Extracting Targeted Training Data from ASR Models, and How to Mitigate It

Ehsan Amid, Om Thakkar*, Arun Narayanan, Rajiv Mathews, Françoise Beaufays*

Google LLC

{eamid, omthkkr, arunnt, mathews, fsb}@google.com

Abstract

Recent work has designed methods to demonstrate that model updates in ASR training can leak potentially sensitive attributes of the utterances used in computing the updates. In this work, we design the first method to demonstrate information leakage about training data from trained ASR models. We design Noise Masking, a fill-in-the-blank style method for extracting targeted parts of training data from trained ASR models. We demonstrate the success of Noise Masking by using it in four settings for extracting names from the LibriSpeech dataset used for training a SOTA Conformer model. In particular, we show that we are able to extract the correct names from masked training utterances with 11.8% accuracy, while the model outputs some name from the train set 55.2% of the time. Further, we show that even in a setting that uses synthetic audio and partial transcripts from the test set, our method achieves 2.5% correct name accuracy (47.7% any name success rate). Lastly, we design Word Dropout, a data augmentation method that we show when used in training along with MTR, provides comparable utility as the baseline, along with significantly mitigating extraction via Noise Masking across the four evaluated settings.

Index Terms: Information extraction, Data augmentation

1. Introduction

Modern end-to-end Automatic Speech Recognition (ASR) systems are increasingly being trained to improve inference in the presence of background noise (e.g., [1, 2, 3]). Training such models often requires audio and transcripts of millions of utterances. Recent work has designed methods to demonstrate that model updates in ASR training can leak potentially sensitive attributes like labels [4] and speaker identity [5] of utterances used in computing the updates. Many prior works have also focused on demonstrating that trained language models (LMs) [6, 7, 8, 9, 10, 11] and image classification models [12, 13] are susceptible to *unintended memorization* of the rare or unique data that was part of the train set. Even models trained on large scale datasets can leak potentially sensitive information about their training data, and the above-listed works design various methods to demonstrate such leakage.

There can be various situations where extracting targeted parts of training data (e.g., specific words following a general structure) can leak the participants' privacy. For instance, obtaining meaningful extractions from a model for phrases like "Alice is infected with [...] disease", or "Bob is [...] years old" can leak sensitive information about the samples used for training it. In this work, we study if targeted parts of training data can be extracted from ASR models using only a query-access to the model. Note that any method requiring only a black-box access has wide applicability since it can operate on even deployed ASR models to showcase privacy leakages. We design a fill-in-the-blank style method called Noise Masking that can demonstrate the susceptibility of ASR models to leak memorized parts of their training data using only query-access to them. To our knowledge, Noise Masking is the first method to showcase leakage about training data from trained ASR models.

We evaluate our Noise Masking method by performing experiments on a state-of-the-art Conformer (L) model [14] trained on the benchmark LibriSpeech dataset [15]. Leveraging the fact that LibriSpeech is a corpus of read English audiobooks, and some minimal knowledge about the language in English books, we show that using Noise Masking on a trained Conformer model can leak memorized names from the dataset.

We also design a data augmentation method, Word Dropout, for training ASR models that we empirically show to provide robustness against Noise Masking when used for training with Multistyle TRaining (MTR) [1]. Further, we also show that training with Word Dropout provides comparable utility as a baseline not using Word Dropout.

Organization of the paper: In Section 2, we describe the details of our Noise Masking method, and provide an empirical evaluation of it. Next, in Section 3, we provide an evaluation of various training methods, including our Word Dropout method, towards mitigating Noise Masking. We show some detailed results for practitioners to analyze data leakage in Section 4. We state the conclusions of this work in Section 5.

2. Extraction via Noise Masking

2.1. The Noise Masking Method

We provide a general description of the Noise Masking method for extracting information about the training data given an ASR model. Assume that an ASR model M is trained over a dataset D consisting of training examples from some population $\mathcal{P}$, and query-access to M is made available. At its most basic level, to design a query via Noise Masking, an analyst requires:

1. $I(\mathcal{P})$, which denotes some target knowledge regarding the population. This knowledge can take various forms; e.g., a street name typically appears before the word "Street", or a number typically follows the word "Apartment", etc. The analyst may guess the existence of such target structures in the training data and move on to the next step.

