

MITIGATING UNINTENDED MEMORIZATION IN LANGUAGE MODELS VIA ALTERNATING TEACHING

Zhe Liu, Xuedong Zhang, Fuchun Peng

Meta AI, Menlo Park, CA, USA

ABSTRACT

Recent research has shown that language models have a tendency to memorize rare or unique sequences in the training corpora which can thus leak sensitive attributes of user data. We employ a teacher-student framework and propose a novel approach called alternating teaching to mitigate unintended memorization in sequential modeling. In our method, multiple teachers are trained on disjoint training sets whose privacy one wishes to protect, and teachers' predictions supervise the training of a student model in an alternating manner at each time step. Experiments on LibriSpeech datasets show that the proposed method achieves superior privacy-preserving results than other counterparts. In comparison with no prevention for unintended memorization, the overall utility loss is small when training records are sufficient.

Index Terms— Language modeling, unintended memorization, knowledge distillation, automatic speech recognition

1. INTRODUCTION

Neural language models (LMs) play important roles in many natural language processing tasks including next word prediction, machine translation, and automatic speech recognition (ASR) [1, 2, 3, 4, 5]. They typically outperform traditional n -gram LMs with better capability of modeling long-range dependency.

State-of-the-art LMs typically involve training over large and diverse corpora which might contain sensitive user information, such as addresses and credit card numbers. Recent research has showed that such sensitive information in training datasets can be detected and extracted in unexpected ways [6, 7, 8, 9, 10]. Particularly, LMs are prone to *unintentionally memorize* rare or unique sequences of data, and when being prompted appropriately, they will be able to emit the memorized text verbatim [11]. This is undesirable because such memorization violates privacy by exposing user information. Therefore, providing privacy guarantees to LM training has become a critical problem and it calls for advanced mitigation techniques for unintended memorization in LMs.

In this paper, we employ a teacher-student framework and propose a novel method called *alternating teaching* to mitigate the issue of unintended memorization in sequential modeling. In our approach, multiple teachers are trained on disjoint training data (e.g. data from different users) whose privacy one wishes to protect, and teachers' predictions are utilized as soft labels to supervise the training of a student model. Unlike teacher ensemble and aggregation methods, at each word-level time step of student model training, we only choose one teacher to provide supervision. That is, teachers are selected in an alternating manner through randomization or permutation. Finally, only the student model is published while all teachers are kept private.

Intuitively, in most scenarios any piece of sensitive information is only contained in the training text of one specific user, and is thus exposed to one teacher model. Then alternating teacher selection at each time step breaks the semantic and linguistic connections between consecutive words in any private sequences, but is still able to learn from common and non-sensitive word combinations that are exposed to all teacher models. Thus, this technique can reduce the level of memorization without appreciable loss of overall utility in the model.

Our approach is inspired by text generation models which generate words in a sequence step-by-step and left-to-right. One promising method for addressing memorization via text generation is to use one teacher to generate a word at each time step and the word with its historical contexts acts as the next input for another teacher. However, this strategy is costly in computation especially when hundreds of millions of sentences need to be generated. Under the knowledge distillation framework, the proposed alternating teaching approach is more efficient and scalable.

We make the following contributions: (1) introducing the new alternating teaching based teacher-student framework for effective mitigation of unintended memorization in LMs; (2) studying the effect of various knowledge distillation mechanisms on alleviating memorization, which includes the use of public training corpora and adding random noises to teachers' output distributions; and (3) providing empirical results and analyses on comparing the utility and privacy protection of various teacher-student learning based approaches.

The rest of the paper is organized as follows. We review related work in Section 2. Section 3 describes the details of our proposed alternating teaching method. Next, Section 4 shows the experiments and results for LM and ASR tasks on the *LibriSpeech* data [12]. We conclude in Section 5.

2. RELATED WORK

Privacy protection is becoming crucial in machine learning research. One direction in this area is *private aggregation of teacher ensembles* (PATE) [13, 14], which transfers to a student model the knowledge of an ensemble of teacher models, with strong privacy guaranteed by noisy aggregation and vote counts of teachers' answers. More recently, authors in [15] adapts PATE to text generation tasks while satisfying *differential privacy* (DP) [16, 17]. Our work differs from PATE and its variants in alternating teacher selection instead of aggregation mechanisms. Moreover, we focus on the empirical measurement of unintended memorization rather than DP-based privacy analysis. Another line of research on privacy-preserving methods is federated learning (FL) [18, 19, 20, 21]. Apart from these works, authors in [22] explores how memorization relates to generalization in learning.

