 Voice Transformer

The new VPC voice transformer is available as a git repository and as a Docker container.

4.1.1 Builder

The builder of this voice transformer extracts x-vectors from a pool of speakers that forms a subset of LibriSpeech:

```
sudo docker run -it --gpus all  \
            -v "$(pwd)"/io:/opt/vpc/io \
            registry.gitlab.inria.fr/comprise/development/vpc-transformer:deliv22 \
            ./build.sh --anoni_pool train-other-500
```

[16] https://gitlab.inria.fr/comprise/deliverables/deliverable_d22

4.1.2 Transformer

```
sudo docker run -it --gpus all  \
            -v "$(pwd)"/io:/opt/vpc/io \
            registry.gitlab.inria.fr/comprise/development/vpc-transformer:deliv22 \
            ./transform.sh --ipath ./inputs/e0003.wav
```

The configuration is stored in io/config/config_transform.sh.

```
wgender=f                             # gender m or f
pseudo_xvec_rand_level=spk            # spk (all utterances will have same xvector)
                                      # or utt (each utterance will have randomly
                                      # selected xvector)
#cross_gender="same"                  # false, same gender xvectors will be selected;
                                      # true, other gender xvectors
#cross_gender="other"                 # false, same gender xvectors will be selected;
                                      # true, other gender xvectors; random gender
                                      # can be selected
cross_gender="random"                 # false, same gender xvectors will be selected;
                                      # true, other gender xvectors; random gender
                                      # can be selected
distance="plda"                       # cosine or plda
#proximity="random"                   # nearest or farthest speaker to be selected
#proximity="farthest"                 # nearest or farthest speaker to be selected
proximity="dense"                     # dense or sparse region to be selected
anoni_pool="train_other_500"          # anonymisation pool
```

4.2 Text Transformer

The text transformer is available as a git repository and as a Docker container as well. See below a full example of installation followed by the transformation of a text file test.txt located in the host inputs directory into a transformed file out.txt located in the results directory.

```
git clone https://gitlab.inria.fr/comprise/development/text-transformer
cd text-transformer
docker run -v $(pwd)/io:/opt/io\
           registry.gitlab.inria.fr/comprise/development/text_transformer:deliv22 \
           python transform.py io/inputs/test.txt ./io/test_out.txt
```

The builder for text transformers was delivered in the last period and is available in the COMPRISE git repository.[17]

4.3 SI-DPSGD

The SI-DPSGD training tool is available in the git repository:

[17] https://gitlab.inria.fr/comprise/text_transformer

```
git clone https://gitlab.inria.fr/comprise/sidp
cd sidp
```

Before running SI-DPSGD on any dataset, users must install `tensorflow privacy`[18] and run `compute_dp_sgd_privacy.py`[19] to obtain the noise multiplier σ corresponding to their choice of ε (privacy loss). For example,

```
compute_dp_sgd_privacy.py --N=60000 --batch_size=256\
                          --noise_multiplier=1.1 \
                          --epochs=60 --delta=1e-5
```

The following scripts run SI-DPSGD for image classification and text classification. For image classification on MNIST data, run:

```
main_mnist.py [experiment_name] [noise_multiplier] [clip]
```

where

- experiment_name is one of the following: ”non-dp”, ”dp” or ”sidp”. ”non-dp” trains a non-differentially private model (i.e., no privacy constraints), ”dp” is the standard DPSGD approach and ”sidp” is the proposed SI-DPSGD approach.
- noise_multiplier is the amount of noise σ to be added to the gradient
- clip is the clipping factor C to preserve the sensitivity of differential privacy.