2. T, a procedure for obtaining relevant transcript(s) given some target knowledge about a population. For example, with target knowledge about addresses, T could provide transcripts like "Alice lives in Apartment [...] on [...] Street".

3. U, a procedure for generating utterances given transcripts. For instance, a Text-To-Speech (TTS) system.

Submitted to Interspeech 2022.

*Equal contribution.

4. N, a procedure for incorporating noise into given utterances. For example, an audio of some music segment.

The analyst can use the target knowledge $I(\mathcal{P})$ to obtain a candidate transcript $t = T(I(\mathcal{P}))$. This can be subsequently used to generate an utterance $u = U(t)$, and incorporate noise into it as $q = N(u)$ for querying the model M with q and obtaining an output t'. The analyst can compare t and t' to evaluate for extractions from the training data D.

In many scenarios, it can be easy for an analyst to guess target knowledge based on the intended use-case(s) of an ASR model. For successful extractions, it can be crucial to get relevant transcripts with such knowledge. However, our method requires only a query-access to a trained model, which allows adaptive applications of it in many common settings.

2.2. Empirical Evaluation

Now, we provide an instantiation of our Noise Masking method, and conduct experiments to evaluate its performance.

Train Dataset: We use the LibriSpeech dataset[1] [15], which contains ∼1000 hours of English audiobook recordings by several speakers. The dataset contains numerous names from books, with a lot of them appearing after *title words* like 'mister', 'miss', 'missus', etc. For our experiments, we consider such names as an example of *sensitive* information: suppose using Noise Masking, a trained model reveals information about the names that appear in the LibriSpeech train set. Then in a real-world application, the model could leak similar sensitive information about its training data.

Model: We use the Conformer (L) architecture and the training method from [14] for training the baseline model on LibriSpeech. For all our experiments, we conduct our analysis on a checkpoint trained for ∼100k iterations.

Noise Masking for Name Extraction: Following the design in Section 2.1, for our implementation of Noise Masking we start with the target knowledge that in English books, the title 'mister' is often followed by names. For transcript generation, we use all the transcripts containing the above target structure from the Librispeech train and test partitions. The train set contains ∼9.6k transcripts having the title 'mister' followed by a name, whereas the test set contains 111 such transcripts.

Next, for each transcript, we use two types of utterance generation: using actual speakers, and synthetic Text-To-Speech (TTS) voices. For actual speaker voices, we use the utterances from Librispeech train and test sets. For synthetic generation, we use four voices (two male, two female) using a WaveNet TTS system [16] to generate TTS utterances for each transcript.

Thus, we categorize our evaluation into four sets:

- **Train:** Transcripts and utterances from LibriSpeech train set. Using this set can, for instance, allow a data practitioner to empirically test for data leakage before deploying a model.
- **Test:** Transcripts and utterances from LibriSpeech test set. Less restrictive since it assumes an analyst can obtain transcripts from a similar distribution as the training data, and use a disjoint set of speakers for utterance generation.[2] We defer analysis using out-of-distribution transcripts to future work.
- **Train TTS:** Transcripts from the train set, and utterances using synthetic TTS generation. Can be used in situations where an analyst has access to the train transcripts (without the sensitive names), but not necessarily an actual speaker.
- **Test TTS:** Transcripts from the test partition, and utterances using synthetic TTS generation. The least restrictive setting where an analyst only has access to some target transcripts similar to those used for training.

For noise addition, we consider six types of noise sources: 'silence', 'car' cabin noise, 'cafe' chatter with background noise, 'music', 'kitchen' background noise, and 'podcast'.

Procedure: Our extraction method consists of passing a *noise-masked* utterance (from an actual speaker, or generated via TTS) to the trained model for inference. Every utterance contains at least one occurrence of the title 'mister'. For each utterance, we use a non-streaming Transformer-Transducer (T-T) model with RNN-T Viterbi forced alignment [17] to calculate the time-alignment of words, and we replace the audio for the word appearing after 'mister' with pure noise. For each masked name, we also mask an additional 100 ms on both sides of it as a margin to avoid any residual information. For instance, in the utterance "mister soames was somehow ...", the name 'soames' is replaced with pure noise (see Figure 1 for an illustration). Though the duration of the noise we add here uses the duration of the masked word in the utterance, our method is also applicable to arbitrary noise lengths. We provide some results for some fixed noise duration (set without any knowledge about the duration of the masked word audio) in Section 4.

