Prεεch: A System for Privacy-Preserving Speech Transcription

Shimaa Ahmed, Amrita Roy Chowdhury, Kassem Fawaz, and Parmesh Ramanathan
University of Wisconsin-Madison
{ahmed27, roychowdhur2, kfawaz, parmesh.ramanathan}@wisc.edu

Abstract

New advances in machine learning have made Automated Speech Recognition (ASR) systems practical and more scalable. These systems, however, pose serious privacy threats as speech is a rich source of sensitive acoustic and textual information. Although offline and open-source ASR eliminates the privacy risks, its transcription performance is inferior to that of cloud-based ASR systems, especially for real-world use cases. In this paper, we propose Prεεch, an end-to-end speech transcription system which lies at an intermediate point in the privacy-utility spectrum. It protects the acoustic features of the speakers' voices and protects the privacy of the textual content at an improved performance relative to offline ASR. Additionally, Prεεch provides several control knobs to allow customizable utility-usability-privacy trade-off. It relies on cloud-based services to transcribe a speech file after applying a series of privacy-preserving operations on the user's side. We perform a comprehensive evaluation of Prεεch, using diverse real-world datasets, that demonstrates its effectiveness. Prεεch provides transcription at a 2% to 32.25% (mean 17.34%) relative improvement in word error rate over Deep Speech, while fully obfuscating the speakers' voice biometrics and allowing only a differentially private view of the textual content.

1 Introduction

New advances in machine learning and the abundance of speech data have made Automated Speech Recognition (ASR) systems practical and reliable [5, 17]. ASR systems have achieved a near-human performance on standard datasets [5, 17], at a scale. This scalability is desirable in many domains, such as journalism [25], law, business, education, and health care, where cost, delay, and third-party legal implications [29] prohibit the application of manual transcription services [12]. For example, recent research has identified private voice transcription as one of the challenges journalists face when interviewing sensitive sources [25].

Several companies, such as Google and Amazon, provide online APIs for speech transcription. This convenience, however, comes at the cost of privacy. A speech recording contains acoustic features that can reveal sensitive information about the user, such as age, gender [39], emotion [4, 40], accent, and health conditions [41]. The acoustic features are also biometric identifiers of the speakers [26], enabling speaker identification and impersonation [20]. Additionally, the textual content of speech can be sensitive [29]. For example, medical recordings can contain private health information about patients [12], and business recordings can include proprietary information. Current cloud services already support several speech processing APIs like speaker identification and diarization. They also support text analysis APIs, such as topic modeling, document categorization, sentiment analysis, and entity detection (Sec. 3.2), that can extract sensitive information from text. Applying these APIs to the recorded speech can significantly undermine the user's privacy.

Offline and open-source transcription services, like Deep Speech [18], solve these privacy challenges as the speech files never leave the user's trust boundary. However, we find that their performance does not match that of a cloud service provider [45], especially on real-world conversations and different accents (Sec. 2.2). Thus, the primary goal of this paper is to: *provide an intermediate solution along the utility-privacy spectrum that uses cloud services while providing a formal privacy guarantee.*

We present Prεεch (Privacy-Preserving Speech) as a means to achieve this goal; it is an end-to-end speech transcription system that: (1) protects the users' privacy along the acoustic and textual dimensions; (2) improves the transcription performance relative to offline ASR; and (3) provides the user with control knobs to customize the trade-offs between utility, usability, and privacy.

Textual Privacy: Prεεch segments and shuffles the input speech file to break the context of the text, effectively transforming it into a bag-of-words. Then, it injects dummy (noise) segments to provide the formal privacy guarantee of differential privacy (DP) [13].

Acoustic Privacy: Preech applies voice conversion to protect the acoustic features of the input speech file and ensure noise indistinguishability.

We evaluate Preech over a set of real-world datasets covering diverse demographics. Our evaluation shows that Preech provides a superior transcription accuracy relative to Deep Speech, the state-of-the-art offline ASR. Also, Preech prevents cloud services from extracting any user-specific acoustic features from the speech. Finally, applying Preech thwarts the learning of any statistical models or sensitive information extraction from the text via natural language processing tools.

In summary, the main contributions of this paper are:

(1) End-to-end practical system: We propose Preech, a new end-to-end system that provides privacy-preserving speech transcription at an improved performance relative to offline transcription. Specifically, Preech shows a relative improvement of 2% to 32.52% (mean 17.34%) in word error rate (WER) on real-world evaluation datasets over Deep Speech, while fully obfuscating the speakers' voice biometrics and allowing only a DP view of the textual content.

(2) Non-standard use of differential privacy: Preech uses DP in a *non-standard way*, giving rise to a set of new challenges. Specifically, the challenges are (1) "noise" corresponds to concrete words, and need to be added in the speech domain (2) "noise" has to be indistinguishable from the original speech (details in Sec. 4.5).

(3) Customizable Design: Preech provides several *control knobs* for users to customize the functionality based on their desired levels of utility, usability, and privacy (Sec. 7.4). For example, in a relaxed privacy setting, Preech's relative improvement in WER ranges from 44% to 80% over Deep Speech (Sec. 7.4.1).

The full version of this paper is available online [3], and some demonstrations of Preech are available at this link [2].

2 Speech Transcription Services

We first provide some background on online and offline speech transcription services. Next, we present a utility evaluation using standard and real-world speech datasets.

2.1 Background

Speech transcription refers to the process of extracting text from a speech file. ASR systems are available to the users either through cloud-based online APIs or offline software.

(1) Cloud-Based Transcription: We utilize two cloud-based speech transcription services – Google's Cloud Speech-to-Text and Amazon Transcribe.

(2) Offline Transcription: We consider the Deep Speech architecture from Baidu [18], which is trained using Mozilla's [1] Common Voice dataset as a representative offline transcription service. This dataset is crowdsourced and open-source. Specifically, we use the Deep Speech 0.4.1 model [2] (released in January 2019). Note that we do not consider offline transcribers that are not open for general use. For example, Google's on-device speech recognizer [1] is an offline transcriber that is currently only supported on Google's Pixel devices and does not allow an API or open-source access, limiting its usability.

Notations: Let S denote the input speech file associated with a ground truth transcript T_S^g. The user can either use a cloud service provider (CSP) or an offline service provider (OSP) to obtain the transcript (denoted by T_S^{CSP} or T_S^{OSP}, respectively).

Transcription Accuracy: The standard metric for quantifying the accuracy loss from transcription is the word error rate (WER) [18]. WER treats the transcript as a sequence of words. It models the difference between the two sequences by counting the number of deleted words (D), the number of substituted words (U), and the number of injected words (I). If the number of words in T_S^g is W, WER is given as: $\frac{D+U+I}{W}$.

2.2 Utility Comparison

In this section, we empirically evaluate the utility gap between the CSP and the OSP over a wide range of standard and real-world datasets. We use these datasets throughout the paper.

Standard Datasets: These datasets include (1) the TIMIT-TEST subset [16], (2) a subset from Librispeech *dev-clean* dataset [31], and (3) the DAPS dataset [28]. TIMIT-TEST [3] subset comprises of 1344 utterances by 183 speakers from eight major dialect regions of the United States. The LibriSpeech subset consists of eleven speakers, 20 utterances each. For DAPS, we use the evaluation subset prepared for the 2018 voice conversion challenge [24] that consists of five scripts read by ten speakers: five males and five females.

Real-world Datasets: We also assess the real-world performance of both transcription services on non-American accent datasets and real conversations among speakers of different demographics. For the accented datasets, we evaluate 200 utterances of two speakers from the VCTK dataset [46]: speaker p262 of a Scottish accent and speaker p266 of an Irish accent. For the real-world datasets, we evaluate 20 minutes of speech from the "Facebook, Social Media Privacy, and the Use and Abuse of Data" hearing before the U.S. Senate [4]. We construct the 20 minutes by selecting three continuous chunks of speech from the hearing such that they include nine speakers: 8 senators and Mark Zuckerberg. Another real-world dataset is the Supreme Court of the United States case "Carpenter v. United States" [5]. For this dataset, we evaluate a total of 40 minutes of speech from the advocates in the case.

[1] https://voice.mozilla.org/en/datasets

[2] https://github.com/mozilla/DeepSpeech

[3] https://catalog.ldc.upenn.edu/LDC93S1

[4] https://www.commerce.senate.gov/2018/4/facebook-social-media-privacy-and-the-use-and-abuse-of-data

[5] https://www.oyez.org/cases/2017/16-402

	Datasets	Google	AWS	Deep Speech
Standard	LibriSpeech	9.14	8.83	9.37
	DAPS	6.70	7.53	10.65
	TIMIT TEST	6.27	7.11	20.08
Non-Standard	VCTK p266	5.15	10.09	26.72
	VCTK p262	4.53	7.87	15.97
	Facebook 1	5.76	7.45	24.72
	Facebook 2	3.07	8.19	26.61
	Facebook 3	8.32	9.42	30.72
	Carpenter 1	9.44	9.44	25.85
	Carpenter 2	9.22	11.53	39.71

Table 1: WER (%) comparison of cloud services, Google and AWS, versus the state-of-the-art offline system, Deep Speech.

Accuracy Comparison: Table 1 presents the WER comparison results. The results show that the CSPs are superior to the OSP on all the datasets. The performance gap, however, is more significant on the non-standard datasets; the CSP outperforms Deep Speech by 60% to 80% in WER.

3 Privacy Threat Analysis

We study the privacy threats that a cloud-based transcription service poses while processing private speech data.

3.1 Voice Analysis

The biometric information embedded in S can leak sensitive information about the speakers, including their emotional status [4, 40], health condition [41], sex [39], and even identity [26]. Furthermore, extracting this information enables critical attacks like voice cloning and impersonation attacks [23, 47]. In this section, we showcase a few representative examples of how cloud-based APIs can pose serious privacy threats to the acoustic features within S.

Speaker Diarization: CSPs utilize advanced diarization capabilities to cluster the speakers within a speech file, even if they have not been observed before. The basic idea is to (1) segment the speech file into segments of voice activity, and (2) extract a speaker-specific embedding from each segment, such that (3) segments with close enough embeddings should belong to the same speaker. We verified the strength of the diarization threat over three multi-speaker datasets: VCTK (mixing p266 and p262), Facebook, and Carpenter. We measure the performance of the IBM diarization service using Watson's Speech-to-Text API [6] via Diarization Error Rate (DER). DER estimates the fraction of time the speech file segments are not attributed to the correct speaker cluster. The DER values are 0%, 4.85%, and 1.32% for the three datasets, respectively. Hence, the API can correctly distinguish between, and cluster, the different speakers, more than 95% of the entire dataset duration despite lacking any prior information about the individual speakers.