For text classification on the AG News corpus, run:

```
python main_text_classification.py [experiment_name] [noise_multiplier] [clip]
```

5 Summary

In this deliverable, we have followed the objectives stated in the proposal, namely the design, implementation, and evaluation of speech and text transformations addressing more types of private information and initial statistical utility/privacy bounds. We have studied in depth statistical privacy bounds through the examination of several privacy measures (Section 2.3) and the introduction of more challenging scenarios concerning the possible knowledge of an attacker (Section 2.2). In the case of text transformation, we have examined the trade-off between utility and privacy through the lens of the information-theoretic approach of differential privacy. We have started to address more types of private information by considering gender and age in Sections 3.3 and 3.4. We have also paved the way for better privacy-preserving learning, by introducing an efficient method for differentially private training of deep learning models (Section 3.2) and by conducting a study of context data and metadata (Section 2.1). On the implementation side, we deliver an improved speech transformation tool that provides better anonymisation and a new differentially private deep learning training tool.

Future work includes several directions:

[18] https://github.com/tensorflow/privacy

[19] https://github.com/tensorflow/privacy/blob/master/tensorflow_privacy/privacy/analysis/compute_dp_sgd_privacy.py

- Separation of the privacy concerns regarding STT and SLU training. Indeed, we observe that there may be privacy leaks when considering joint speech and text data at the dialogue or even the sentence level. Therefore we plan to build anonymised speech datasets for STT training by collecting short utterance segments and text-only datasets for SLU training.

- The extensive study of the x-vector space has revealed further insights and we are now ready to improve again our transformation tool using a better strategy to select target (pseudo-)speakers. We have started to work on it.

- Another important way to improve anonymisation procedures is to follow the idea of adding noise to fulfil the requirements of differential privacy. Therefore, an intensive study of the robustness to noise of our training methods is useful. Again, we have started to work on that topic and we will continue in the next period.

- The attacker's knowledge is of primary importance when studying privacy measures. In the adversarial approaches we have studied in the previous period, the design of a good attacker is essential to obtain a robust anonymised representation of speech data. We have started to study how variational Bayes approaches can strengthen the adversary by virtually building a large range of possible attackers.

- On the software side, we will integrate these tools in the COMPRISE SDK Client Library.

References

[Aba+16] M. Abadi et al. “Deep learning with differential privacy”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 308–318.

[ABV18] A. Akbik, D. Blythe, and R. Vollgraf. “Contextual String Embeddings for Sequence Labeling”. In: *COLING 2018, 27th International Conference on Computational Linguistics*. 2018, pp. 1638–1649.

[Ara+19] P. C. M. Arachchige et al. “Local Differential Privacy for Deep Learning”. In: *IEEE Internet of Things Journal* (2019), pp. 1–1.

[Arr+17] L. Arras et al. “Explaining Recurrent Neural Network Predictions in Sentiment Analysis”. In: *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 159–168.

[BKH16] J. L. Ba, J. R. Kiros, and G. E. Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[Blu+15] C. Blundell et al. “Weight uncertainty in neural networks”. In: *arXiv preprint arXiv:1505.05424* (2015).

[Cha+19] M. Chamikara et al. “Local differential privacy for deep learning”. In: *arXiv preprint arXiv:1908.02997* (2019).

[CL19] J.-c. Chou and H.-Y. Lee. “One-Shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization”. In: *Proc. INTERSPEECH*. 2019, pp. 664–668.

[Cla+18] T. Clanuwat et al. *Deep Learning for Classical Japanese Literature*. Dec. 3, 2018.

[Deh+10] N. Dehak et al. “Front-end factor analysis for speaker verification”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2010), pp. 788–798.

[Dev+18] J. Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018).

[DR+14] C. Dwork, A. Roth, et al. “The algorithmic foundations of differential privacy”. In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

[DRV10] C. Dwork, G. N. Rothblum, and S. Vadhan. “Boosting and differential privacy”. In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE. 2010, pp. 51–60.

[Fan+19] F. Fang et al. “Speaker Anonymization Using x-vector and Neural Waveform Models”. In: *Proc. 10th ISCA Speech Synthesis Workshop*. 2019, pp. 155–160.

[GA19] T. Gröndahl and N. Asokan. “Effective writing style imitation via combinatorial paraphrasing”. In: *CoRR* abs/1905.13464 (2019).

[Gom+17] M. Gomez-Barrero et al. “General framework to evaluate unlinkability in biometric template protection systems”. In: *IEEE Transactions on Information Forensics and Security* 13.6 (2017), pp. 1406–1420.