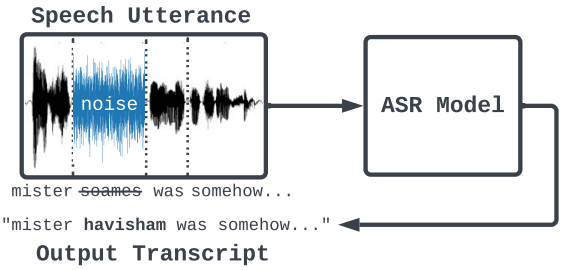

Figure 1: *An illustration of Noise Masking for name extraction on a LibriSpeech utterance. The name ('soames') is replaced with pure noise in the input utterance, and fed to a model for inference. In its output transcription, the model substitutes noise with another name ('havisham') from the train set.*

Performance Metrics: Our goal is to measure how often the model replaces the missing (masked) names with some name from the train set. To evaluate the success of our extraction method, we consider the following metrics:

- **Average true name accuracy:** This metric measures how often the model outputs the 'true' name (i.e., the name in the original utterance) from the noise-masked utterance. This corresponds to our highest level of leakage, and the model might have *memorized* the true name.
- **Average any name success rate:** A slightly milder case of the above is when the model outputs *some* name (including the true name) from the train set. We measure the average success rate of any name. To construct a set of such names, we first tabulate the words that appear after various title words in the training dataset. Next, we filter out the instances containing common words, pronouns, etc., for example, '... mister and ...', '... miss my ...', etc. We consider the remaining words as the set of sensitive names.[3]

[1] Available online at: https://www.tensorflow.org/datasets/catalog/librispeech

[2] In LibriSpeech, the sets of training and test speakers are disjoint.

[3] While our manual filtering removes many false positives for

Results: In Table 1 (#1), we present the results of Noise Masking, using 'silence' as well as the other five noise types, in the four different data settings (original/TTS version of train/test set) on the baseline model. We observe a significant amount of information leakage from the baseline model in all cases, even using 'silence' which is perhaps the most basic form of noise. Noise Masking provides the highest leakage on the train utterances (e.g., 11.8% correct name accuracy and 55.2% any name success rate using silence). Remarkably, our method is able to extract significant information from TTS version of the train set, as well as both original and TTS versions of test set. In summary, the results indicate that the train transcript information and utterances from real speaker are advantageous, but not necessary for successful extractions using Noise Masking.

3. Towards Mitigations for Noise Masking

Now, we discuss different training strategies that we empirically test towards mitigating our Noise Masking method. For each proposed strategy, we use it to train a Conformer model on LibriSpeech, and show the results of Noise Masking for name extraction on the trained model in Table 1. We also report the Word Error Rate (WER) of each model on LibriSpeech test-clean/test-other partitions in Table 2. The performance of our baseline (#1) from Section 2.2 is aligned with the Conformer benchmark on Librispeech [14].

3.1. Upper bound: Name Silencing

We start with a simple mitigation strategy designed with the knowledge of our instantiation of Noise Masking (Section 2.2). A trivial way of preventing the model against such leakage is to simply remove such names from the train set before training the model. In the Name Silencing strategy, for each train sample, we remove all the names appearing after various title keywords ('mister', 'miss', 'misses', etc.) from the transcript, and we replace the corresponding audio in the utterance with silence. We achieve the latter using the non-streaming T-T model as before to find the time-alignment of words.

For evaluation, we see in Table 2 (#2) that the model trained using Name Silencing has utility comparable to the baseline. From the results on Noise Masking in Table 1 (#2), perhaps predictably we see that Noise Masking achieves no true name accuracy in any of the considered settings. We observe some success (though extremely small relative to the baseline) for the any name metric. Such leakage could be attributed to the fact that Name Silencing 'scrubs' only the names appearing after certain title keywords in the dataset. Thus, there may still exist train samples containing names that pass such filtering.