Speaker Identification: A speaker identification task maps the speech segments in a speech file to an individual. We use the Azure Identification API, which consists of two stages: (1) user enrollment and (2) identification (whether a given voice sample matches any of the enrolled users). The enrollment stage requires only 30 seconds of speech from each user to extract their voice-print. We enrolled 22 speakers as follows: 10 from DAPS, two from VCTK, two from Carpenter, and eight from Facebook. The identification accuracy was nearly 100% for all speakers.

Speaker Cloning and Impersonation: Lastly, we applied a Tacotron-based speech synthesizer from Google [20]; a network that can synthesize speech in the voice of any speaker. The network generates a target speaker's embedding, which it uses to synthesize speech on a given piece of text. In our setting, we used the network to generate the speakers' embedding in our evaluation datasets. Then, we synthesized eight speech utterances using the embeddings of each speaker. We enrolled the speakers in Azure's Speech Identification API using their natural voice samples and tested whether the API will map the synthesized segments to the corresponding speaker. Except for the second speaker in Carpenter, the cloned samples were successfully identified as the true speakers.

3.2 Text Analysis

CSPs possess natural language processing (NLP) capabilities that enable automated statistical analyses on large sets of documents. Those analyses fall into two broad categories. The first type involves identifying specific words from the transcript that correspond to sensitive information such as an address, name, and SSN using named-entity extraction [14]. The other type of analysis involves statistically analyzing the entire transcript on the whole to extract some semantic or user-identifying information. This analysis uses two types of information: the set of words (i.e., bag-of-words representation of the transcript) and their order of appearance (to capture the context).

Bag-of-Words Analysis: One of the most commonplace analysis that treats a document as a bag-of-words is *topic modeling* [37, 43]. Topic modeling is an unsupervised machine learning technique that identifies clusters of words that best characterize a set of documents. Another popular technique is *stylometry analysis*, which aims at attributing authorship (in our case, the speaker) of a document based on its literary style. It is based on computing a set of stylistic features like mean word length, words histogram, special character count, and punctuation count from the disputed document [30].

[6] https://www.ibm.com/cloud/watson-speech-to-text

Context-based Analysis: An example of context-based analysis is sentiment analysis (understanding the overall attitude in a block of text). Text categorization is another example; it refers to classifying a document according to a set of predetermined labels.

4 Prεεch

Our discussion in the previous sections highlights a trade-off between privacy and utility. The OSP provides perfect privacy at the cost of higher error rates, especially for non-standard speech datasets. On the other hand, clear privacy violations accompany revealing the speech recording to the CSP. Motivated by this trade-off, we present Prεεch, a practical system that lies at an intermediate point along the utility-privacy spectrum of speech transcription.

4.1 System and Threat Models

We consider the scenario where users have audio recordings of private conversations that require high transcription accuracy. For example, a journalist with recordings of confidential interviews is a paradigmatic user for Prεεch. Other examples include a therapist with recordings of patient therapy sessions or a course instructor with oral examination records of students. Prεεch, however, does not target real-time transcription applications. For example, voice assistants and online transcription (e.g. a live-streaming press conference) are *out-of-scope*. Thus, for our target use cases, the latency of transcription is not a critical concern.

The adversary is the CSP or any other entity having direct or indirect access to the stored speech at the CSPs. This adversary is capable of the aforementioned voice- and text-based analysis.

4.2 Prεεch Overview

Prεεch provides an end-to-end tunable system which aims at satisfying the following design goals:

1. protect the users' privacy along the acoustic and textual dimensions;
2. improve on the transcription accuracy compared to offline models; and
3. provide the users with control knobs to customize Prεεch's functionality according to their desired level of utility, usability, and privacy.

To this end, Prεεch applies a series of *privacy-preserving operations* to the input speech file before sending it to the CSP. Fig. 1 shows the high-level overview of Prεεch. Below, we briefly describe Prεεch's privacy-preserving operations.

4.2.1 Preserving Textual Privacy

Prεεch protects the privacy of the textual content of an input speech file S through the following three operations:

Segmentation and shuffling: Prεεch breaks S into a sequence of segments, denoted by $\mathbb{S}$. This is followed by shuffling the segments to remove all ordering information. Thus, segmenting and shuffling S transform its textual content into a bag-of-words representation.

Sensitive word scrubbing (SWS): First, Prεεch applies the OSP to identify the list of sensitive keywords that contain numbers, proper nouns, or any other user-specified words. Next, Prεεch applies keyword spotting, KWS, (identify portions of the speech that correspond to a keyword) to each of the segments in $\mathbb{S}$. Only the segments that do not contain a keyword pass to the CSP for transcription.

Dummy word injection to ensure differential privacy: The bag-of-words representation of a transcript corresponds to its word histogram (Sec. 4.5). As discussed in Sec. 3.2, several statistical analyses can be built on the word histogram of the transcript T_S^{CSP} such as topic modeling or stylometry analysis. Thus, protecting the privacy of this word histogram is a primary focus of Prεεch, and the privacy guarantee we choose is that of differential privacy. To this end, Prεεch ensures DP by adding a suitable amount of dummy words to S before sending it to the CSP. This way, the CSP is allowed only a differentially private view of the word histogram and any subsequent statistical model built over it (by Thm. 4.1 in Sec. 4.5).

The main challenge in this setting is that the dummy words must be added in the speech domain, which Prεεch addresses as follows. First, Prεεch estimates the general domain of the text for S (specifically its *vocabulary*, details in Sec. 4.5) from T_S^{OSP}. Next, it generates dummy text segments using a state-of-the-art NLP language model. Finally, Prεεch applies text-to-speech (TTS) transforms to these dummy segments and adds them to S. However, leaving it just at this would be insufficient as the CSP can potentially distinguish between the two different sources of speech (TTS generated dummy segments and segments in S) based on their acoustic features. Therefore, Prεεch provides the user with multiple options to synthesize *indistinguishable* dummy segments, namely (1) voice cloning [20], and (2) voice conversion [21, 44]. These options offer different trade-offs between utility, usability, and privacy (Secs. 4.5.2 and 4.6). As stated in Sec. 3.2, text-based attacks exploit individual sensitive words or the order of the words or the word histogram. Thus, from the above discussion, Prεεch protects privacy along all three dimensions (evaluation results in Sec. 7).

4.2.2 Preserving Voice Privacy

Voice conversion, VC, is a standard speech processing technique that transforms the voice of a source speaker of a speech

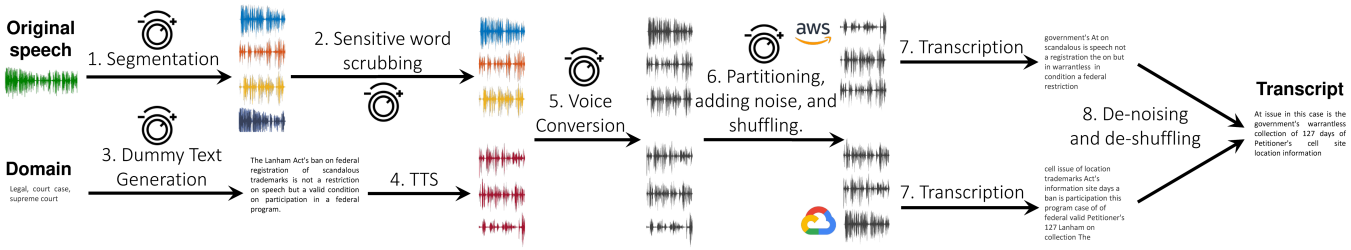


Figure 1: High-level overview of Prεεch, showing the knobs where a user can tune the associated trade-offs.

utterance to that of another speaker. Prεεch applies voice conversion to fulfill a two-fold agenda. First, it obfuscates the sensitive voice biometric features in S. Second, VC ensures that the dummy segments (noise added to ensure differential privacy) are acoustically indistinguishable from the original speech file segments. There are two main categories in voice conversion: one-to-one VC, and many-to-one VC (Sec. 4.6).

4.2.3 End-to-End System Description

Fig. 1 depicts the workflow of Prεεch. Given a speech file S, the first step **(1)** is to break S into a sequence of disjoint and short speech segments, $\mathbb{S}$. This is followed by **(2)** sensitive word scrubbing where speech segments containing numbers, proper nouns, and user-specified keywords are removed from $\mathbb{S}$. Next, **(3)** given the domain of S's textual content (its vocabulary), Prεεch generates a set of text segments (as is suitable for satisfying the DP guarantee as discussed in Sec. 4.5), and subjects it to TTS transformation **(4)**. At this point, Prεεch has audio segments for the input speech, $\mathbb{S}$, as well as the dummy segments, $\mathbb{S}_d$. If the user also wants to hide the voice biometric information in S, Prεεch applies **(5)** voice conversion over all the segments in $\mathbb{S} \bigcup \mathbb{S}_d$ to convert them to the same target speaker. This process hides the acoustic features of S and ensures that the segments in $\mathbb{S}$ and $\mathbb{S}_d$ are indistinguishable. This is followed by Prεεch partitioning $\mathbb{S}$ across $N > 0$ non-colluding CSPs (Sec. 4.5). This partitioning reduces the number of dummy segments that are required to achieve the DP guarantee (Sec. 4.5). Next, Prεεch adds a suitable amount of dummy segments from $\mathbb{S}_d$ to each partition $\mathbb{S}_i, i \in [N]$ and shuffles them. Additionally, Prεεch keeps track of time-stamps of the dummy segments, TS_i and order of shuffling, $Order_i$ for each such partition **(6)**. After obtaining the transcript **(7)** for each partition from the N CSPs, Prεεch removes $\mathbb{S}_d$'s transcripts and de-shuffles the remaining portion of the transcript using TS_i and $Order_i$, and outputs the final transcript to the user **(8)**.

In what follows, we elaborate on the key components of Prεεch, namely segmentation, sensitive word scrubbing, DP word histogram release, and voice conversion.

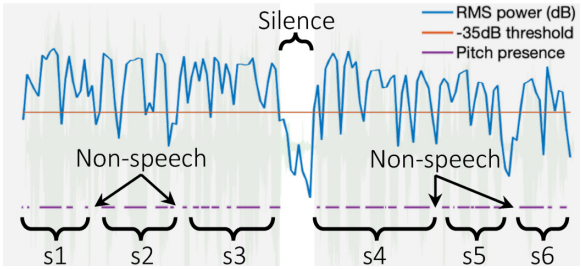


Figure 2: An illustration of Prεεch's segmentation algorithm. The coarse segments in light gray. The absence of pitch information indicate non-speech instances, which further breaks down the coarse segments into finer segments.