3. METHODOLOGY

3.1. The Alternating Teaching Framework

To help mitigate unintentionally memorization in LMs, the proposed alternating teaching framework transfers knowledge from alternating teacher models trained on partitions of the data to a student model. Our method consists of three key parts: (1) multiple teacher models, (2) teacher selection mechanism, and (3) a student model.

3.1.1. Teacher Models

Each teacher model is an LM trained independently on a subset of the data. The data is partitioned into disjoint subsets by users to ensure no pair of teachers will have trained on data from the same user. In other words, all records from any user is only included in the training corpus of one specific teacher model.

More formally, let $\mathcal{D}_k = \{\mathcal{D}_k^{\text{pri}} \cup \mathcal{D}_k^{\text{pub}}\}$ be the training corpus from the k th user, where $\mathcal{D}_k^{\text{pri}}$ is a set of records with sensitive information and $\mathcal{D}_k^{\text{pub}}$ is a corpus without sensitive information. Then let $\mathcal{D} = \cup_{k=1}^K \mathcal{D}_k$ be the entire training set over all users' data. Without the loss of generality, assuming the training set is partitioned into M disjoint subsets by users, denote $\mathcal{B}_m = \cup_{k=(m-1)d+1}^{md} \mathcal{D}_k$ for each $m = 1, \dots, M$ and $d = \lfloor K/M \rfloor$. Thus we have $\mathcal{D} = \cup_{m=1}^M \mathcal{B}_m$ and $\mathcal{B}_{m'} \cap \mathcal{B}_{m''} = \emptyset$ for any $m' \neq m''$.

The m th teacher model, denoted as $f_m(\theta_m)$ with θ_m being the weights, is trained using text set of \mathcal{B}_m . Cross entropy (CE) loss is usually used for LM training. Given any training example with T words, $(w_1, w_2, \dots, w_T) \in \mathcal{B}_m$, the following shows this function at step t

$$\mathcal{L}_t^{\text{CE}}(\theta_m) = - \sum_{w \in V} \mathbf{1}\{w = w_t\} \cdot \log p_{\theta_m}(w|w_{1:t-1}) \quad (1)$$

where V is the vocabulary set and $f_m(\theta_m)$ predicts a word w with a probability $p_{\theta_m}(w|w_{1:t-1})$ at step t .

Thus we obtain a set of teacher models $\{f_m(\theta_m)\}_{m=1}^M$. Note that any teacher is not privacy-preserving and susceptible to unintended memorization since it trains on combined sets of sensitive corpus and non-sensitive corpus.

3.1.2. Alternative Teaching Mechanism

Once all teacher LMs are trained, given any $(w_1, w_2, \dots, w_T) \in \mathcal{D}$, each teacher $\{f_m(\theta_m)\}_{m=1}^M$ conducts the inference on it and outputs probability distribution $p_{\theta_m}(\cdot|w_{1:t-1})$ over all words in the vocabulary at step t . Here, we discuss how these predictions can be combined to provide supervisions.

One promising approach is the aggregation mechanism where at each step, the predicted probabilities from teachers are averaged on each word in the vocabulary. In particular, the following ensemble teacher output is used to supervise a student model at step t

$$g^{\text{agg}}(\cdot|w_{1:t-1}) = \frac{1}{M} \sum_{m=1}^M p_{\theta_m}(\cdot|w_{1:t-1}) \quad (2)$$

Then a student model is trained on this aggregated output of the M teachers, such that it learns to accurately mimic the ensemble. Intuitively, this aggregation strategy ensures no single teacher and thus no single user's dataset dictates the student's training. This will help alleviate any unintended memorization. However, one disadvantage of this approach is that when the presence of some private sequence in one specific teacher is very strong and even dominating, simply

taking the average over the probabilities of all teachers might not be adequate to provide a full coverage and still reveal such sensitive information to the student.

In the newly proposed alternative teaching mechanism, at each time step we only leverage the prediction output from one teacher rather than using all teachers' aggregation, and alternate the choices of teachers over different steps. This can be performed through randomization or fixed permutation.

In the randomization based teacher selection, for each step t , we randomly generate $r^{\text{random}}(t) \in \{1, 2, \dots, M\}$ and the corresponding teacher model is chosen as the supervisor. It aims to disconnect consecutive words in private sequences but generally has no issues in learning common and non-sensitive sequences that are present in majority of teachers.