Our results show some shortcomings of such approaches: mitigation strategies that rely on filtering targeted information can require significant domain knowledge about the data, and may still be susceptible to some leakage since the filtering is limited to a specific type of information. For instance, to protect users' email addresses, we would need to apply a different filtering strategy than Name Silencing. Note that while such a technique may not be easily scalable to all types of potentially sensitive information in a given dataset, it essentially induces an upper-bound on the amount of mitigation we can hope to achieve in this case for models with comparable utility.

names, we expect a few edge cases to pass our filtering process, which we consider negligible for our analysis.

Table 1: *Performance of Noise Masking (Average true name % / Average any name %) over different noises on different splits of the LibriSpeech dataset. In each case, we show the extraction result using 'silence' separately from the result obtained from the rest of the noises ('Others') on average.*

#	Setting / Model	Noise	Train	Train TTS	Test	Test TTS
1	Baseline	Silence	11.8/55.2	1.6/32.0	0.9/49.1	1.9/35.8
		Others	7.9/52.2	2.7/46.8	2.3/46.4	2.5/47.7
2	Name Silencing	Silence	0.0/0.5	0.0/0.8	0.0/0.0	0.0/0.0
		Others	0.0/0.7	0.0/1.1	0.0/0.9	0.0/1.3
3	MTR	Silence	12.2/49.4	0.7/12.8	0.0/36.8	1.9/12.3
		Others	13.5/56.6	5.4/50.5	7.4/54.2	4.0/48.3
4	Word Dropout	Silence	0.1/1.0	0.0/1.2	0.0/1.9	0.0/0.9
		Others	11.9/56.4	5.0/53.6	6.2/47.5	4.3/45.3
5	Word Dropout + MTR	Silence	0.3/1.6	0.1/1.6	0.0/2.8	0.0/2.8
		Others	5.9/21.1	2.2/18.1	2.3/17.4	1.1/14.2

Table 2: *Utility for all model checkpoints used for evaluation*

#	Model	test-clean WER	test-other WER
1	Baseline	2.0	4.5
2	Name Silencing	2.1	4.6
3	MTR	2.0	4.4
4	Word Dropout	2.1	4.5
5	Word Dropout + MTR	2.1	4.6

3.2. Multistyle TRaining (MTR)

Multistyle TRaining (MTR) [1] is a popular framework [2, 3] in which a room simulator is used to combine clean audio with a variety of noises. We choose MTR to check if training models using utterances augmented with background noises can induce robustness against Noise Masking. For all experiments using MTR in this paper, the noises for MTR training are different from those used by our Noise Masking implementation. Following [3], we mix clean and MTR data with an 8:2 ratio.

We see from Table 2 (#3) that the model trained using MTR provides better utility than the baseline. However, from Table 1 (#3) we see that for all of the evaluated settings, Noise Masking using noises other than silence results in more prominent leakage than the baseline, especially for true name accuracy.

3.3. Word Dropout

Building on our intuition from Name Silencing, we design a simpler approach called Word Dropout where for each sample we randomly remove words from the input transcript, and replace the corresponding audio in the utterance with silence. Word Dropout is designed to be general, and can be applied to any dataset without requiring much domain knowledge about it. For the experiments in this paper using Word Dropout, we mask one word randomly from each utterance. However, the number of masked words can be treated as a hyperparameter in general.

Table 2 (#4) shows that the model trained using Word Dropout provides comparable utility as the baseline. Moreover, we can see from Table 1 (#4) that while Word Dropout provides extreme robustness against Noise Masking with 'silence', we observe no mitigation for the other noises considered.

Next, we combine Word Dropout and MTR to see if training using utterances with randomly silenced words and augmented background noises can provide mitigation against Noise Masking. We see from Table 2 (#5) that the model trained using Word Dropout + MTR provides a comparable utility as the baseline. Moreover, we see from Table 1 (#5) that our combined strat-

egy achieves significant robustness against Noise Masking in all four settings for all types of noises considered (detailed results in Section 4). In each case, the leakage via Noise Masking is significantly lower than the baseline, especially for the any name success metric. Thus, we see that while using MTR by itself can result in an increased leakage via Noise Masking, and Word Dropout alone only provides mitigation against Noise Masking using 'silence', their combination provides substantial mitigation against a variety of noises while maintaining comparable model utility as the baseline. It is important to note that unlike Name Silencing, Word Dropout can be applied without any domain knowledge about a dataset.

