Training Data Leakage Analysis in Language Models

Huseyin A. Inan*
huseyin.inan@microsoft.com
Microsoft Research

Osman Ramadan*
osman.ramadan@microsoft.com
Microsoft Corporation

Lukas Wutschitz
lukas.wutschitz@microsoft.com
Microsoft Corporation

Daniel Jones
jonesdaniel@microsoft.com
Microsoft Corporation

Victor Rühle
virueh@microsoft.com
Microsoft Corporation

James Withers
jawithe@microsoft.com
Microsoft Corporation

Robert Sim
rsim@microsoft.com
Microsoft Research

ABSTRACT

Recent advances in neural network based language models lead to successful deployments of such models, improving user experience in various applications. It has been demonstrated that strong performance of language models comes along with the ability to memorize rare training samples, which poses serious privacy threats in case the model is trained on confidential user content. In this work, we introduce a methodology that investigates identifying the user content in the training data that could be leaked under a strong and realistic threat model. Motivated from the notion of plausible deniability, we introduce a privacy metric and illustrate how the proposed metric can be utilized to investigate the efficacy of mitigations such as differentially private model training under realistic deployment scenarios.

ACM Reference Format:
Huseyin A. Inan, Osman Ramadan, Lukas Wutschitz, Daniel Jones, Victor Rühle, James Withers, and Robert Sim. 2021. Training Data Leakage Analysis in Language Models. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Advances in language modeling have produced high-capacity models which perform very well on many language tasks. Language models are of particular interest as they are capable of generating free-form text, given a context, or even unprompted. There is a plethora of applications where language models have the opportunity to improve user experience, and many of them have been deployed in practice to do so, such as text auto-completion in emails and predictive keyboards (Fig. 1). Language models with massive capacities have been shown to achieve strong performance in other tasks as well, e.g. translation, question-answering etc. even in a zero shot setting without fine-tuning in some cases [6].

On the other hand, recent studies have demonstrated that these models can memorize training samples, which can be subsequently reconstructed using probing attacks, or even during free-form generation [7, 8]. While domain adaptation of general phrases is intended, the model should not leak or memorize any information linkable to a user in the training set, which could lead to a privacy breach according to GDPR, such as singling out of a user [4].

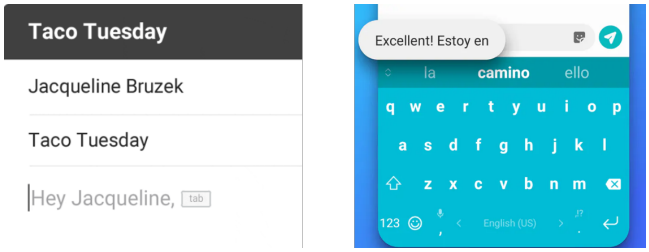

Figure 1: Two language model deployments in practice. The figure on the left (image credit: [20]) is the Smart Compose feature for Gmail [10] and the figure on the right (image credit: [25]) is the Microsoft SwiftKey Keyboard.

Among various privacy mitigation techniques, differential privacy (DP) [13] has become the gold standard notion of privacy, widely employed in the industry [3, 19, 46]. Training machine learning models with DP-SGD [1, 38] allow the participants to be protected under the theoretical guarantee that with a high probability the same machine learning model could have been obtained had they not been part of the dataset. This guarantee gives each participant the notion of **plausible deniability** and protects against GDPR's singling out [12]. Training machine learning models with DP has also achieved favorable utility-privacy trade-offs in the context of language modeling as shown in [24].

The guarantees provided by DP hold with the premise that it is implemented correctly, as previous work showed that the implementation may be error-prone from multiple angles [18, 42]. Therefore, it is of importance to have an additional privacy analysis step to ensure that the mitigation is intact, and no privacy violation may occur through interactions with a deployed private model.

*Both authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

In this work we propose a methodology for privacy investigations of a language model trained on confidential user content. Furthermore, motivated by the notion of plausible deniability, we introduce a metric that quantitatively measures how well the expected plausible deniability holds prior to the model deployment. We consider a realistic threat model under the strictest black-box assumptions about access to the model, i.e. that attackers can access only the model's top-k prediction at each token position, given an input prefix. This choice of threat model enables us to assess a model's risk for realistic deployment scenarios, assuming best practices in API hardening are employed. Another advantage of our metric is that it can assess the privacy efficacy of DP under this realistic adversary as DP-SGD analysis is based on a very powerful adversary which has access to all intermediate computations used to train a model.