The randomized teacher selection happens in each step at every batch during training. A more restricted teacher selection strategy is through permutation but kept the chosen order fixed over the entire training process. Specifically, let $\pi(M)$ be a random permutation of the sequence $\{1, 2, \dots, M\}$, then the teacher index at step t , denoted as $r^{\text{perm}}(t)$, is chosen as the j th element of $\pi(M)$, where $j = (t \bmod M)$ if the corresponding remainder is non-zero; otherwise $j = M$. The assignment of $r^{\text{perm}}(t)$ stays intact across different batches and epochs.

In either case, let $r(t)$ be the selected teacher index at step t , then we write

$$g^{\text{alt}}(\cdot|w_{1:t-1}) = p_{\theta_{r(t)}}(\cdot|w_{1:t-1}) \quad (3)$$

as the predicted distribution which is used to supervise the student model at step t .

3.1.3. Student Model

Since any training corpora are naturally labeled for LM task, the student model, denoted by $f(\theta)$, is supervised by both the labels from its training set and combined teachers' outputs. Then for any sequence (w_1, w_2, \dots, w_T) , the following computes the loss function consisting of two parts

$$\mathcal{L}(\theta) = \sum_{t=1}^T \left((1 - \lambda) \cdot \mathcal{L}_t^{\text{CE}}(\theta) + \lambda \cdot \mathcal{L}_t^{\text{KL}}(\theta) \right) \quad (4)$$

$$\mathcal{L}_t^{\text{CE}}(\theta) := - \sum_{w \in V} \mathbf{1}\{w = w_t\} \cdot \log p_{\theta}(w|w_{1:t-1}) \quad (5)$$

$$\mathcal{L}_t^{\text{KL}}(\theta) := D_{\text{KL}}(g^{\text{alt}}(\cdot|w_{1:t-1}) || p_{\theta}(\cdot|w_{1:t-1})) \quad (6)$$

where $D_{\text{KL}}(P || Q)$ represents the Kullback–Leibler divergence between distributions P and Q , and λ is a hyperparameter which balances the two parts of $\mathcal{L}_t^{\text{CE}}(\theta)$ and $\mathcal{L}_t^{\text{KL}}(\theta)$.

The student model can be trained on any auxiliary, non-sensitive corpora, including publicly available collections of text data. However, when such dataset is not available or the student model suffers from utility loss due to distillation, the original set \mathcal{D} can still be used to train the student model. In that case, the hyperparameter λ shall be set as 0 since \mathcal{D} contains private information and we do not want it is directly exposed to the student model. In that case, the student model fully learns from combined teachers' outputs.

3.2. The Gaussian Noise Mechanism

Building upon the alternating teaching framework described above, random noises can be added to the outputs from teacher models so that they can further mask the presence of private sequences and thus

make sensitive information less susceptible to leakage. We apply the Gaussian mechanism which adds noise independently sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$ to each coordinate of the predicted probabilities from teachers, after which the re-normalization over vocabulary space is needed. The hyperparameter σ governs the strength of privacy protection. Specifically, the teacher supervision part $\mathcal{L}_t^{\text{KL}}(\theta)$ in (6) can be adjusted as

$$D_{\text{KL}}(s(g^{\text{alt}}(\cdot|w_{1:t-1}) + \mathcal{N}(0, \sigma^2)) || p_{\theta}(\cdot|w_{1:t-1})) \quad (7)$$

where $s(\cdot)$ is a normalization function over the vocabulary such that all probabilities with added noises are truncated to non-negative and their sum equals to 1 after normalization.

4. EXPERIMENTS

4.1. Datasets

Our experiments use the LibriSpeech data [12] and its extended text-only corpus [23]:

- LibriSpeech ASR corpus and text transcripts. It is a corpus of around 1000 hours of 16kHz read English audiobooks. The dataset consists of train, validation, and test splits, which contain 281K, 6K, and 6K utterances from approximately 2400 speakers, respectively;
- LibriSpeech extended text-only corpus. It is from 14500 public domain books which contains around 40M sentences. The dataset is only for LM training purpose.

In the next subsection, we will describe how these datasets are augmented with “private sequences” such that we can measure the performance of mitigating unintended memorization over different methods.

In some portion of our experiments, we also utilize the training dataset from Wikitext-103 [24]. This is treated as an auxiliary and non-sensitive public corpus, on which the student LM is trained.