Note: The technique of Differentially Private Stochastic Gradient Descent (DP-SGD) [18, 19] has been often used for obtaining guaranteed bounds on the amount of leakage about the training data from the training process. However, extending the bounds to target types of training data (e.g., via a group privacy argument [20]) can result in weak guarantees. Moreover, we consider strategies using which models can provide comparable utility as the baseline without increasing the computation cost significantly. Since the privacy-utility-computation trade-offs for DP-SGD can be substantial for large models [18, 21], we do not provide a comparison using DP-SGD.

4. Detailed Analysis for Leakage

We provide a detailed analysis of our Noise Masking results above using LibriSpeech train utterances, aimed at helping practitioners check for such leakage before deploying a model. Specifically, we design two additional metrics, and show results for all the six noises used for Noise Masking on each model.

Additional Metrics: The following two metrics help further understand the information leakage for each case.

- **Number of unique names:** There are 3622 unique names in the filtered set of names we use to compute the any name success rate. We measure the total number of unique names among the names extracted from a model.
- **Number of extrapolated names:** While we only use 'mister' for our extraction results, we measure how many of the extracted names do not ever appear after 'mister' in the train set. Such leakages could have resulted from a model's extrapolation of names coming after other title keywords. Among the 3622 unique names, 1084 never appear after 'mister'.

Detailed Results on the Train Set: In Table 3, we show the results using different types of noises on the train set. We see that regardless of the noise type, there is a high chance the baseline outputs a name given a noise masked utterance as the input. Moreover, we see that the baseline is able to extrapolate for a lot of names not strictly following the target structure. For instance, using 'Car' noise, we can recover 447 out of 3622 unique names, 42 of which never even appear after 'mister' in the train set. The model trained with Word Dropout + MTR is more robust than the baseline in terms of all the considered metrics across all the noises. In Appendix A, we provide the detailed results for all the models considered in this paper, on all the four evaluation settings.

Results with Fixed Noise Duration: In Table 4, we show the results of Noise Masking using train utterances and a fixed duration (set with no knowledge of the masked word duration) of 'silence' noise on the baseline model.[4] For reference, we provide the results with noise using the 'original' masked word duration. We see that while using the masked word duration provides a boost to the true/any name metrics, a fixed noise duration results in essentially the same amount of unique/extrapolated names being extracted across the range of durations used.

[4]We also conduct experiments for the other five noises on the baseline, and the results show similar trends as shown in Table 4. We present the rest of the results in Appendix A.

Table 3: *Detailed results of our Noise Masking implementation using train utterances and different types of noise.*

Model	Noise	True	Any	Unique	Extrapolated
Baseline	Silence	11.8%	55.2%	457	41
	Car	7.3%	47.7%	447	42
	Cafe	7.9%	50.4%	478	38
	Music	9.0%	62.1%	539	51
	Kitchen	7.7%	47.2%	463	50
	Podcast	7.5%	53.9%	595	73
Word Dropout + MTR	Silence	0.3%	1.6%	101	5
	Car	6.1%	19.5%	358	22
	Cafe	5.9%	19.2%	359	24
	Music	5.9%	20.6%	405	24
	Kitchen	6.1%	19.2%	353	22
	Podcast	5.6%	26.9%	558	68

Table 4: *Effect of noise duration on the extraction results using 'silence' and train utterances on the baseline model. 'Original' corresponds to the setting where the duration of noise added uses the duration of the masked word in the utterance.*

Duration	True	Any	Unique	Extrapolated
Original	11.8%	55.2%	457	41
100 ms	6.3%	30.1%	421	40
200 ms	6.3%	30.7%	453	42
500 ms	6.2%	29.4%	430	35
800 ms	6.0%	29.5%	432	43
1000 ms	6.5%	30.3%	433	35
1500 ms	6.3%	29.9%	433	44

5. Conclusion

In this work, we designed Noise Masking, the first method to our knowledge for demonstrating leakage of training data from trained ASR models. We conducted experiments in four settings to show the success of our method for extracting names from LibriSpeech used for training a SOTA Conformer model. Lastly, we designed Word Dropout, a data augmentation method that, when combined with MTR, provides comparable utility as the baseline along with significantly mitigating extraction via Noise Masking in all evaluated settings.