1.1 Contributions

This paper makes the following contributions:

(1) We propose a methodology that investigates the user content in the training set that could be leaked by the model when prompted with the associated context.
(2) Motivated by the notion of plausible deniability offered by DP, we introduce a privacy metric that gives a measure of this notion holding over users in the training set.
(3) We consider a realistic and practical threat model to evaluate the efficacy of DP[1] under realistic deployment scenarios. We show in our experiments that relatively larger epsilon values provide reasonable plausible deniability while achieving favorable utility for the model.

2 THREAT MODEL

Our threat model is tailored for privacy considerations when a language model is trained on confidential user content, which contains linkable information that would lead to privacy violations in case they are leaked by the model [4, 41]. Such privacy considerations are in fact legitimate as language models perform next token prediction so they could be used in a generative fashion by entering a particular text prefix and asking the model to auto-complete indefinitely. Here, the danger is imminent as it is not *a priori* clear what will be leaked from the user content in the training data. Since the main objective of training language models is modeling the underlying distribution of a language, well-generalized models are not expected to memorize the user-specific information in the training data, as they are in general out-of-distribution and irrelevant to the learning task, hence unnecessary to improve the model performance. Recent results show that this is not the case [5, 7, 8, 14, 32]. When the data distribution is long-tailed (as is the natural language [30]), it has been shown that label memorization is necessary for achieving near-optimal accuracy on test data [5, 14]. Therefore, it is imperative to use privacy-preserving tools and build privacy monitoring techniques to minimize the chances of an "accidental" data leakage to prevent privacy violations.

[1]In fact, our metric can also be used on a pipeline where DP is not employed, e.g. to evaluate the efficacy of heuristic mitigations.

Based on the discussion above, we consider a practical threat model that is relevant to the language models deployed in practice. We assume a black-box access, where a curious or malevolent user can query a pre-trained and deployed language model on any sequence of tokens $w_1, \ldots, w_i$ and receive the top-k predictions[2] returned by the model for the next token w_{i+1}[3].

The threat model allows a curious user to know whether any sensitive information in their data is leaked by the model. Therefore, the data owner can use any prefix in their data to query the model. The threat model also includes the case of a malevolent user, who can input directed queries in order to extract sensitive information about a targeted user.

3 TRAINING DATA LEAKAGE ANALYSIS

In this section, we introduce our framework to investigate a model trained on user content for the purpose of user-level privacy protection. We fix the notation first.

Notation. For a language model trained on user content, let $\mathcal{D} = \cup_{i\in\{1,2,\ldots,n\}} \mathcal{D}_i$ be the training data, which is the union of all user content where $\mathcal{D}_i$ corresponds to the content of user i. Without loss of generality, we assume that $\mathcal{D}_i$ contains sequence(s) of tokens $w_1, w_2, \ldots$ of arbitrary length.

We introduce our training data leakage analysis on a language model after being trained on the training data $\mathcal{D}$. The first step of our framework is to run the model through the training data and collect its correct predictions in the training data. We illustrate this step with an example in Fig. 2. This collection consists of sequences of tokens where the correct prediction is observed in top-k predictions of the model consecutively. We emphasize that consecutive correct predictions is an important phenomenon because the longer the model leaks a training sequence $w_{i+1}, w_{i+2}, \ldots$ having seen the context $w_1, \ldots, w_i$, the more it discloses user content, causing privacy concerns. Therefore, we do not break sequences where the model provides correct predictions consecutively and collect all such sequences in the training data. In Algorithm 1 we provide the pseudo-code to collect the correct predictions of the model as described above. Let us denote this collection as $\mathcal{S}$.

As we are considering user-level privacy, in the next step we count for each sequence in $\mathcal{S}$ the number of distinct users for which the sequence is found in their data. In other words, we generate a dictionary where each sequence in $\mathcal{S}$ is mapped to the number of users having this sequence in their dataset. We note that sequences for which the user count is large are indicative of general population-level phrases, which is beneficial for the model to learn to achieve domain adaptation. On the other extreme, sequences for which the user count is one is inarguably the case with the most potential to result in privacy violations since leakage of a unique content of a user may lead to singling out of that user.