4.3 Segmentation Algorithm

A key component of Prεεch is breaking the textual context by segmenting S. We represent S as a sequence of segments $\mathbb{S}$, where each segment can contain multiple words. Prεεch applies a hierarchical segmentation approach that starts with a stage of silence detection based on the energy level, followed by pitch detection to detect speech activity for finer segmentation. The mechanism is illustrated in Fig. 2.

We define a *period of silence* as the time duration when the RMS power of the speech signal drops below -35 dB for at least 500ms. The initial segmentation stage detects such silence periods from S resulting in coarse segments. A human speech signal can be viewed as a modulated periodic signal where the signal period is referred to as the *glottal cycle* [27]. In the second stage, Prεεch uses the existence of glottal cycles [7] to detect human voice, which breaks down the coarse segments into finer ones. A time duration of at least 20 ms without the presence of glottal cycles is regarded as *non-speech*.

As some segments might be abrupt or too short to allow for correct speech recognition, Prεεch performs two additional optimization steps. First, it merges nearby fine segments to ensure a minimum length per segment. Second, it does not partition segments at the boundaries of the identified human speech and allows 40 ms of non-speech to be included at the beginning and the end of each segment.

Control Knob: Segmenting S presents with a trade-off – smaller segments result in better privacy guarantee at the expense of deteriorated transcription accuracy due to semantic context loss. Prεεch allows the user to tune the *minimum length of the segments as a means to control this trade-off.*

4.4 Sensitive Word Scrubbing

Prεεch performs sensitive word scrubbing (SWS) as follows. First, it obtains the offline transcript of S, T_S^{OSP}. Next, it applies named entity recognition (NER) on T_S^{OSP}. NER is an NLP technique that seeks to locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, monetary values, etc. Prεεch also gives the option for users to specify some keywords of their choice. This allows customization of the sensitive keyword list as users have subjective ideas of what they might consider sensitive.

After the list of sensitive words is finalized, Prεεch applies keyword spotting (KWS) on the segments. KWS is needed for the following three reasons. First, KWS is used to spot the user-defined keywords which cannot be identified by NER. Second, the initial T_S^{OSP} is generated on S without segmentation to achieve the highest estimation accuracy. However, for Prεεch, we need to identify the segments containing the keywords. Finally, the OSP might not transcribe the named-entities correctly at all locations. For example, the name "Carpenter" might be repeated 20 times in S, while the OSP transcribes it accurately only five times. KWS has higher accuracy in spotting keywords than the OSP's transcription accuracy.

Control Knob: KWS takes the list of keywords and matches them phonetically to a speech file based on a sensitivity score. This sensitivity score sets a threshold for the phonetic similarity required for a keyword to be spotted. A low score results in false positives by flagging phonetically similar words as keywords which degrades the utility by transcribing non-sensitive segments using the OSP. Conversely, a high score could result in some keywords being missed and revealed to the CSP. Hence, the sensitivity score is a trade-off parameter between privacy and utility (Sec. 7.3.1).

4.5 Differentially Private Word Histogram

We define *vocabulary*, $\mathcal{V}$, to be the domain of non-stop and stemmed words from which T_S^g is constructed. Let c_i denote the frequency of the word $w_i \in \mathcal{V}$ in T_S^g. As is typical in the NLP literature, we model the transcription as a bag of words: $BoW = \{w_i : c_i | w_i \in \mathcal{V}\}$. Additionally, let H represent $[c_i]$ – the count vector of BoW. In other words, the bag of words model represents a histogram on the vocabulary, i.e., a mapping from $\mathcal{V}$ to $\mathbb{N}^{|\mathcal{V}|}$.

4.5.1 Privacy Definition

As discussed in Sec. 3.2, the aforementioned word histogram is sensitive and can only be released to the CSP in a privacy-preserving manner. Our privacy guarantee of choice is DP which is the de-facto standard for achieving data privacy [11, 13, 15]. DP provides provable privacy guarantees and is typically achieved by adding noise to the sensitive data.

Definition 4.1 ((ε,δ)-differentially private d-distant histogram release)**.** A randomized mechanism $\mathcal{A} : \mathbb{N}^{|\mathcal{V}|} \to \mathbb{N}^{|\mathcal{V}|}$, which maps the original histogram into a noisy one, satisfies (ε,δ)-DP if for any pair of histograms H_1 and H_2 such that $||H_1 - H_2||_1 = d$ and any set $O \subseteq \mathbb{N}^{|\mathcal{V}|}$,

$$Pr[\mathcal{A}(H_1) \in O] \leq e^{\varepsilon} \cdot Pr[\mathcal{A}(H_2) \in O] + \delta. \quad (1)$$

In our context, the DP guarantee informally means that from the *CSP*'s perspective, the observed noisy histogram, $\tilde{H}$, could have been generated from any histogram within a distance d from the original histogram, H. We define the set of all such histograms to be the ε-indistinguishability neighborhood for H. In other words, from $\tilde{H}$ the *CSP* will not be able to distinguish between T_S^{CSP} and any other transcript that differs from T_S^{CSP} in d words from $\mathcal{V}$.

An important result for differential privacy is that any post-processing computation performed on the output of a differentially private algorithm does not cause any loss in privacy.

Theorem 4.1. *(Post-Processing) Let $\mathcal{A} : \mathcal{X} \mapsto R$ be a randomized algorithm that is (ε,δ)-DP. Let $f : R \mapsto R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{A} : \mathcal{X} \mapsto R'$ is (ε,δ)-DP.*

Another result is that the privacy of DP-mechanism can be amplified if it is preceded by a sampling step.

Theorem 4.2. *Let $\mathcal{A}$ be an (ε,δ)-DP algorithm and $\mathcal{D}$ is an input dataset. Let $\mathcal{A}'$ be another algorithm that runs $\mathcal{A}$ on a random subset of $\mathcal{D}$ obtained by sampling it with probability β. Algorithm $\mathcal{A}'$ will satisfy (ε',δ')-DP where $\varepsilon' = ln(1 + \beta(e^{\varepsilon} - 1))$ and $\delta' < \beta\delta$.*

Additionally, we define a DP mechanism namely the truncated Laplace mechanism [6] which is used in Prεεch.

Definition 4.2 (Truncated Laplace mechanism for histogram)**.** Given a histogram H, the truncated Laplace mechanism, $Lp(\varepsilon,\delta,d)$, adds a non-negative integer noise vector $[\max(\eta,0)]^{|\mathcal{V}|}$ to H, where η follows a distribution, denoted by $L(\varepsilon,\delta,d)$ with a p.d.f $\Pr[\eta = x] = p \cdot e^{-(\varepsilon/d)|x-\eta^0|}$, where $p = \frac{e^{\varepsilon/d}-1}{e^{\varepsilon/d}+1}$ and $\eta_0 = -\frac{d \cdot \ln((e^{\varepsilon/d}+1)\delta)}{\varepsilon} + d$.

Theorem 4.3. *The truncated Laplace mechanism satisfies (ε,δ)-DP for d-distant histogram releases [6].*

(a) Original (b) ε=1, δ=0.05, and d=2 (c) ε=1, δ=0.05, and d=5 (d) ε=1, δ=0.05, and d=10

Figure 3: The word cloud of the Facebook dataset visualizing the histogram as it changes after adding different levels of noise.

Fig. 3 visualizes the histogram of the Facebook dataset as a word cloud for different noise levels. As evident from the original word cloud, the histogram emphasizes few important words such as Facebook, people, information, and users. With increased value of d, the resulting histogram has a roughly uniform distribution of the included words.

4.5.2 Discussion

Prεεch's use of DP is different from the most standard use-case of DP (like numeric datasets). It deals with concrete units like words instead of numeric statistics – introducing new challenges; we discuss these challenges and how Prεεch circumvents them in this section.

Vocabulary definition: The foremost task for defining the word histogram is defining the vocabulary, $\mathcal{V}$. The most conservative approach to define $\mathcal{V}$ is to consider the total set of all English stemmed and non-stop words. Such a vocabulary would be prohibitively large for efficient and practical usage. However, note that such a definition of $\mathcal{V}$ is an overestimate as no real-world document would contain all possible English words. Recall that our objective of adding noise is to obfuscate any statistical analysis built on top of the document's *BoW* (histogram), such as a topic modeling and stylometry analysis. Typically, *BoW* based statistical analyses are concerned only with the set of most frequent words. For example, any standard topic model captures only the top m percentile most frequent words in a transcript [37, 43]. The same applies to stylometry analysis, which is based on measures of the unique distribution of frequently used words of different individuals.

Thus, as long as the counts of the most common words of the transcript are protected (via DP), the subsequent statistical model (like topic model) built over the word histogram will be privacy-preserving too (by Thm. 4.1). However, high-frequency words might not be the only ones that contain important information about T_S. To tackle this, we also include words with large Term Frequency-Inverse Document Frequency (TF-IDF) weight to our vocabulary. This weight is a statistical measure used to evaluate how significant a word is to a document relative to a baseline corpus. The weight increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the baseline corpus. This offset adjusts for the fact that some words appear more frequently in general. To this end, Prεεch makes an estimate of the vocabulary from T_S^{OSP}. Although existing offline transcribers have high WER, we found (empirically) that they can identify the set of domain words of S with high accuracy (details in Sec. 7.3). For computing the TF-IDF values, IDF is computed using an external NLP corpus like Wikipedia articles. Thus formally, $\mathcal{V} = \{w | w \in \{$ top m percentile of the most frequent words in $T_S^{OSP}\} \cup \{$ words with TF-IDF value $\geq \Delta$ in $T_S^{OSP}\}\}$. Note that $\mathcal{V}$ should be devoid of all sensitive words which are scrubbed off from S in step 2 of Fig. 1. Additionally, the vocabulary can be extended to contain *out-of-domain* words, i.e., random English words that are not necessarily part of the original document. This helps in protecting against text classification attacks (Sec. 7.3).

Specificities of the word histogram: As discussed above, the goal of the DP mechanism is to generate noisy counts for each $w_i \in \mathcal{V}$. An artifact of our setting is that this noise has to be non-negative and integral. This is because dummy words (for the noisy counts) can only be added to S; removing any word from S is not feasible as this would entail in recognizing the word directly from S, which would require accurate transcription. Hence, Prεεch uses the truncated Laplace mechanism to ensure non-negative and integral noise.

Setting privacy parameters: The parameters ε and δ quantify the privacy provided by a DP-mechanism; lower the values higher is the privacy guarantee achieved. The distance parameter d, intuitively, connects the privacy definition in the word histogram, which is purely a formal representation, to a semantic privacy notion. For example, it can quantify how much the noisy topic models computed by the CSP (from T_S^{CSP}) should differ from that of T_S^g. Thus, the user can tune d depending on the target statistical analysis. In the following, we detail a mechanism, as a guide for the user, for choosing d when the target statistical analysis is topic modeling.