It is important to note that the instantiation of our Noise Masking method used in this paper is intended to illustrate a specific type of information leakage from a trained model. Devising noises that aid extraction better can only improve the performance of our method. Moreover, there can be a variety of target knowledge with which Noise Masking can be used for other sensitive extractions, and it can be practically infeasible to design mitigation methods customized to each type of extraction (e.g., like we designed the Name Silencing strategy in Section 3.1). While we designed Word Dropout with general applicability in mind, approaches for training ASR models that can provide better protection against leakages of structured training data while achieving utility comparable to SOTA models is an interesting direction that we leave for future work.

6. References

[1] C. Kim, A. Misra, K. K. Chin, T. Hughes, A. Narayanan, T. N. Sainath, and M. Bacchiani, "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, F. Lacerda, Ed. ISCA, 2017, pp. 379–383. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1510.html

[2] A. Narayanan, A. Misra, K. C. Sim, G. Pundak, A. Tripathi, M. Elfeky, P. Haghani, T. Strohman, and M. Bacchiani, "Toward domain-invariant speech recognition via large scale training," in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*. IEEE, 2018, pp. 441–447. [Online]. Available: https://doi.org/10.1109/SLT.2018.8639610

[3] D. S. Park, Y. Zhang, C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "Specaugment on large scale datasets," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 2020, pp. 6879–6883. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9053205

[4] T. Dang, O. Thakkar, S. Ramaswamy, R. Mathews, P. Chin, and F. Beaufays, "Revealing and protecting labels in distributed training," *CoRR*, vol. abs/2111.00556, 2021. [Online]. Available: https://arxiv.org/abs/2111.00556

[5] T. Dang, O. Thakkar S. Ramaswamy, R. Mathews, P. Chin, and F. Beaufays, "A method to reveal speaker identity in distributed ASR training, and how to counter it," *CoRR*, vol. abs/2104.07815, 2021. [Online]. Available: https://arxiv.org/abs/2104.07815

[6] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, N. Heninger and P. Traynor, Eds. USENIX Association, 2019, pp. 267–284. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/carlini

[7] C. Song and V. Shmatikov, "Auditing data provenance in text-generation models," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds. ACM, 2019, pp. 196–206. [Online]. Available: https://doi.org/10.1145/3292500.3330885

[8] O. Thakkar, S. Ramaswamy, R. Mathews, and F. Beaufays, "Understanding unintended memorization in federated learning," *CoRR*, vol. abs/2006.07490, 2020. [Online]. Available: https://arxiv.org/abs/2006.07490

[9] S. Ramaswamy, O. Thakkar, R. Mathews, G. Andrew, H. B. McMahan, and F. Beaufays, "Training production language models without memorizing user data," *CoRR*, vol. abs/2009.10031, 2020. [Online]. Available: https://arxiv.org/abs/2009.10031

[10] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel, "Extracting training data from large language models," in *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, M. Bailey and R. Greenstadt, Eds. USENIX Association, 2021, pp. 2633–2650. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

[11] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang, "Quantifying memorization across neural language models," *CoRR*, vol. abs/2202.07646, 2022. [Online]. Available: https://arxiv.org/abs/2202.07646

[12] V. Feldman and C. Zhang, "What neural networks memorize and why: Discovering the long tail via influence estimation," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1e14bfe2714193e7af5abc64ecbd6b46-Abstract.html

[13] B. Balle, G. Cherubin, and J. Hayes, "Reconstructing training data with informed adversaries," *CoRR*, vol. abs/2201.04845, 2022. [Online]. Available: https://arxiv.org/abs/2201.04845

[14] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Proc. Interspeech 2020*, pp. 5036–5040, 2020.

[15] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, 2015, pp. 5206–5210. [Online]. Available: https://doi.org/10.1109/ICASSP.2015.7178964

[16] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel wavenet: Fast high-fidelity speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 3915–3923. [Online]. Available: http://proceedings.mlr.press/v80/oord18a.html

[17] J. Kim, H. Lu, A. Tripathi, Q. Zhang, and H. Sak, "Reducing streaming ASR model delay with self alignment," in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, and P. Motlícek, Eds. ISCA, 2021, pp. 3440–3444. [Online]. Available: https://doi.org/10.21437/Interspeech.2021-322

[18] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proc. of the 2014 IEEE 55th Annual Symp. on Foundations of Computer Science (FOCS)*, 2014, pp. 464–473.