[He+16] K. He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[HXY15] Z. Huang, W. Xu, and K. Yu. "Bidirectional LSTM-CRF Models for Sequence Tagging". In: *ArXiv* abs/1508.01991 (2015).

[IS15] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[ISO17] ISO/IEC FDIS 30136. "Information Technology—Performance Testing of Biometric Protection Schemes". In: *ISO/IEC JTCI SC37 Biometrics* (2017).

[KB14] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014).

[LeC+98a] Y. LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[LeC+98b] Y. LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE*. 1998, pp. 2278–2324.

[Mik+13] T. Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 3111–3119.

[Mir17] I. Mironov. "Rényi differential privacy". In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE. 2017, pp. 263–275.

[MJ20] A. Moretón and A. Jaramillo. "How can Private Information Recorded by Voice-enabled Systems be Identified?" Under submission. 2020.

[Mor15] M. Morise. "CheapTrick, a spectral envelope estimator for high-quality speech synthesis". In: *Speech Communication* 67 (2015), pp. 1–7.

[Pan+15] V. Panayotov et al. "LibriSpeech: an ASR corpus based on public domain audio books". In: *Proc. ICASSP*. 2015, pp. 5206–5210.

[Pas+17] A. Paszke et al. "Automatic differentiation in pytorch". In: (2017).

[Pas+19] A. Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[Pov+11] D. Povey et al. *The Kaldi speech recognition toolkit*. Tech. rep. 2011.

[Pov+18] D. Povey et al. "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks." In: *Interspeech*. 2018, pp. 3743–3747.

[PPK15] V. Peddinti, D. Povey, and S. Khudanpur. "A time delay neural network architecture for efficient modeling of long temporal contexts". In: *Interspeech*. 2015, pp. 3214–3218.

[Qia+18] J. Qian et al. "Hidebehind: Enjoy Voice Input with Voiceprint Unclonability and Anonymity". In: *Proc. the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2018, pp. 82–94.

[Raj+19] D. Raj et al. *Probing the Information Encoded in X-vectors*. 2019.

[SN03] D. Sundermann and H. Ney. "VTLN-based voice conversion". In: *Proc. 3rd IEEE International Symposium on Signal Processing and Information Technology*. 2003, pp. 556–559.

[Sny+18] D. Snyder et al. "X-vectors: Robust DNN embeddings for speaker recognition". In: *Proc. ICASSP*. 2018, pp. 5329–5333.

[Sri+20] B. M. L. Srivastava et al. “Evaluating Voice Conversion-based Privacy Protection against Informed Attackers”. In: *ICASSP*. 2020.

[SS15] R. Shokri and V. Shmatikov. “Privacy-preserving deep learning”. In: *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM. 2015, pp. 1310–1321.

[SSF18] R. Shetty, B. Schiele, and M. Fritz. “A4NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation”. In: *27th USENIX Security Symposium (USENIX Security 18)*. Aug. 2018.

[SZ14] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014.

[THG04] M. .-.-J. Tang, D. Z. Hakkani-Tür, and A. GokhanTur. “Preserving Privacy in Spoken Language Databases”. In: 2004.

[UVL17] D. Ulyanov, A. Vedaldi, and V. Lempitsky. “Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis”. In: *Proc. CVPR*. 2017, pp. 6924–6932.

[VYM19] C. Veaux, J. Yamagishi, and K. MacDonald. *CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)*. 2019.

[Wah00] W. Wahlster, ed. *Verbmobil: Foundations of Speech-to-Speech Translation*. Berlin: Springer, 2000.

[War65] S. L. Warner. “Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias”. In: *Journal of the American Statistical Association* 60.309 (1965). PMID: 12261830, pp. 63–69.

[Wat+18] S. Watanabe et al. “ESPnet: End-to-End Speech Processing Toolkit”. In: *Proc. INTERPSPEECH*. 2018, pp. 2207–2211.

[WTY19] X. Wang, S. Takaki, and J. Yamagishi. “Neural source-filter-based waveform model for statistical parametric speech synthesis”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 5916–5920.