Let us assume that the user has a set of speech files $\{S_j\}$ to be transcribed. Let D_j denote the ground truth transcript corresponding to speech file S_j. The objective is to learn t topics from the corpus $\bigcup_j D_j$ with at least k words per topic (a topic is a distribution over a subset of words from the corpus). Let $\mathcal{T} = \{\mathbb{T}_1, \cdots, \mathbb{T}_t\}$ represent the original topic model built on $\bigcup_j D_j = \bigcup_j T_{S_j}^g$ and $\mathcal{T}' = \langle \mathbb{T}'_1, \cdots, \mathbb{T}'_t \rangle$ represent the noisy topic model computed by the CSP.

The following theorem (Thm. 4.4) provides a lower bound on the pairwise ℓ_1 distance between the true and noisy topics as a function of the privacy parameters of the DP word

histogram release mechanism (specifically, the term C_{min} is a function of (d, ε, δ)).

Theorem 4.4. *For any pair of topics* $(\mathbb{T}, \mathbb{T}') \in \mathcal{T} \times \mathcal{T}'$,

$$||T - T'||_1 \geq 2 \frac{1}{\left(1-(t-1)\frac{k}{\max_j |D_j|}\right)} \left(\frac{C_{min}}{t} - \frac{1}{2}\left(1 - t\frac{k}{\max_j |D_j||}\right)\right),$$

where $C_{min} = min_{j,l}\left\{ \frac{v\cdot(|D_j|-|w_{l,j}|\omega_j)}{|D_j|\cdot(|D_j|+v\cdot\omega_j})\right\}$, $|D_j|$ *is the total number of words in* D_j, ω_j *is the total number of unique words,* v *is the variance of the distribution* $Lp(\varepsilon', \delta', d)$, $\delta' = \beta\delta$ *and* $|w_{l,j}|$ *is the number of times the word* $w_l \in \mathcal{V}$ *appears in* D_j.

The proof of this theorem and the descriptions of the parameters are presented in the full paper [3] .

Dummy word injection: As discussed earlier, achieving differential privacy requires adding dummy words to S. Prεεch generates the dummy text corpus using an NLP language model (Sec. 6). The model takes in a short text sample from the required topic and generates an entire document of any required length based on that input. In some scenarios, the user can also provide a corpus of non-publicly available documents with the same vocabulary. This scenario is valid in many practical settings. For instance, in an educational institution, the sensitive speech files requiring transcription might be the interviews/oral exams of the students conducted on a specific subject, and the noise corpus can be the lecture notes of the same subject.

Next, Prεεch generates a set of dummy segments, $\mathbb{S}_d$, from the dummy corpus above. Let us assume that each of the true segments contains at most k non-stop words (depends on the segment length). Prεεch ensures that each dummy segment also contains no more than k non-stop words. Additionally, each such segment must contain only one word from the vocabulary $\mathcal{V}$. This means that although the physical noise addition is carried at the segment level, it is still equivalent to adding noise at the level of words (belonging to $\mathcal{V}$) as we only care about $w_i \in \mathcal{V}$. Each dummy segment is injected only once per CSP. Since the dummy segments have to be added in the speech domain, Prεεch applies TTS transforms to the segments in $\mathbb{S}_d$ such that they have the same acoustic features as $\mathbb{S}$. This condition ensures that $\mathbb{S}_d$ are indistinguishable from $\mathbb{S}$ in terms of their acoustic features. Prεεch provides the user with two broad options to satisfy this condition – voice cloning or voice conversion.

Voice cloning is a TTS system that generates speech in a target speaker voice. Given a speech sample from the target speaker, the system generates an embedding of the speaker's voice biometric features. It uses this embedding to synthesize new utterances of any linguistic content in the target speaker's voice. Prεεch utilizes such a technology to clone the original speaker's voice and uses it to generate acoustically similar dummy segments $\mathbb{S}_d$. Prεεch applies a state-of-the-art voice cloning system [20], which generates a close-to-natural synthetic voice using a short ($\sim$ 5 sec.) target voice sample.

We evaluate this cloning system in Sec. 3.1, and the cloned samples are successfully identified as the true speakers. However, voice cloning does not protect the speakers' voice biometrics, and can be potentially thwarted by a stronger adversary. Hence, Prεεch provides voice conversion (VC) as a stronger privacy-preserving option for the user. VC transforms the voice of a source speaker to sound like a target speaker. Prεεch utilizes VC to obfuscate the true speakers' voice biometrics as well as to mitigate the DP noise indistinguishability concern by converting the true and dummy segments into a single target speaker voice (Sec. 4.6). We discuss the utility-privacy trade-offs of both options in Sec. 7.

It is important to note that the dummy segments do not affect the WER of T_S^{CSP}. It is so because Prεεch can exactly identify all such dummy segments (from their timestamps) and remove them from T_S^{CSP}. Additionally, since the transcription is done one segment at a time, the dummy segments do not affect the accuracy of the true segments ($\mathbb{S}$) either. Segmentation and voice conversion are the culprits behind the WER degradation, as will be evident in Sec. 7. Thus in Prεεch, the noise (in the form of dummy segments) can ensure differential privacy without affecting the utility. This is in contrast to standard usage of differential privacy for releasing numeric statistics where the noisy statistics result in a clear loss of accuracy. However, the addition of the dummy segments in Prεεch does increase the monetary cost of using the online service that has to transcribe more speech data than needed. We analyze this additional cost in Sec. 7.

In practice, we have multiple well-known cloud-based transcription services with low WER like Google Cloud Speech-to-Text, Amazon Transcribe, etc. Prεεch uses them to its advantage in the following way. Prεεch splits the set of segments $\mathbb{S}$ into N different sets (step 3 in Sec. 4.5.3) $\mathbb{S}_i, i \in [N]$ where N is the number of CSPs with low WER. Then, Prεεch sends each subset to a different CSP (after adding suitable noise segments to each set and shuffling them). Since each engine is owned by a different, often competing corporation, it is reasonable to assume that the CSPs are *non-colluding*. Thus, assuming that each segment contains at most one word in $\mathcal{V}$, each subset of segments $\mathbb{S}_i$ can be viewed as randomly sampled sets from $\mathbb{S}$ with sampling probability $\beta = 1/N$. From Thm. 4.2, this partitioning results in a privacy amplification.

4.5.3 Mechanism

We summarize the DP mechanism by which Prεεch generates the dummy segments for S. The inputs for the mechanism are (1) $\mathbb{S}$ – the short segments of the speech file S, (2) the privacy parameters ε and δ and (3) N – the number of non-colluding CSPs to use. This mechanism works as follows:

- Identify the vocabulary $\mathcal{V} = \{w | w \in \{$ top m percentile of the most frequent words in $T_S^{OSP}\} \cup \{$ words with TF-IDF value $\geq \Delta$ in $T_S^{OSP}\}\}$ through running an offline transcriber over S.

- Tune the value of d based on the lower bound from Thm. 4.4, ε and δ.
- Generate N separate noise vectors, $\eta_i \sim [Lp((\ln(1+\frac{1}{\beta}(e^{\varepsilon}-1)), \beta\delta, d)]^{|\mathcal{V}|}, i \in [N]$. Thus for every partition i, Prεech associates each word in $\mathcal{V}$ with a noise value, a non-negative integer.
- From the NLP generated text, extract all the text segments that contain words from $\mathcal{V}$. For each partition i, sample the text segments from this corpus to match the noise vector η_i. This is the set of noise (dummy) segments for partition i, $\mathbb{S}_{d,i}$. Iterate on generating text from the NLP language model until the required noise count is satisfied.
- Randomly partition $\mathbb{S}$ into N sets $\mathbb{S}_i, i \in [N]$ where $\Pr[\text{segment } s \text{ goes to partition } i] = \beta = 1/N, s \in \mathbb{S}$.
- For each partition $i \in [N]$, shuffle the dummy segments in $\mathbb{S}_{d,i}$ (after applying TTS and VC) with the segments in $\mathbb{S}_i$ (after applying VC), and send it to the CSP_i.

The first 4 steps in the above mechanism are performed in stage 3 in Prεech (Fig. 1) while steps 5-6 are performed in stage 6.

Theorem 4.5. *Any topic model computed by $CSP_i, i \in [N]$ from $T_S^{CSP_i}$ is (ε, δ)-DP.*

Proof. From Thm. 4.2 and Thm. 4.3, we conclude that the word histogram $\tilde{H}_i$ computed from $T_S^{CSP_i}$ is (ε, δ) - DP for distance d. Thm. 4.1 proves that the topic model from $\tilde{H}_i$ is still (ε, δ)-DP as it is a post-processing computation. □

4.5.4 Novelty of Prεech's Use of Differential Privacy

Here, we summarize the key novelty in Prεech's use of DP:
(1) Typically, DP is applied to statistical analysis of numerical data where "noise" corresponds to numeric values. In contrast, in Prεech, "noise" corresponds to concrete units – words. To tackle this challenge, we applied a series of operations (segmentation, shuffling, and partitioning) to transform the speech transcription into a *BoW* model, where the DP guarantee can be achieved. Moreover, the noise addition has to be done in the speech domain. This constraint results in new challenges: the lack of a priori access to the word histogram domain $\mathcal{V}$, and generating indistinguishable dummy speech segments.
(2) In our setting, the use of a DP mechanism does not introduce a privacy-utility trade-off from the speech transcription standpoint. Prεech performs transcription one segment at a time. It keeps track of the timestamps of the dummy segments and completely removes their corresponding text from the final transcription (Sec. 4.2.3). This filtration step is achievable in Prεech, unlike numeric applications of DP, because of the atomic nature of transcription. However, the dummy segments increase the monetary cost of transcription, resulting in a privacy-monetary cost trade-off as shown in Table 3. To tackle this issue, Prεech takes advantage of the presence of multiple CSPs (Sec. 4.5.2). Thus, the idea of utilizing multiple CSPs for cost reduction (Thm. 4.2) is a novel contribution.
(3) We introduce an additional parameter d, the distance between the pair of histograms, in our privacy definition (Defn. 4.1). Intuitively, d connects the privacy definition in the word histogram model, which is purely a formal representation, to a semantic privacy notion (e.g., ℓ_1 distance between true and noisy topic models, Thm. 4.4) as shown in Fig. 6 and 7. This contribution builds on ideas like group privacy [13] and generalized distance metrics [10].