[19] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS'16)*, 2016, pp. 308–318.

[20] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[21] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu, "Practical and private (deep) learning without sampling or shuffling," in *ICML*, 2021.

A. Additional Experimental Results

In Tables 5-8, we provide the detailed results for each setting (train, train TTS, test, and test TTS utterances) and using each type of noise for Noise Masking on all the models evaluated in this paper. Specifically, we demonstrate the results for the baseline, Name Silencing, MTR, Word Dropout, and Word Dropout + MTR models.

Table 5: *Detailed results of our Noise Masking implementation using train utterances and different types of noise.*

Model	Noise	True	Any	Unique	Extrapolated
Baseline	Silence	11.8%	55.2%	457	41
	Car	7.3%	47.7%	447	42
	Cafe	7.9%	50.4%	478	38
	Music	9.0%	62.1%	539	51
	Kitchen	7.7%	47.2%	463	50
	Podcast	7.5%	53.9%	595	73
Name Silencing	Silence	0.0%	0.5%	33	4
	Car	0.0%	0.7%	56	5
	Cafe	0.0%	0.5%	40	5
	Music	0.0%	0.7%	58	6
	Kitchen	0.0%	0.6%	55	5
	Podcast	0.0%	1.2%	75	14
MTR	Silence	12.2%	49.4%	422	35
	Car	12.2%	48.9%	503	37
	Cafe	12.7%	49.7%	499	36
	Music	16.1%	73.5%	588	48
	Kitchen	12.5%	49.7%	510	46
	Podcast	14.0%	61.2%	646	77
Word Dropout	Silence	0.1%	1.0%	70	5
	Car	10.5%	48.3%	456	34
	Cafe	10.9%	48.6%	464	33
	Music	17.0%	84.6%	584	42
	Kitchen	10.5%	46.6%	451	34
	Podcast	10.8%	54.1%	610	69
Word Dropout + MTR	Silence	0.3%	1.6%	101	5
	Car	6.1%	19.5%	358	22
	Cafe	5.9%	19.2%	359	24
	Music	5.9%	20.6%	405	24
	Kitchen	6.1%	19.2%	353	22
	Podcast	5.6%	26.9%	558	68

Table 6: *Detailed results of our Noise Masking implementation using train TTS utterances and different types of noise.*

Model	Noise	True	Any	Unique	Extrapolated
Baseline	Silence	1.6%	32.0%	284	32
	Car	2.5%	45.5%	450	55
	Cafe	2.6%	45.0%	432	49
	Music	3.0%	49.4%	514	67
	Kitchen	2.7%	45.1%	439	47
	Podcast	2.7%	48.9%	561	72
Name Silencing	Silence	0.0%	0.8%	55	9
	Car	0.0%	1.0%	124	23
	Cafe	0.0%	1.0%	113	21
	Music	0.0%	1.2%	138	26
	Kitchen	0.0%	1.0%	110	23
	Podcast	0.0%	1.2%	144	25
MTR	Silence	0.7%	12.8%	226	33
	Car	5.0%	46.2%	540	56
	Cafe	5.2%	46.4%	560	56
	Music	6.0%	56.0%	615	61
	Kitchen	5.1%	46.8%	552	57
	Podcast	5.7%	57.3%	720	90
Word Dropout	Silence	0.0%	1.2%	85	15
	Car	4.2%	45.2%	465	44
	Cafe	4.6%	45.5%	497	47
	Music	7.2%	80.7%	588	63
	Kitchen	4.3%	44.9%	480	53
	Podcast	4.7%	51.5%	623	81
Word Dropout + MTR	Silence	0.1%	1.6%	115	22
	Car	2.3%	18.4%	446	47
	Cafe	2.3%	17.4%	452	50
	Music	2.0%	16.7%	468	50
	Kitchen	2.3%	17.6%	423	39
	Podcast	1.9%	20.4%	636	89

Table 7: *Detailed results of our Noise Masking implementation using test utterances and different types of noise.*