[2]The parameter k depends on the application as in Fig. 1.
[3]We note that even the availability of the next token prediction(s) may not always be the case if the model does not return any prediction under certain conditions (e.g. when the prediction score is below a pre-fixed triggering threshold [10]). Our work is trivially applicable in such settings as well.
[4]Assuming a single sequence for simplicity, and without loss of generality.

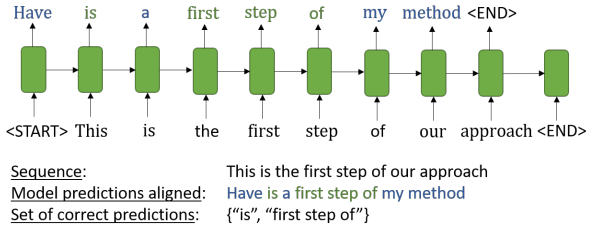

Figure 2: An illustration of the collection of correct model predictions. We run the model through each sequence in the training data and obtain the top-k (top-1 in this example) prediction(s). We then collect the sequence of tokens where the model consecutively provides the correct prediction.

Algorithm 1 The collection of correct model predictions.

Input: A language model $LM(\cdot)$ and the corresponding training data $\mathcal{D}$
Output: The (multi)set $\mathcal{S}$ of correct predictions
Initialize $\mathcal{S} = []$
for $i = 1$ **to** n **do**
  Initialize $W =$ ""
  Let $D_i = [w_1, \ldots, w_{|D_i|}]$[4]
  for $l = 1$ **to** $|D_i|$ **do**
    Obtain top-k predictions $preds = LM(D_i[:l])$
    if $w_l \in preds$ **then**
      Append w_l to W
    else if $W \neq$ "" **then**
      Append W to $\mathcal{S}$ and initialize $W =$ ""
    end if
  end for
end for

Let us denote the set of sequences in $\mathcal{S}$ that are unique to a user as $\mathcal{S}_{\text{uniq}}$. These unique sequences could potentially be learned due to the presence of their corresponding users in the training data. Therefore, motivated by the notion of plausible deniability, we attempt to understand whether these predictions of the private model could have been made had these users were not in the training set. In this regard, we introduce a reference model that is trained on a dataset containing no user in the set $\mathcal{S}_{\text{uniq}}$. Our privacy metric, the worst-case *leakage epsilon*, is defined as follows:

$$\epsilon_l \triangleq \max_{w \in \mathcal{S}_{\text{uniq}}} \log\left(\frac{\text{PP}_{\text{reference}}(w)}{\text{PP}_{\text{private}}(w)}\right), \tag{1}$$

where $\text{PP}_{\text{model}}(\cdot)$ denotes to the perplexity of a sequence given by the model. Our privacy metric measures the perplexity ratio with respect to a reference model maximized over the sequences in the set $\mathcal{S}_{\text{uniq}}$ to capture the worst-case scenario.

We note that a smaller ϵ_l for a private model translates into a better privacy protection. This is because the unique sequences leaked by the model will have relatively similar perplexities with respect to a reference model, which is trained on a set that does not include any user in $\mathcal{S}_{\text{uniq}}$, therefore, providing plausible deniability for all the users in the private training set.

4 CASE STUDY: TAB ATTACK

We study a large-scale example as a realistic setup for the deployed language models in practice. We consider an attack setting that has access to top-1 predictions of a language model. Having in mind the text auto-completion feature in emails where the predictions are applied by pressing the TAB key on the keyboard (see Fig. 1), we dub this as the *tab attack*. We investigate the unique sequences ($\mathcal{S}_{\text{uniq}}$) that could be leaked via the tab attack when the model is queried with the corresponding context. We apply our privacy metric over $\mathcal{S}_{\text{uniq}}$ to assess the attack surface under the tab attack threat model.

Dataset. We use a large dataset of Reddit posts, as decribed by Al-Rfou et al. [2], that contains 140M sentences from 4.4M users for a randomly chosen month (Oct 2018). It is randomly split into 90% training and 10% validation sets. We provide three sets of language models trained on this private Reddit dataset.

(1) A language model trained on the Reddit dataset. This will be referred to as *Private LM* in our results.
(2) A language model trained on the Reddit dataset with differential privacy [1, 24]. We take three snapshots of the model during training, corresponding to three DP language models with epsilons 3.28, 4.68, and 6.20[5]. The training begins with a random initialization of the weights. The models will be referred to as *DP-LM RanIni* $\epsilon = \cdot$.
(3) A language model trained on the Reddit dataset with differential privacy. Here, the model weights are initialized from a public model trained on Google News dataset [9]. It has been shown that transfer learning helps obtaining strong privacy guarantees with a minor cost in utility [1, 24, 31, 39]. We similarly take two snapshots of the model during training, corresponding to two DP language models with epsilons 2.98 and 6.68. These models and the public model will be referred to as *DP-LM PubIni* $\epsilon = \cdot$ and *Public LM* respectively.