4.5.5 Control Knobs

The construction of the DP word histogram provides the user with multiple control knobs for customization:
Parameter d: According to Def. 4.1, from $\tilde{H}$ the *CSP* will not be able to distinguish between T_S^{CSP} and any other transcript that differs from T_S^{CSP} in d words from $\mathcal{V}$. Thus, higher the value of d, larger is the ε-indistinguishability neighborhood for $\tilde{H}$ and hence, better is the privacy guarantee. But it results in an increased amount of noise injection (hence, increased monetary cost – details in Sec. 7.3).
Vocabulary: The size of $\mathcal{V}$ is a control knob, specifically, the parameters m and Δ and the number of *out-of-domain* words. The trade-off here is: the larger the size of $\mathcal{V}$, the greater is the scope of the privacy guarantee. However, the noise size scales with $|\mathcal{V}|$ and hence incurs higher cost (details in Sec. 7.3).
Voice transformation for noisy segments: Prεech provides two options for noise synthesis – voice cloning and voice conversion. Voice cloning does not affect the transcription utility, measured in WER, because it does not apply any transformations on the original speaker's voice. However, it fails to protect the sensitive biometric information in S. Moreover, there is no guarantee that a strong adversary cannot develop a system that can distinguish the cloned speech segments from the original ones. This puts Prεech's effectiveness at the risk of the arms race between the voice cloning system's performance and the adversary's strength. This limitation is addressed by voice conversion at the cost of transcription utility. We quantify these utility-privacy trade-offs in Sec. 7.
Number of CSPs used for transcription: As discussed above, employing multiple CSPs lowers the monetary cost incurred. However, as shown in Table 1, AWS has a higher WER than Google. Hence, using both the CSPs results in lower overall utility than just using Google's cloud service.

4.6 Voice Conversion

Below, we discuss the two main categories of VC systems, highlighting their privacy-utility trade-offs.

4.6.1 One-to-One Voice Conversion

One-to-one VC maps a predefined source speaker voice to a target speaker voice. In Prεech, we use sprocket [21], which is based on spectral conversion using a Gaussian mixture

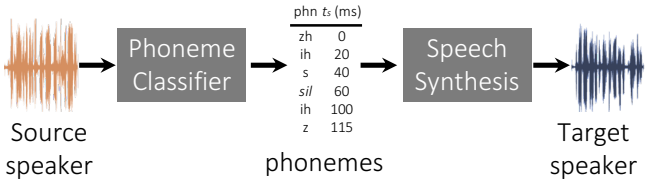


Figure 4: An illustration of the many-to-one VC pipeline.

model (GMM). Sprocket's training phase takes three steps: (1) acoustic features extraction of the source and target speakers samples, (2) time-alignment of the source and target features, and (3) GMM model training. During conversion, sprocket extracts the acoustic features of the new utterances, converts them using the learned GMM model, and generates the target waveform. Prεεch applies sprocket to convert the voice of all source speakers, including the synthesized dummy segments, into the *same target speaker voice*.

4.6.2 Many-to-One Voice Conversion

For perfect voice privacy, the VC system should (1) map any voice (even if previously unseen) to the same target voice, (2) not leak any distinguishing acoustic features, and (3) operate on speech containing multiple speakers. To this end, Prεεch deploys the two-stage many-to-one VC [44] mechanism. As shown in Fig. 4, the first stage is a phoneme classifier that transfers the speech utterance into phonetic posterior grams (PPG) matrix. A PPG is a time-aligned phonetic class [44], where a phoneme is the visual representation of a speech sound. Thus, the phoneme classifier removes the speaker-identifying acoustic features by mapping the spoken content into speaker-independent labels. In the second stage, a speech synthesizer converts the PPGs into the target voice.

The PPGs intermediate stage is irreversible and speaker-independent. It guarantees that the converted dummy segments $\mathbb{S}_d$ and converted original segments $\mathbb{S}$ cannot be distinguished from each other. However, the actual implementation of the system carries many challenges. The first stage is a performance bottle-neck as it needs large phonetically aligned training data to generalize to new unseen voices. We overcome this challenge by generating a custom training speech dataset with aligned phonemes as described in Sec. 6.

4.6.3 Control Knobs

The aforementioned VC techniques present an interesting utility-usability-privacy trade-off. The one-to-one VC technique gives better accuracy than many-to-one VC since it is trained for a specific predefined set of source speakers (details in Sec. 7.4.1). However, this utility gain comes at the price of usability and privacy. First, unlike many-to-one VC, sprocket needs parallel training data – a set of utterances spoken by both the source and target speakers. Hence, it requires an enrollment phase to get the source speaker's voice samples, thereby limiting the scalability of Prεεch for previously unseen speakers. Second, one-to-one VC does not provide perfect indistinguishability. These two limitations are mitigated by applying many-to-one VC (Sec. 7.4.1).

5 End-to-End Threat Analysis

In this section, we go over the end-to-end system design of Prεεch and identify potential privacy vulnerabilities.

Voice Privacy: Many-to-one VC removes all the identifying features from S, like the speakers' voices, background noise, and recording hardware, thereby protecting voice privacy.

Textual Privacy: For sensitive word scrubbing, the best-case scenario from a privacy point of view is to have the user spell out the entire keyword list. However, due to its high usability overhead, Prεεch uses NER instead to identify named entities automatically from T_S^{OSP}. In Sec. 7.3.1, we empirically show that Prεεch can achieve near-perfect true positive rate in identifying the segments containing sensitive words. However, this is only an empirical result and is dataset dependent.

Our main defense against statistical analysis on the text is the DP guarantee on the word histogram. This DP guarantee would break down if the adversary can distinguish the dummy segments from the true segments. Many-to-one VC technique, by design, ensures that both sets of segments have the same acoustic features. However, the possibility of distinguishing them based on their textual features still remains. To address this threat, we rely on state-of-the-art NLP models with low perplexity (log-likelihood) scores to generate the dummy text corpus. The low perplexity scores ensure that the auto-generated text is as close as possible to the natural language generated by humans [19, 36]. Although there is no formal guarantee about the adversary's ability to distinguish dummy and true segments based on their textual features, we have empirically analyzed this threat in Sec. 7.3.3 and Sec. 7.3.4. We leverage state-of-the-art NLP techniques to mount attacks on the dummy segments. Our results show that the adversary fails to distinguish between the dummy and true segments. However, the extent of such robustness is based on the efficacy of state-of-the-art NLP techniques.

Word correlations can also weaken the DP guarantee ($d-w$, if w is the maximum size of word groups with high correlation). This can be addressed by either increasing d or considering n-gram ($n = w$) word histograms. However, this would increase the requisite amount of dummy segments.

Long segments can also be a source of privacy vulnerability as each segment contains more contextual information. Hence, in the prototype Prεεch presented in the paper, we use short segments that contain at most two non-stop words.

Another weakness is related to vocabulary estimation, especially if some of the distribution-tail words are deemed to be sensitive. Prεεch provides no formal guarantees on the words that do not belong to $\mathcal{V}$. Although our empirical evaluation shows that the OSP has a very high accuracy for the weighted estimation of $\mathcal{V}$ (Sec. 7.3.2), some sensitive distribution-tail words might still be missed due to the OSP's transcription errors. Additionally, our formal DP guarantee holds only for the word histogram (*BOW*) on $\mathcal{V}$. Textual analysis models other than *BOW* are empirically evaluated in Sec. 7.3.3 and Sec. 7.3.4.

Finally, if the CSP can reorder the segments (even partially since the speech file it receives contains dummy segments as well), it will be able to distinguish the dummy segments from the true ones and hence, learn the textual content of the file. For this again, we show empirically that current NLP techniques fail to reorder the segments (Sec. 7.3.4) even in the worst-case setting where all the segments go to one CSP. However, as before, this is an empirical result only.

Formal Privacy Guarantee: *For a speech file S, Prεεch provides perfect voice privacy (when using many-to-one VC) and an* (ε, δ)*-DP guarantee on the word histogram for the vocabulary considered (BOW), under the assumption that the dummy segments are indistinguishable from the true segments.*

6 Implementation

In this section, we describe the implementation details of Prεεch's building blocks (shown in Fig. 1).

Segmentation: We implement the two-level hierarchical segmentation algorithm described in Sec. 4.3. The silence detection based segmentation is implemented using the Python pydub package[7]. We used Praat[8] to extract the pitch information required for the second level of the segmentation algorithm.

Sensitive Keyword Scrubbing: We use the NLP Python framework spaCy [9] for named entity recognition (NER) from the text. The keyword lists per each dataset can be found in the full paper [3] . We employ PocketSphinx[10] for keyword spotting, a lightweight ASR that can detect keywords from continuous speech. It takes a list of words (in the text) and their respective sensitivity thresholds and returns segments that contain speech matching the words. PocketSphinx is a generic system that can detect any keyword specified in runtime; it is not trained on a pre-defined list of keywords and requires no per-user training or enrollment.

Generating Dummy Segments: We use the open source implementation [11] of OpenAI's state-of-the-art NLP language model, *GPT*2 [36], to generate the noise corpus.

Using this predictive model, we generate a large corpus representing the vocabulary of the evaluation datasets. An example of the generated text is available in the full paper [3] . To generate the dummy segments, we segment each document at the same level as the speech segmentation algorithm. We build a hash table associating each vocabulary word with the segments that contain it. Prεεch uses a dummy segment only once per CSP to prevent it from identifying repetitions.

Text-to-Speech: We use the multi-speaker (voice cloning) TTS synthesizer [20] to generate the speech files corresponding to the dummy segments. We use a pre-existing system implementation and pretrained models [12].

One-to-One Voice Conversion: We use the open-source sprocket software [13]. As described in Sec. 4.6.1, sprocket requires a parallel training data and the target voice should be unified for all source speakers. For the VCTK datasets, we use speaker p306 as the target voice. Since we also evaluate Prεεch on non-standard datasets (Facebook and Carpenter cases), we had to construct the parallel training data for their source speakers. For this, we use TTS to generate the required target voice training utterances in a single *synthetic* voice.

Many-to-One Voice Conversion: We utilize pre-existing architectures and hyperparameters [14] for the two-stage many-to-one VC [44] mechanism, shown in Fig. 4. The first network, net_1, is trained on a set of {raw speech, aligned phoneme labels} samples from a multi-speaker corpus, where the labels are the set of 61 phonemes from the TIMIT dataset. The only corpus that has a manual transcription of speech to the phonemes' level is the TIMIT dataset – a limited dataset. We found that training net_1 on TIMIT alone results in an inferior WER performance. For better generalization, we augment the training set by automatically generating phoneme-aligned transcriptions of standard ASR corpora. We use the Montreal Forced Aligner [15] to generate the aligned phonemes on LibriSpeech and TED-LIUM [38] datasets. The second network, net_2, synthesizes the phonemes into the target speaker's voice. It is trained on a set of {PPGs, raw speech} pairs from the target speaker's voice. We use the *trained* net_1 to generate the PPGs data for training net_2. As such, we only need speech samples of the target speaker to train net_2. This procedure also allows net_2 to account for net_1's errors. We use Ljspeech[16] as the target voice for its relatively large size – 24 hours of speech from a single female.