4.2. Canaries

To measure the level of unintended memorization in LMs, we build on the “secret sharer” framework introduced in [8]. Specifically, random textual sequences, called *canaries*, are inserted into a training corpus, and a model trained on this corpus is then analyzed to measure the frequencies of having these canaries memorized. Here, the canaries aim to mimic sensitive data.

The procedure of inserting canaries into LibriSpeech datasets is described as follows:

- (1) First, each record is assigned a user ID. For the LibriSpeech ASR corpus, user ID of any utterance is just the speaker id. For the LibriSpeech text-only corpus, we randomly shuffle all the records, and create synthetic users where each user owns 100 records, assigned sequentially from the shuffled set;
- (2) Next, we randomly pick 100 users for each of the two LibriSpeech datasets. For each user, a random 5-word canary is generated which simulates the “private sequence” from that specific user. No canary is shared by different users. Note that each word in any generated canary is among the vocabulary set of LMs;
- (3) For each generated canary, we insert it into the LibriSpeech training corpora at a certain frequency (i.e. number of times it is repeated). Specifically, the 100 canaries (from 100 users) are evenly partitioned into 4 groups with canaries’ repeating

frequencies being 5%, 10%, 50%, and 100%, respectively. For each canary, let n_u be the number of training records in the corresponding user and p_u be the repeating frequency based on the group it belongs to, then $p_u \cdot n_u$ is the number of records that the canary is inserted into the training corpora of the corresponding user.

The procedure illustrated above is intended to simulate real-world scenarios where any occurrences of user-specific unique or rare out-of-distribution canaries are typically limited to a very small fraction of users, but these users can exhibit either low or high usage of those canaries [21].

Given a prefix of a canary, we use the following two techniques to evaluate the mitigation of unintended memorization for any LM:

- Beam Search (BS). We leverage a greedy beam search to see if the canary is included in the top 100 most-likely 5-word continuations from the 1-word prefix of the canary;
- Random Sampling (RS). We say any canary is unintentionally memorized by a LM if the canary has the least perplexity among 1000 random suffixes, given the 2-word prefix of the canary.

In our experiments, we report the frequencies of times that the 100 generated canaries are detected by BS or RS in any LM.

4.3. Setups

With the generated canaries inserted into the two LibriSpeech training sets, LMs are trained on the text corpora with their perplexity (PPL) measured on the test split of LibriSpeech ASR corpus. The level of unintended memorization is evaluated using the BS and RS techniques described above.

The LM in our experiments is LSTM based with embeddings dimension 300, and 2 layers of 1500 hidden units. The word vocabulary set is around 10K. We use Adam optimizer and early stopping based on the validation set of LibriSpeech ASR corpus.

In our experiments, we consider the following approaches in the comparison of utility and unintended memorization mitigation:

- The **Baseline** LM is directly trained on LibriSpeech data (either ASR corpus or text-only corpus) with canaries;
- **Baseline (1T)** refers to the student LM with knowledge distilled from a single teacher model. Here, both teacher and student models are trained on LibriSpeech data with canaries;
- **Agg** represents the student model supervised by aggregation based teacher ensembles. We use the notations of **Agg (2T)** and **Agg (5T)** to denote there are 2 and 5 teachers for knowledge distillation, respectively. Again, all teachers and the student are trained on LibriSpeech corpora with canaries;
- **Alt-Random** and **Alt-Perm** are our proposed approaches of alternating teaching with randomization and fixed permutation strategies for selecting teachers per step, respectively.

In each of knowledge distillation based methods, the parameter of λ is set to 0. In other words, the student model is only supervised by teachers’ outputs.

On the test split of LibriSpeech ASR corpus, we also evaluate the ASR performance, in terms of word-error-rate (WER), with the LMs being used as second-pass rescorers on the generated 20-best hypotheses. The ASR model is a RNN-T model with the Emformer encoder [25], LSTM predictor, and a joiner. It has around 80 million parameters and is trained from scratch using the train split of LibriSpeech ASR corpus. Note that we only measure the impact on

WER when LMs are trained on LibriSpeech text-only corpus since they will be more effective in rescoring. This is because the ASR model does not include such text-only corpus in its model training.