The model architecture is same for all these models and the details are specified below.

Model. We use a one-layer GRU model as the language model for the next-word prediction task. The embedding size is set to 160 and the hidden size to 512, and the vocabulary is fixed to the most frequent 10k words in the training corpus (out of 3.2M words). We use the Adamax optimizer with the learning rate set to 1e-3 and the batch size is set to 3072 in the DP training and to 512 otherwise.

Reference Model. The reference model in Eq. (1) is taken as follows. For a given private model, we generate the set $\mathcal{S}_{\text{uniq}}$ and take the users in $\mathcal{S}_{\text{uniq}}$ and remove all their data from the training data. We subsequently train a new model using exactly same procedure as the original model on the remaining users. We consider the new model as the reference model in Eq. (1) since it has not seen any data of users in $\mathcal{S}_{\text{uniq}}$ during its training.

We provide in Table 1 the performances of the models and the result of the tab attack for each of them. We discuss the results of this experiment in what follows.

We observe from Table 1 that the private LM that is trained without DP leaks a huge number of unique sequences (3757) from

[5]The models satisfy user-level DP and $\delta \lesssim 1/(\#\text{ users})$ same for all models.

Table 1: Results of training RNN-based language models on Reddit dataset. We provide the perplexity and accuracy on the validation set to compare the performances of the models. In the next column, we provide the number of unique sequences ($|\mathcal{S}_{\text{uniq}}|$) for each model. We calculate leakage epsilon ϵ_l for some of the models for comparison in the last column.

Model	Val perp	Val acc (%)	# unique seq. ($|\mathcal{S}_{\text{uniq}}|$)	ϵ_l
Private LM	69.4	23.7	3757	17.75
DP-LM RanIni $\epsilon = 3.28$	290.0	14.5	0	-
DP-LM RanIni $\epsilon = 4.68$	130.3	19.6	5	-
DP-LM RanIni $\epsilon = 6.20$	107.8	20.8	11	0.64
Public LM	757.5	13.1	159	-
DP-LM PubIni $\epsilon = 2.98$	183.1	19.7	157	0.29
DP-LM PubIni $\epsilon = 6.68$	92.8	22.2	246	1.33

the training data. There are 759 unique sequences for which the number of tokens is larger than 9. A majority of these examples are coming from highly-repeated sentences (728 of these sequences are repeated somewhere between 50-34372 times) by the bots in the Reddit dataset[6]. Expectedly, the resulting $\epsilon_l = 17.75$ is very large and it does not offer any reasonable plausible deniability. We calculated ϵ_l among the unique sequences that do not repeat more than once and found that it is 4.60. This shows the importance of de-duplication at a granular level (e.g. removal of sentence duplicates) as also observed in [7, 8, 21].

For the DP-LMs that are snapshots of a model trained with random initialization of weights, we observe a small number of unique sequences leaked by the models. Interestingly, we get no unique sequence with the first one having $\epsilon = 3.28$, although there is a high cost in terms of utility. We observe the efficacy of user-level DP training by noting that the unique sequences with large repetitions that were memorized by the private model have all disappeared with DP-LMs.

For the DP-LMs initialized from a public model, we observe relatively larger number of unique sequences leaked by the models. However, this is not surprising as the public model itself can predict 159 unique sequences in the private data, without seeing any private data in its training. Since the DP training is initialized from the public model, it should be expected to yield a larger number of unique sequences. This shows that our privacy metric leakage epsilon ϵ_l may provide a better ground for a fair comparison of models trained in different ways (e.g. random initialization vs. transfer learning).