7 Evaluation

We evaluate how well Prεεch meets the design objectives of Sec. 4. Specifically, we aim to answer the following questions:

[7] https://pypi.org/project/pydub/
[8] http://www.fon.hum.uva.nl/praat/
[9] https://github.com/explosion/spaCy
[10] https://github.com/cmusphinx/pocketsphinx
[11] https://github.com/huggingface/transformers
[12] https://github.com/CorentinJ/Real-Time-Voice-Cloning
[13] https://github.com/k2kobayashi/sprocket
[14] https://github.com/andabi/deep-voice-conversion
[15] https://montreal-forced-aligner.readthedocs.io/en/latest/
[16] https://keithito.com/LJ-Speech-Dataset/

Datasets	Cloning	One-to-One	Many-to-One	OSP
VCTK p266	5.15 (80.73%)	16.55 (38.06%)	21.92 (17.96%)	26.72
VCTK p262	4.53 (71.63%)	7.39 (53.73%)	10.82 (32.25%)	15.97
Facebook1	8.26 (66.59%)	14.60 (40.94%)	20.30 (17.88%)	24.72
Facebook2	9.75 (63.36%)	18.27 (31.34%)	19.44 (26.94%)	26.61
Facebook3	14.93 (51.40%)	23.25 (24.32%)	27.06 (11.91%)	30.72
Carpenter1	14.43 (44.18%)	23.88 (7.62%)	22.63 (12.46%)	25.85
Carpenter2	13.53 (65.93%)	33.71 (15.11%)	38.90 (2.04%)	39.71

Table 2: WER (%) of end-to-end Preech which represents the accumulative effect of segmentation, SWS, and different settings of voice privacy and its relative improvement in (%) over OSP (Deep Speech).

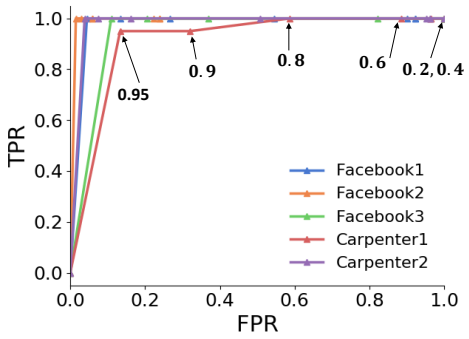


Figure 5: ROC curve for sensitive words detection at different values of the sensitivity score.

(*Q1.*) Does Preech preserve the transcription utility?
(*Q2.*) Does Preech protect the speakers' voice biometrics?
(*Q3.*) Does Preech protect the textual content of the speech?
(*Q4.*) Does the different control knobs provide substantial flexibility in the utility-usability-privacy spectrum?

We answer the first three questions for a prototype implementation of Preech that provides the maximum degree of formal privacy and hence, the least utility. For evaluating *Q4*, we relax the privacy guarantee to obtain utility and usability improvements.

Prototype Preech: For the prototype Preech presented in the paper: (1) segmentation length is adjusted to ensure that each segment contains at most two non-stop words (2) noisy segments are generated via the GPT2 language model (3) a single CSP (Google) is utilized (4) many-to-one VC is applied to both the dummy and true segments.

7.1 Q1. Transcription Utility

We assess the transcription WER after deploying end-to-end Preech on the non-standard datasets. Recall that Table 1 in Sec. 2.2 shows the baseline WER performance of the CSP and OSP before applying Preech.

WER Analysis: Column 4 in Table 2 shows the end-to-end WER for the prototype Preech which represents the accumulative effect of segmentation, SWS, and many-to-one VC. Although VC is the main contributor to Preech's WER, as is evident from Sec. 7.4.1 and Sec. 7.3.1, there are two main observations. First, many-to-one VC is superior to Deep Speech. Specifically, Preech's relative improvement over Deep Speech ranges from 11.91% to 32.25% over the evaluation datasets (except for Carpenter2). Recall that we trained the VC system using standard ASR corpora, while we evaluate the WER on non-standard cases. Still, Preech's WER is superior to that of Deep Speech, which has been trained through hundreds of hours of speech data. Second, Preech does not have the same performance for all the datasets. This observation arises again from the lack of diversity in our VC training set. For example, the speaker in Carpenter 1 speaks loudly, allowing VC to perform well. On the other hand, the second speaker (Carpenter 2) is not as clear or loud, which results in an inferior VC performance. This observation is consistent with Deep Speech as well.
Our experiments show that these results can be improved by adding samples of the source speaker voice to the training pipeline of net_1 and net_2. We chose not to go with this approach as this limits the usability of the system, and in such a case sprocket (Sec. 7.4.1) would be a better choice.

7.2 Q2. Voice Biometric Privacy

To test the voice biometric privacy, we conduct two experiments using the voice analysis APIs (details in Sec. 3.1). In the first experiment, we assess the CSP's ability to separate speech belonging to different speakers after Preech applies the VC system. On our multi-speaker datasets, IBM diarization API concludes that there is only one speaker present.

Furthermore, we run the diarization API after adding the dummy segments (after TTS and VC). Again, the API detects the presence of only one speaker. Thus, not only does Preech hide the speaker's biometrics and map them to a single target speaker but also ensures noise indistinguishability, which is key to its privacy properties.

The second experiment tests Preech's privacy properties against a stronger adversary, who has access to samples from the true speakers. We enroll segments from the true speakers as well as the fake target speaker to Azure's Speaker Identification API. We pass the segments from Preech (after adding dummy segments and applying VC) to the API. When many-to-one VC is applied, in all evaluation cases, the API identifies the segments as belonging to the fake target speaker. Not a single segment was matched to the original speaker. Both experiments show that prototype Preech is effective in sanitizing the speaker's voice and ensuring noise indistinguishability.

Datasets	$\|\mathcal{V}\|$	# words in T_S^g	#Extra words due to dummy segments d=2	d=5	d=15
VCTK p266	483	922 ($0.22)	2915 ($0.68)	7247 ($1.69)	23899 ($5.58)
VCTK p262	471	914 ($0.21)	2845 ($0.66)	7157 ($1.67)	23230 ($5.42)
Facebook	1098	5326 ($1.24)	6660 ($1.55)	16567 ($3.87)	54038 ($12.62)
Carpenter	1474	7703 ($1.80)	8915 ($2.08)	22296 ($5.20)	72907 ($17.02)

Table 3: Number of extra words due to dummy segments and the additional monetary cost in USD with varying d, at $\varepsilon = 1$ and $\delta = 0.05$.

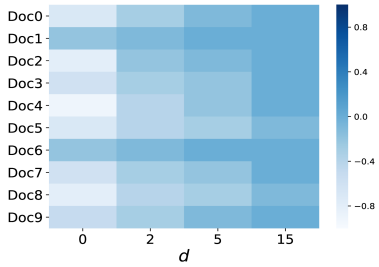


Figure 6: Sentiment scores heatmap of 10 documents with varying d, at $\varepsilon = 1$ and $\delta = 0.05$.

7.3 Q3. Textual Privacy

We perform an extensive evaluation of the textual privacy, including sensitive word scrubbing, analysis of the DP mechanism, and defense against statistical analysis.

7.3.1 Sensitive Words Scrubbing:

We run PocketSphinx keyword spotting on each dataset at different sensitivity scores ranging from 0.2 to 1[17]. Fig. 5 shows the detection true positive rate (TPR) versus the false positive rate (FPR) at different sensitivity scores. As the figure shows, the sensitivity score is a trade-off knob between privacy (high TPR) and utility (low FPR). We observe that Prεεch is able to achieve almost perfect TPR with low FPR values.

Next, we evaluate the impact of SWS on the transcription utility. We set a sensitivity score of 0.95 for all the datasets to have a near-perfect TPR while minimizing the FPR. Our experiments show that the total duration of the segments flagged with sensitive keywords at this score is: 0.13%, 0.06%, 0.18%, 0.20%, and 0.08% of the total duration of each dataset in Fig. 5. Then, we transcribe the sensitive-flagged segments using Deep Speech. The overall transcription accuracy after SWS (i.e., equivalent to choosing voice cloning in Prεεch as cloning results in no addition WER) is presented in the second column of Table 2. Since the segments are short, the portion of speech transcribed locally is limited. Hence, the impact of the OSP transcription errors is not significant.

7.3.2 DP Mechanism Analysis:

We follow the DP mechanism described in Sec. 4.5.3.

Vocabulary Estimation: We estimate the vocabulary $\mathcal{V}$ using the OSP transcript. Let $\mathcal{W}$ represent the set of unique words in T_S^g. We define the accuracy of the vocabulary estimation, D_{acc}, as the ratio between the count of the correctly identified unique words from T_S^{OSP}, $|\mathcal{W}|_{est}$, and the count of the unique words in T_S^g, $|\mathcal{W}|$. For our datasets, the domain estimation accuracy is at least 75.54%. We also calculate the weighted estimation accuracy defined as: $D_{weighted} = \frac{\sum P(w_{est}).\mathbb{1}_{w_{est}\in\mathcal{W}}}{|\mathcal{W}|}$ where $P(w_{est})$ is the weight of the estimated word w_{est} in T_S^g. $D_{weighted}$ is more informative since it gives higher weights to the most frequent words in T_S^g. The weighted estimation accuracy is 99.989% in our datasets. From $\mathcal{W}_{est}$ we select $\mathcal{V}$ over which we apply the DP mechanism. Additionally, we extend our vocabulary to contain a set of random words from the English dictionary.

Histogram Distance: We analyze the distance between the original and noisy histograms (after applying Prεεch) and its impact on the cost of online transcription. Because of the nature of Prεεch's DP mechanism, the noise addition depends on four values only: $|\mathcal{V}|$, ε, δ, and d.

For all our experiments, we fix the values of $\varepsilon = 1$ and $\delta = 0.05$. Table 3 shows the amount of noise (dummy words) and their transcription cost in USD [18] for each of the evaluation datasets at different values of d. Each dataset has a different vocabulary size $|\mathcal{V}|$ and word count. The increase in the vocabulary size requires adding more dummy segments to maintain the same privacy level. In Prεεch, adding more noise comes at an increased monetary cost, instead of a utility loss. The table highlights the *trade-off* between privacy and the cost of adding noise.