4.4. Results

We first measure the performance of (student) LMs trained on the LibriSpeech ASR corpus with canaries. All teacher models are also trained on it. Table 1 shows the PPL results on the test split as well as the percentage of canaries being uncovered by BS and RS techniques. Here, the 100 canaries are partitioned into two categories with low repeating frequencies and high repeating frequencies, and we report their results separately. From the results

- In the *Baseline* methods, canaries are substantially memorized by the LMs. Memorization is detectable even for canaries that appear only a few times in the training corpus;
- It is expected to see that having more teacher models leads to stronger mitigation of memorized canaries in all methods. Particularly, the proposed *Alt* performs better than *Agg*, and *Alt-Perm* has the fewest canaries detected;
- Degradation on PPL are observed when student models are supervised by multiple teachers. This is expected since LibriSpeech ASR corpus only contains less than 300K training records, thus partitioning them into multiple disjoint sets for training teacher models will cause accuracy loss due to insufficiency of training data.

Table 1. Results for LMs trained on LibriSpeech ASR corpus. PPL and percentages of canaries detected by BS and RS are reported. Low: group of canaries with low (5% or 10%) repeating frequencies; High: canaries with high (50% or 100%) repeating frequencies.

Method	Utility		BS		RS	
	PPL	WER	Low	High	Low	High
Baseline	76.4	6.92	95%	100%	100%	100%
Baseline (1T)	76.0	6.61	92%	100%	100%	100%
Agg (2T)	83.8	6.61	66%	100%	100%	100%
Alt-Random (2T)	84.4	6.62	62%	100%	100%	100%
Alt-Perm (2T)	86.6	6.62	0%	0%	100%	100%
Agg (5T)	100.5	6.64	0%	32%	48%	100%
Alt-Random (5T)	101.9	6.64	0%	4%	2%	98%
Alt-Perm (5T)	107.0	6.64	0%	0%	0%	18%

Table 2 displays the results where all the teachers are trained using LibriSpeech ASR corpus with canaries, but the student LMs are trained on Wikitext-103 data with teachers’ supervision. The observations on the comparison of different methods are similar to the ones in Table 1, where *Alt-Perm* has the smallest number of canaries being detected. Although the PPL results do not change much from the ones in Table 1, we can see that training the student models using an auxiliary and non-sensitive dataset here achieves reduced memorization.

Next, we measure the performance of (student) LMs trained on LibriSpeech text-only corpus with inserted canaries. The test split of LibriSpeech ASR corpus is used for evaluating the PPL and WER results. Seen from Table 3, the utility gaps are relatively small over methods with different numbers of teachers, which can be explained by the large training corpus of 40M sentences. Particularly, WERs only differ in less than 1% comparing *Agg* or *Alt* with *Baseline*. Thus, all these approaches tend to match the baseline utility while being empirically less prone to memorization. Again, we notice that

Table 2. Results for student LMs trained on Wikitext-103; Teacher LMs are still trained on LibriSpeech ASR corpus.

Method	Utility		BS		RS	
	PPL	WER	Low	High	Low	High
Baseline	76.4	6.92	95%	100%	100%	100%
Baseline (Wiki, 1T)	76.5	6.61	92%	100%	100%	100%
Agg (Wiki, 2T)	84.6	6.61	14%	48%	100%	100%
Alt-Random (Wiki, 2T)	84.9	6.62	4%	30%	100%	100%
Alt-Perm (Wiki, 2T)	87.9	6.62	0%	0%	94%	92%
Agg (Wiki, 5T)	101.2	6.64	0%	12%	18%	98%
Alt-Random (Wiki, 5T)	101.4	6.64	0%	6%	2%	88%
Alt-Perm (Wiki, 5T)	108.5	6.64	0%	0%	0%	12%

Alt-Perm (5T) achieves the strongest mitigation of memorization comparing with others, and *Alt-Random* (5T) obtains less memorization than *Agg* (5T).

Table 3. Results for LMs trained on LibriSpeech text-only corpus.

Method	Utility		BS		RS	
	PPL	WER	Low	High	Low	High
NoLM	-	6.92	-	-	-	-
Baseline	46.9	6.59	2%	78%	38%	100%
Baseline (1T)	48.5	6.61	2%	68%	22%	98%
Agg (2T)	49.2	6.61	2%	64%	18%	94%
Alt-Random (2T)	49.4	6.62	0%	50%	12%	94%
Alt-Perm (2T)	49.4	6.62	0%	0%	14%	92%
Agg (5T)	50.5	6.64	0%	12%	4%	72%
Alt-Random (5T)	51.0	6.64	0%	0%	2%	40%
Alt-Perm (5T)	51.0	6.64	0%	0%	0%	24%

Lastly, we study the effect of adding Gaussian noises on top of the *Alt-Perm* framework. Seen from the results in Table 4, the *Alt-Perm* (5T) method with noise scale parameter $\sigma = 1e^{-4}$ has no canaries detected by BS or RS, while the WER is compromised by around 1.5% compared with *Baseline*.