We calculate leakage epsilon ϵ_l for three DP-LMs for comparison. We observe that DP-LM PubIni $\epsilon = 2.98$ model has $\epsilon_l = 0.29$, much smaller than the models DP-LM RanIni $\epsilon = 6.20$ with $\epsilon_l = 0.64$ and DP-LM PubIni $\epsilon = 6.68$ with $\epsilon_l = 1.33$. This is expected since $\epsilon = 2.98$ provides a much stronger privacy guarantee compared to $\epsilon = 6.20$ and $\epsilon = 6.68$. We note that all three models have quite small ϵ_l values, indicating that the unique sequences leaked by these models can also be simply learned from other users because they have similar perplexities with respect to the reference model. This offers a reasonable plausible deniability even when the DP-ϵ value is relatively large such as $\epsilon = 6.68$ and does not provide a strong theoretical privacy guarantee.

[6] An example of a unique sequence memorized by the model is "has been automatically removed because the title does not include one of the required tags ." repeated 5377 times in the bot's data.

5 RELATED WORK AND CONCLUSION

A wide body of work has demonstrated privacy issues in general for machine learning models trained on personal data. Language models are among the most to suffer as they are capable of generating text which may potentially leak sensitive user content and lead to serious privacy violations.

Zhang et al. [45] show that deep learning models can achieve perfect accuracy even on randomly labeled data. Such memorization capability may in fact be needed to achieve near-optimal accuracy on test data when the data distribution is long-tailed as recently shown by Brown et al. [5], Feldman [14]. Unfortunately this can lead to a successful training data extraction attack, as in the case for the work [8] that can recover training examples from the GPT-2 language model [33]. In their method, Carlini et al. [8] generate a list of sequences by sampling from the GPT-2 language model and then curate it by using the perplexity measure. In a related line of work which exploits the transfer learning setup, Zanella-Béguelin et al. [44] have demonstrated that having simultaneous black-box access to the pre-trained and fine-tuned language models allows them to extract rare sequences from the smaller and typically more sensitive fine-tuning dataset. Both attacks rely on the model output beyond top-1 or top-3 predictions along with the perplexity measure. Access to this information may easily be restricted in deployed language models. Nevertheless, there are serious privacy concerns since the attacks can extract personally identifiable information even if they are present in one document in the training data. We believe that our proposed procedure for privacy investigations of a language model trained on user content could be very beneficial to protect user-level privacy in the presence of such attacks.

On the other hand, Carlini et al. [7] introduced the exposure metric to quantitatively assess the unintentional memorization phenomenon occurring in generative sequence models. They do so by inserting randomly-chosen canary sequences a varying number of times into the training data and measuring the relative difference in perplexity between inserted canaries and non-inserted random sequences. Our work is complementary in the sense that we are investigating the information leaked from user content in the training data, having in mind a strong threat model where one can query the language model with the precise context appearing in the training data. We believe that our proposed metric along with the exposure metric can be employed together to provide strong privacy guarantees for a deployed language model.

Another line of work has studied the vulnerability of machine learning models to membership inference attack [11, 15–17, 22, 23, 26, 28, 34–37, 40, 43]. The goal is to determine if a particular data record (or more generally data of a given user) belongs to the training set of the model. Although being an indirect leakage, membership inference is considered as a confidentiality violation and potential threat to the training data from models [27]. A recent work [29] used a similar idea to calculate lower bounds for the privacy offered by DP-SGD under various settings.

The main framework with theoretical guarantees for user-level privacy is the application of differential privacy (DP) [13] to model training. DP makes provable guarantees about the privacy of a stochastic function of a given dataset. Differentially private stochastic gradient descent (DP-SGD) has been developed and applied to training machine learning models [1, 38]. This is an active area of research with the goal of pushing the frontiers of privacy-utility trade-off for deep neural networks.

5.1 Conclusion

This work introduced a methodology to investigate information leaked by a language model from its training data in terms of privacy. Based on the notion of plausible deniability offered by DP, we proposed a metric that could be used to quantify how well the users participating in the training data enjoy plausible deniability in the private model to be deployed. We believe our framework can be incorporated into the training platform of language models that would help assess the model from the perspective of privacy, along with its utility.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *ACM CCS*.

[2] Rami Al-Rfou, Marc Pickett, Javier Snaider, Y.-H. Sung, Brian Strope, and Ray Kurzweil. 2016. Conversational Contextual Cues: The Case of Personalization and History for Response Ranking. arXiv:1606.00372 [cs.CL]

[3] Apple. [n.d.]. Apple Differential Privacy Technical Overview. https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

[4] Art. 29 WP. 2014. Opinion 05/2014 on "Anonymisation Techniques". https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf

[5] Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. 2020. When is Memorization of Irrelevant Training Data Necessary for High-Accuracy Learning? *arXiv preprint arXiv:2012.06421* (2020).