7.3.3 Statistical Analysis

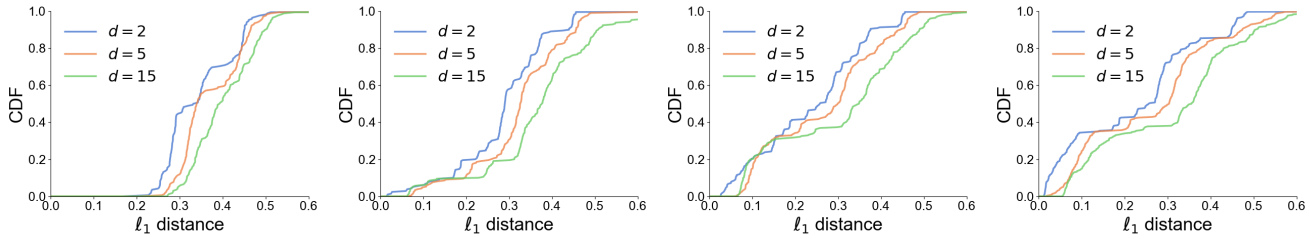
In this section, we evaluate the statistical analyses (details in Sec. 3.2) performed by the adversary to extract textual information on the noisy transcripts obtained from Prεεch.

Topic Model: We generate the topic models from the documents corresponding to the original and noisy word histograms, and evaluate their ℓ_1 distance. The topic model operates on a corpus of documents; hence we include eight more Supreme Court cases to our original evaluation datasets (Facebook and Carpenter). In this evaluation, we treat all these ten documents as one corpus; we aim to generate the topic model before and after applying Prεεch to the whole corpus.

We use AWS Comprehend API to generate the topic model. The API needs the number of topics as a hyperparameter that ranges from 1 to 100. Based on our apriori knowledge of the

[17] The sensitive keywords list for each dataset is in the full paper [3] .

[18] The pricing model of Google Speech-to-Text is: $0.009 / 15 seconds.



(a) 8 topics (b) 10 topics (c) 12 topics (d) 14 topics

Figure 7: Topics ℓ_1 distance CDF at d = 2, 5, and 15 for $t = 8$, 10, 12, and 14

true number of topics, we evaluate the topic model on the following number of topics $t = 8, 10, 12$, and 14.

We statistically evaluate the ℓ_1 distance between true and noisy topics. The topic model $\mathcal{T} = \{\mathbb{T}_1, \cdots, \mathbb{T}_t\}$ is a set of t topics where each $\mathbb{T}_i, i \in [t]$ is a word distribution. We use the Hungarian algorithm to match each noisy topic $\mathbb{T}'_i \in \mathcal{T}'$ to its closest match in $\mathcal{T}$, the true topic model. We evaluate the topics ℓ_1 distance for 21 runs. At each run, we generate a random noise vector per document, select the corresponding dummy segments, and evaluate the topic model on the set of original and noisy documents. Fig. 7 shows the empirical CDF of the topics ℓ_1 distance at different values of d. As the figure shows, the higher the distance parameter d, the larger is the ℓ_1 distance between true and noisy topics.

Stylometry: In this experiment, we assume that the CSP applies stylometry analysis on T_S^{CSP} in an attempt to attribute it to an auxiliary document whose authors are known to the CSP. To evaluate the worst-case scenario, we assume the adversary possesses the original document T_S^g, and we compute the ℓ_2 distance of the stylometric feature vectors generated from T_S^{CSP} w.r.t T_S^g.

First, we compute the ℓ_2 distance of T_S^{CSP} before applying Prεεch. The respective values for the Facebook and Carpenter datasets are 28.19 and 60.45. T_S^{CSP} differs from T_S^g in lexical features due to transcription errors and because the CSP generates the punctuation instead of the actual author.
Second, we apply Prεεch on the two datasets at different values of the distance parameter: $d = 0, 2, 5, 15$. The corresponding ℓ_2 distances for the Facebook (Carpenter) dataset equal: 73.14 (83.64), 328.80 (577.72), 947.58 (1629.79), and 2071.18 (3582.10). Note that the ℓ_2 distance at $d = 0$ shows the effect of segmentation and SWS only on obfuscating the lexical features. Clearly, adding the dummy segments increases the ℓ_2 distance. This is expected as most of the lexical features are obfuscated by the DP mechanism.

Category Classification: Google's NLP API can classify a document to a predefined list of 700+ document categories[19]. First, we run the classification API on the original documents from the topic modeling corpus. All of them classify as Law & Government. Running the API on Prεεch processed documents, using an extended-vocabulary (i.e., contains random words), dropped the classification accuracy to 0%. None of the documents got identified as legal, law, or government even at the smallest distance parameter value $d = 2$. Although a portion of the noise words belongs to the original Law & Government category, segmentation, shuffling, and the out-of-domain noise words successfully confuse the classifier.

Sentiment Analysis: Sentiment analysis generates a score in the $[-1, 1]$ range, which reflects the positive, negative, or neutral attitude in the text. First, we evaluate the sentiment scores of the original ten documents. For all of them, the score falls between -0.2 and -0.9, which is expected as they represent legal documents. Next, we evaluate the scores from Prεεch processed documents considering an extended-vocabulary. We find that all scores increase towards a more positive opinion. Fig. 6 shows a heatmap of the sentiment scores as we change the distance parameter d for the then evaluation documents. Thus, Prεεch's two-pronged approach – 1) addition of extended-vocabulary noise 2) removal of ordering information via segmentation and shuffling, proves to be effective. In a setting where the adversary has no apriori knowledge about the general domain of the processed speech, the noise addition mechanism gains extend from DP guarantee over the histogram to other NLP analyses as well.

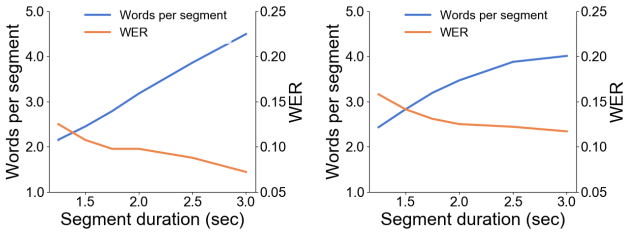
7.3.4 Indistinguishability Of Dummy Segments

The indistinguishability of the dummy segments is critical for upholding the DP guarantee in Prεεch. We perform two experiments to analyze whether current state-of-the-art NLP models can distinguish the dummy segments from their textual content.

Most Probable Next Segment: In this experiment, the adversary has the advantage of knowing a true segment $\mathbb{S}_t$ that is at least a few sentences long from the Facebook dataset. We use the state-of-the-art GPT [20] language model by OpenAI [35] to determine the most probable next segment following $\mathbb{S}_t$ using the model's perplexity score. In NLP, the perplexity score measures the likelihood that a piece of text follows the

[19] https://cloud.google.com/natural-language/docs/categories

[20] https://github.com/huggingface/transformers



(a) Facebook (b) Carpenter

Figure 8: Segmentation trade-off between utility and privacy. WER(%) is measured using Google Cloud Speech-to-Text.

language model. We get the perplexity score of stitching $\mathbb{S}_t$ to each of the other segments at the CSP. The segment with the lowest perplexity score is selected as the most probable next segment. We iterate over all the true segments of the Facebook dataset, selecting them as $\mathbb{S}_t$. We observed that a dummy segment is selected as the most probable next segment in 53.84% of the cases. This result shows that the language model could not differentiate between the true and dummy segments even when part of the true text is known to the adversary.

Segments Re-ordering: Next, we attempt to re-order the segments based on the perplexity score. We give the adversary the advantage of knowing the first true segment $\mathbb{S}_0$. We get the perplexity score of $\mathbb{S}_0$, followed by each of the other segments. The segment with the lowest score is selected as the second segment $\mathbb{S}_1$ and so on. We use the normalized Kendall tau rank distance K_τ to measure the sorted-ness of the re-ordered segments. The normalized K_τ distance measures the number of pairwise disagreements between two ranking lists, where 0 means perfect sorting, and 1 means the lists are reversed. The K_τ score for running this experiment on the Facebook dataset is 0.512, which means that the re-ordered list is randomly shuffled w.r.t the true order. Hence, our attempt to re-order the segments has failed.

These empirical results show that it is hard to re-order the segments or distinguish the dummy segments. This is expected due to three reasons: (1) the segments are very short; (2) the dummy segments are generated using a state-of-the-art language model; and (3) we observed that most of the transcription errors happen in the first and last words of a segment due to breaking the context. These errors add to the difficulty of re-ordering. Moreover, if the user partitions S among multiple CSP's (Sec.4.5.3), then consecutive segments would not go to the same CSP with high probability. This setting would increase Prεεch's protection against re-ordering attacks.

7.4 Q4: Flexibility of the Control Knobs

7.4.1 Utility-Privacy Trade-off

In this section, we empirically evaluate the controls knobs that provide a utility-privacy trade-off.

Minimum segment length: Fig. 8 shows the trade-off between the number of words per segment and WER as function of the minimum segment length. As expected, increasing the minimum duration of a segment results in an increase in the number of words per segment. The WER in turn drops when the number of words per segment increase as the transcription service has more textual context. However, it can lead to potential privacy leakage. The results in Fig. 8 indicate that for two real-world datasets, the number of words per segment can be kept between 2 and 3 with an acceptable degradation of the WER.

Voice Cloning: Voice cloning does not affect the true segments (it is only applied to dummy segments), resulting in no additional WER degradation. The WER for deploying voice cloning is incurred only due to segmentation and SWS. Thus, as shown in column 2 of Table 2, the relative improvement in WER ranges from 44% to 80% over Deep Speech. This approach, however, has two limitations. First, the speaker's voice biometrics from S are not protected. Second, there is no guarantee that an adversary would not be able to distinguish the cloned speech segments from the original ones.

Sensitivity score of KWS: As shown in Fig. 5, lower the sensitivity score, higher is the TPR and hence greater is the privacy (most prominent in the Carpenter2 dataset). However, this also increases the FPR, which means a larger number of non-sensitive segments are transcribed via the OSP resulting in reduced accuracy.

One-To-One VC: Table 2, column 3, shows that one-to-one VC outperforms many-to-one VC on most of the datasets. This result is expected since sprocket is trained and tested on the same set of source speakers while the many-to-one VC system generalizes to previously unseen speakers.

We observe that the improvement for the VCTK dataset is more significant than others. Recall that in our one-to-one VC implementation in Sec. 6, the target voice for VCTK is a natural voice – speaker p306. The target voice for the other datasets is a synthetic one, which hinders the quality of the converted voice and the transcription accuracy. We investigate this observation by training sprocket for VCTK on a synthetic target voice as well. The WER then increased to 19.33% and 9.21% for p266 and p262. Hence, we attribute the difference in the relative improvement to the target voice naturalness. In practice, the target voice could easily be a natural pre-recorded voice, and the users are asked to repeat the same utterances at the enrollment phase.

However, the one-to-one VC technique suffers from some privacy loss. The one-to-one VC system translates the acoustic features from a source to a target speaker's voice. Hence, it may leak some features from the source speaker. We observed that one-to-one VC is vulnerable to speaker identification analysis. Specifically, using Azure's Speaker Identification API, 10% of the voice-converted segments using sprocket were identified to their true speakers.