Table 4. Results for LMs trained on LibriSpeech text-only corpus, with Gaussian noise mechanism being applied.

Method	Utility		BS		RS	
	PPL	WER	Low	High	Low	High
Alt-Perm (5T)	51.0	6.64	0%	0%	0%	24%
Alt-Perm (5T, $\sigma = 1e^{-5}$)	53.1	6.65	0%	0%	0%	10%
Alt-Perm (5T, $\sigma = 1e^{-4}$)	59.2	6.69	0%	0%	0%	0%

5. CONCLUSION

In this work, we propose the alternating teaching method to mitigate unintended memorization in sequential modeling. With experiments on LibriSpeech datasets, we show this approach achieves stronger mitigation than other counterparts and significantly reduces memorized sequences. Compared with the baselines without protections for memorizing private data, the overall quality of proposed method is not compromised when there exists sufficient training data.

Future work might include extending the proposed framework to user-level DP-based privacy analysis.

6. REFERENCES

- [1] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur, “Recurrent neural network based language model,” in *Proc. Interspeech*, 2010.
- [2] Xie Chen, Xunying Liu, Mark JF Gales, and Philip C Woodland, “Improving the training and evaluation efficiency of recurrent neural network language models,” in *Proc. ICASSP*, 2015.
- [3] Xunying Liu, Yongqiang Wang, Xie Chen, Mark JF Gales, and Philip C Woodland, “Efficient lattice rescoring using recurrent neural network language models,” in *Proc. ICASSP*, 2014.
- [4] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhiheng Chen, and Rohit Prabhavalkar, “An analysis of incorporating an external language model into a sequence-to-sequence model,” in *Proc. ICASSP*, 2018.
- [5] Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney, “Language modeling with deep transformers,” in *Proc. Interspeech*, 2019.
- [6] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proc. ACM SIGSAC*, 2015.
- [7] Congzheng Song and Vitaly Shmatikov, “Auditing data provenance in text-generation models,” in *Proc. ACM SIGKDD*, 2019.
- [8] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song, “The secret sharer: Evaluating and testing unintended memorization in neural networks,” in *28th USENIX Security Symposium*, 2019.
- [9] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al., “Extracting training data from large language models,” in *30th USENIX Security Symposium*, 2021.
- [10] W Ronny Huang, Steve Chien, Om Thakkar, and Rajiv Mathews, “Detecting unintended memorization in language-model-fused ASR,” in *Proc. Interspeech*, 2022.
- [11] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang, “Quantifying memorization across neural language models,” *arXiv preprint arXiv:2202.07646*, 2022.
- [12] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an ASR corpus based on public domain audio books,” in *Proc. ICASSP*, 2015.
- [13] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar, “Semi-supervised knowledge transfer for deep learning from private training data,” in *Proc. ICLR*, 2017.
- [14] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson, “Scalable private learning with PATE,” in *Proc. ICLR*, 2018.
- [15] Zhiliang Tian, Yingxiu Zhao, Ziyue Huang, Yu-Xiang Wang, Nevin Zhang, and He He, “SeqPATE: Differentially private text generation via knowledge distillation,” 2022, https://openreview.net/forum?id=5sP_PUUS78v.
- [16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith, “Calibrating noise to sensitivity in private data analysis,” in *Theory of Cryptography Conference*. Springer, 2006, pp. 265–284.
- [17] Cynthia Dwork and Aaron Roth, “The algorithmic foundations of differential privacy,” *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.
- [18] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, “Federated learning: Strategies for improving communication efficiency,” *NeurIPS Workshop on Private Multi-Party Machine Learning*, 2016.
- [19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *AIS-TATS*, 2017.
- [20] Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays, “Training production language models without memorizing user data,” *arXiv preprint arXiv:2009.10031*, 2020.
- [21] Om Dipakbhai Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Françoise Beaufays, “Understanding unintended memorization in language models under federated learning,” in *Proc. ACL*, 2020.
- [22] Vitaly Feldman, “Does learning require memorization? a short tale about a long tail,” in *Proc. ACM SIGACT*, 2020.
- [23] “LibriSpeech language models, vocabulary and G2P models,” <https://www.openslr.org/11/>.
- [24] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher, “Pointer sentinel mixture models,” *arXiv preprint arXiv:1609.07843*, 2016.
- [25] Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer, “Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition,” in *Proc. ICASSP*, 2021.