Model	Noise	True	Any	Unique	Extrapolated
Baseline	Silence	0.9%	49.1%	26	1
	Car	3.8%	44.3%	28	0
	Cafe	0.0%	41.5%	26	1
	Music	3.8%	54.7%	36	0
	Kitchen	1.9%	41.5%	29	1
	Podcast	1.9%	50.0%	34	2
Name Silencing	Silence	0.0%	0.0%	0	0
	Car	0.0%	1.9%	2	0
	Cafe	0.0%	0.9%	1	0
	Music	0.0%	1.9%	2	0
	Kitchen	0.0%	0.0%	0	0
	Podcast	0.0%	0.0%	0	0
MTR	Silence	0.0%	36.8%	14	0
	Car	7.5%	51.9%	36	0
	Cafe	6.6%	44.3%	31	0
	Music	6.6%	62.3%	44	1
	Kitchen	7.5%	57.5%	36	0
	Podcast	8.5%	54.7%	38	2
Word Dropout	Silence	0.0%	1.9%	2	0
	Car	6.6%	42.5%	34	2
	Cafe	7.5%	40.6%	25	1
	Music	7.5%	71.7%	39	1
	Kitchen	4.7%	39.6%	32	1
	Podcast	4.7%	43.4%	38	2
Word Dropout + MTR	Silence	0.0%	2.8%	3	1
	Car	2.8%	11.3%	10	1
	Cafe	2.8%	18.9%	14	0
	Music	0.9%	15.1%	16	1
	Kitchen	2.8%	17.9%	16	1
	Podcast	1.9%	23.6%	23	2

Table 8: *Detailed results of our Noise Masking implementation using test TTS utterances and different types of noise.*

Model	Noise	True	Any	Unique	Extrapolated
Baseline	Silence	1.9%	35.8%	17	1
	Car	0.9%	56.6%	28	1
	Cafe	2.8%	46.2%	22	2
	Music	1.9%	44.3%	29	0
	Kitchen	6.6%	50.0%	25	0
	Podcast	0.0%	41.5%	25	0
Name Silencing	Silence	0.0%	0.0%	0	0
	Car	0.0%	0.9%	1	1
	Cafe	0.0%	1.9%	2	0
	Music	0.0%	0.0%	0	0
	Kitchen	0.0%	2.8%	3	0
	Podcast	0.0%	0.9%	1	0
MTR	Silence	1.9%	12.3%	8	0
	Car	0.9%	42.5%	32	0
	Cafe	3.8%	44.3%	33	0
	Music	6.6%	52.8%	34	0
	Kitchen	3.8%	45.3%	36	0
	Podcast	4.7%	56.6%	40	1
Word Dropout	Silence	0.0%	0.9%	1	0
	Car	3.8%	39.6%	26	0
	Cafe	4.7%	37.7%	25	0
	Music	7.5%	68.9%	42	1
	Kitchen	2.8%	34.0%	26	0
	Podcast	2.8%	46.2%	33	1
Word Dropout + MTR	Silence	0.0%	2.8%	3	2
	Car	0.9%	14.2%	12	0
	Cafe	2.8%	19.8%	16	1
	Music	0.9%	10.4%	10	1
	Kitchen	0.0%	14.2%	13	0
	Podcast	0.9%	12.3%	13	1

In Table 9, we also provide the results of Noise Masking on the baseline model using a fixed-length noise of 100 ms duration on train utterances. For reference, we also provide results for noise that is of the same duration as the masked word in the utterance. We observe that although the extraction performance degrades without using the original word duration, we can still recover a significant amount of unique and extrapolated names using any type of noise.

Table 9: *Extraction results using different noises and train utterances on the baseline model. 'Original' corresponds to the setting where the duration of noise added uses the duration of the masked word in the utterance, and 'Fixed' denotes the results using a fixed-length noise of 100 ms duration.*

Noise	Duration	True	Any	Unique	Extrapolated
Silence	Original	11.8%	55.2%	457	41
	Fixed	6.3%	30.1%	421	40
Car	Original	7.3%	47.7%	447	42
	Fixed	0.9%	6.6%	278	38
Cafe	Original	7.9%	50.4%	478	38
	Fixed	1.2%	8.4%	297	30
Music	Original	9.0%	62.1%	539	51
	Fixed	5.3%	40.2%	541	56
Kitchen	Original	7.7%	47.2%	463	50
	Fixed	0.4%	4.3%	253	30
Podcast	Original	7.5%	53.9%	595	73
	Fixed	0.9%	12.4%	462	57