7.4.2 Usability-Privacy Trade-off

In our setting, usability can be measured along three dimensions: latency, monetary cost, and implementation overhead. However, we would like to stress that Preech is not designed for real-time speech transcription. Hence, latency is not a primary concern for Preech. Nevertheless, we include it in the following discussion for the sake of completeness.

Latency Evaluations: Note that all the operations of Preech are performed on speech segments. Hence, the latency is linear in the number of segments. We evaluate the end-to-end system latency per segment (with length $\sim$ 6s) for the OSP, the CSP, and Preech; the latency values are 2.17s, 1.70s, and 14.90s, respectively. We observe that the overhead of Preech is mostly attributed to the many-to-one VC (11s per segment on average). When voice cloning (or one-to-one VC) is applied instead, Preech's end-to-end per segment latency reduces to 3.90s (or 11.47s) at the expense of a privacy loss as discussed in Sec.7.4.1.

Vocabulary Size: Considering a larger $\mathcal{V}$ (Sec. 4.5.3) increases the scope of the DP guarantee. For example, adding external words provides protection against statistical analysis like text classification (Sec.7.3). However, larger $\mathcal{V}$ results in increased amount of dummy segments and hence, increased monetary cost (Table 3). For example, extending $\mathcal{V}$ by $\sim$ 1000 out-of-domain words for the Carpenter dataset incurred a total cost of \$25 at $d = 15$.

Distance Parameter d: As explained in Sec. 4.5.2, larger the value of d, greater is the scope of privacy. However, the amount of required noise increases by d. For example, for the dataset VCTK p266, increasing d from 2 to 15 increases the cost by roughly \$5 (Table 3).

7.4.3 Utility-Usability Trade-off

The following control knobs provide a venue for customizing the utility-usability trade-off.

Number of CSPs: As discussed in Sec. 4.5.2, using multiple CSPs reduces the amount of dummy segments (and hence, the monetary cost) in Preech. However, it comes at the price of utility; the transcription accuracy of the different available CSPs varies. For example, from Table 1, we observe that AWS has a higher WER than Google. Thus, using multiple CSPs may result in a lower mean utility.

One-to-One VC: As discussed above, one-to-one VC technique has lower WER than many-to-one VC technique (Table 2). However, it requires access to representative samples of the source speaker voice for parallel training thereby limiting scalability for previously unseen speakers (Sec. 4.6).

8 Related Work

In this section, we provide a summary of the related work.

Privacy by Design: One class of approaches redesigns the speech recognition pipeline to be private by design. For example, Srivastava et al. proposes an encoder-decoder architecture for speech recognition [42]. Other approaches address the problem in an SMC setting by representing the basic operations of a traditional ASR system using cryptographic primitives [32]. VoiceGuard is a system that performs ASR in the trusted execution environment of a processor [8]. However, these approaches require redesigning the existing systems.

Speech Sanitization: Recent approaches have considered the problem from a similar perspective as ours. They sanitize the speech before sending it to the CSP. One such approach randomly perturbs the MFCC, pitch, tempo, and timing features of a speech before applying speech recognition [45]. Others sanitize the speaker's voice using vocal tract length normalization (VTLN) [33, 34]. A recent approach modifies the features relevant to emotions from an audio signal, makes them less sensitive through a GAN [4]. Last, adversarial attacks against speaker identification systems can provide some privacy properties. These approaches apply minimal perturbations to the speech file to mislead a speaker identification network [9, 22].

These approaches are different from ours in two ways. First, they do not consider the textual content of the speech signal. The only exception is the approach by Qian et al. [34], which addresses the problem of private publication of speech datasets. This approach requires a text transcript with the audio file, which is not the case for the speech transcription task. In addressing the textual privacy of a speech signal, Preech adds indistinguishable noise to the speech file. The proposed techniques fail to provide this property. Second, the approaches above only consider voice privacy against a limited set of features, such as speaker identification or emotion recognition. Preech applies many-to-one VC to provide perfect voice privacy.

9 Conclusion

In this paper, we have proposed Preech, an end-to-end system for speech transcription that (1) protects the users' privacy along the acoustic and textual dimensions at (2) an improved performance relative to offline ASR, (3) while providing customizable utility, usability, and privacy trade-offs.

Acknowledgment

The work reported in this paper was supported in part by the NSF under grants 1661036, 1838733, 1942014, and 1931364. We also acknowledge Google for providing us with Google Cloud Platform credits and NVIDIA Corporation with the donation of the Quadro P6000 GPU used for this research. We would like to thank the anonymous reviewers for their useful comments and Micah Sherr for shepherding this paper.

References

[1] An all-neural on-device speech recognizer. https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html.

[2] Prεεch demo. https://bit.ly/2Vytbx7.

[3] S. Ahmed, A. R. Chowdhury, K. Fawaz, and P. Ramanathan. Preech: A system for privacy-preserving speech transcription. *arXiv preprint arXiv:1909.04198*, 2019.

[4] R. Aloufi, H. Haddadi, and D. Boyle. Emotionless: Privacy-preserving speech analysis for voice assistants. *arXiv preprint arXiv:1908.03632*, 2019.

[5] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[6] J. Bater, X. He, W. Ehrich, A. Machanavajjhala, and J. Rogers. Shrinkwrap: Differentially-private query processing in private data federations. *arXiv preprint arXiv:1810.01816*, 2018.

[7] P. Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Institute of Phonetic Sciences - University of Amsterdam*, 17:97–110, 1993.

[8] F. Brasser, T. Frassetto, K. Riedhammer, A.-R. Sadeghi, T. Schneider, and C. Weinert. Voiceguard: Secure and private speech processing. In *Interspeech*, pages 1303–1307, 2018.

[9] W. Cai, A. Doshi, and R. Valle. Attacking speaker recognition with deep generative models. *arXiv preprint arXiv:1801.02384*, 2018.

[10] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi. Broadening the scope of differential privacy using metrics. In E. De Cristofaro and M. Wright, editors, *Privacy Enhancing Technologies*, pages 82–102, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[11] R. Chen, N. Mohammed, B. C. Fung, B. C. Desai, and L. Xiong. Publishing set-valued data via differential privacy. *Proceedings of the VLDB Endowment*, 4(11):1087–1098, 2011.

[12] M. Davino. Assessing privacy risk in outsourcing. *Assessing Privacy Risk in Outsourcing/AHIMA, American Health Information Management Association*, 2004.

[13] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[14] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[15] A. Friedman and A. Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 493–502. ACM, 2010.

[16] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93, 1993.

[17] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[18] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[19] T. Hofmann. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*, 2013.

[20] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, z. Chen, P. Nguyen, R. Pang, I. Lopez Moreno, and Y. Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in Neural Information Processing Systems 31*, pages 4480–4490. Curran Associates, Inc., 2018.

[21] K. Kobayashi and T. Toda. sprocket: Open-source voice conversion software. In *Odyssey*, pages 203–210, 2018.

[22] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet. Fooling end-to-end speaker verification with adversarial examples. *ICASSP 2018*, Apr 2018.

[23] J. Lindberg and M. Blomberg. Vulnerability in speaker verification-a study of technical impostor techniques. In *Sixth European Conference on Speech Communication and Technology*, 1999.

[24] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling. The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. *arXiv preprint arXiv:1804.04262*, 2018.

[25] S. E. McGregor, P. Charters, T. Holliday, and F. Roesner. Investigating the computer security practices and needs of journalists. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 399–414, 2015.

[26] H. Muckenhirn, M. M. Doss, and S. Marcell. Towards directly modeling raw speech signal for speaker verification using cnns. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4884–4888. IEEE, 2018.

[27] K. S. R. Murty, B. Yegnanarayana, and M. A. Joseph. Characterization of glottal activity from speech signals. *IEEE signal processing letters*, 16(6):469–472, 2009.

[28] G. J. Mysore. Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech?—a dataset, insights, and challenges. *IEEE Signal Processing Letters*, 22(8):1006–1010, 2014.

[29] A. Nautsch, C. Jasserand, E. Kindt, M. Todisco, I. Trancoso, and N. Evans. The gdpr & speech data: Reflections of legal and technology communities, first steps towards a common understanding. *arXiv preprint arXiv:1907.03458*, 2019.

[30] S. Nutanong, C. Yu, R. Sarwar, P. Xu, and D. Chow. A scalable framework for stylometric analysis query processing. In *ICDM 2016*, pages 1125–1130. IEEE, 2016.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *ICASSP 2015*, pages 5206–5210. IEEE, 2015.

[32] M. A. Pathak, B. Raj, S. D. Rane, and P. Smaragdis. Privacy-preserving speech processing: cryptographic and string-matching frameworks show promise. *IEEE signal processing magazine*, 30(2):62–74, 2013.

[33] J. Qian, H. Du, J. Hou, L. Chen, T. Jung, X.-Y. Li, Y. Wang, and Y. Deng. Voicemask: Anonymize and sanitize voice input on mobile devices. *arXiv preprint arXiv:1711.11460*, 2017.

[34] J. Qian, F. Han, J. Hou, C. Zhang, Y. Wang, and X.-Y. Li. Towards privacy-preserving speech data publishing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1079–1087. IEEE, 2018.

[35] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[36] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[37] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP 2009*, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[38] A. Rousseau, P. Deléglise, and Y. Esteve. Ted-lium: an automatic speech recognition dedicated corpus. In *LREC*, pages 125–129, 2012.

[39] S. Safavi, M. Russell, and P. Jančovič. Automatic speaker, age-group and gender identification from children's speech. *Computer Speech & Language*, 50:141–156, 2018.

[40] B. Schuller and A. Batliner. *Computational paralinguistics: emotion, affect and personality in speech and language processing*. John Wiley & Sons, 2013.

[41] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. MüLler, and S. Narayanan. Paralinguistics in speech and language—state-of-the-art and the challenge. *Computer Speech & Language*, 27(1):4–39, 2013.

[42] B. M. L. Srivastava, A. Bellet, M. Tommasi, and E. Vincent. Privacy-Preserving Adversarial Representation Learning in ASR: Reality or Illusion? In *INTERSPEECH 2019*, Graz, Austria, Sept. 2019.

[43] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[44] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *ICME 2016*, pages 1–6. IEEE, 2016.

[45] T. Vaidya and M. Sherr. You talk too much: Limiting privacy exposure via voice input. In *International Workshop on Privacy Engineering (IWPE)*, 2019.

[46] C. Veaux, J. Yamagishi, K. MacDonald, et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[47] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li. Spoofing and countermeasures for speaker verification. *Speech Commun.*, 66(C):130–153, Feb. 2015.