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, and Melanie Subbiah. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165* (2020).

[7] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *USENIX Security 2019*.

[8] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting Training Data from Large Language Models. *arXiv preprint arXiv:2012.07805* (2020).

[9] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. http://arxiv.org/abs/1312.3005

[10] Mia Xu Chen, Benjamin N. Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Zhifeng Chen, Timothy Sohn, and Yonghui Wu. 2019. Gmail Smart Compose: Real-Time Assisted Writing *(KDD '19)*. New York, NY, USA, 2287–2295.

[11] Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2020. Label-Only Membership Inference Attacks. *arXiv preprint arXiv:2007.14321* (2020).

[12] Aloni Cohen and Kobbi Nissim. 2020. Towards formalizing the GDPR's notion of singling out. *Proceedings of the National Academy of Sciences* 117, 15 (2020), 8344–8352. https://www.pnas.org/content/117/15/8344

[13] Cynthia Dwork. 2011. Differential privacy. *Encyclopedia of Cryptography and Security* (2011).

[14] Vitaly Feldman. 2020. Does Learning Require Memorization? A Short Tale about a Long Tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing* (Chicago, IL, USA) *(STOC 2020)*. New York, NY, USA, 954–959.

[15] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2019. LOGAN: Membership Inference Attacks Against Generative Models. *Proceedings on Privacy Enhancing Technologies* 2019, 1 (2019), 133 – 152.

[16] Sorami Hisamoto, Matt Post, and Kevin Duh. 2020. Membership Inference Attacks on Sequence-to-Sequence Models: Is My Data In Your Machine Translation System? *TACL* 8 (2020), 49–63.

[17] Paul Irolla and Grégory Châtel. 2019. Demystifying the Membership Inference Attack. In *2019 12th CMI Conf. on Cybersecurity and Privacy (CMI)*. 1–7.

[18] Daniel Kifer, Solomon Messing, Aaron Roth, Abhradeep Thakurta, and Danfeng Zhang. 2020. Guidelines for Implementing and Auditing Differentially Private Systems. *arXiv preprint arXiv:2002.04049* (2020).

[19] Janardhan Kulkarni and Sergey Yekhanin. 2017. Collecting telemetry data privately. https://www.microsoft.com/en-us/research/blog/collecting-telemetry-data-privately/

[20] Paul Lambert. 2018. SUBJECT: Write emails faster with Smart Compose in Gmail. https://www.blog.google/products/gmail/subject-write-emails-faster-smart-compose-gmail

[21] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating Training Data Makes Language Models Better. *arXiv preprint arXiv:2107.06499* (2021).

[22] Klas Leino and Matt Fredrikson. [n.d.]. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. *29th USENIX Security Symposium 2020* ([n. d.]).

[23] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. 2018. Understanding Membership Inferences on Well-Generalized Learning Models. *arXiv preprint arXiv:1802.04889* (2018).

[24] Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations (ICLR)*.

[25] Microsoft SwiftKey. [n.d.]. https://www.microsoft.com/en-us/swiftkey

[26] Fatemehsadat Mireshghallah, Huseyin A. Inan, Marcello Hasegawa, Victor Rühle, Taylor Berg-Kirkpatrick, and Robert Sim. 2021. Privacy Regularization: Joint Privacy-Utility Optimization in Language Models. *arXiv preprint arXiv:2103.07567* (2021).

[27] Sasi Kumar Murakonda and Reza Shokri. 2020. ML Privacy Meter: Aiding Regulatory Compliance by Quantifying the Privacy Risks of Machine Learning. *arXiv preprint arXiv:2007.09339* (2020).

[28] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. 739–753.

[29] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. 2021. Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning. *arXiv preprint arXiv:2101.04535* (2021).

[30] M.E.J. Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics* 46, 5 (2005), 323–351.

[31] Nicolas Papernot, Steve Chien, Shuang Song, Abhradeep Thakurta, and Ulfar Erlingsson. 2020. Making the Shoe Fit: Architectures, Initializations, and Tuning for Learning with Privacy. https://openreview.net/forum?id=rJg851rYwH

[32] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 2463–2473.

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

[34] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Herve Jegou. 2019. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. Long Beach, California, USA, 5558–5567.

[35] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. 2018. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. *arXiv preprint arXiv:1806.01246* (2018).

[36] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*. 3–18.

[37] Congzheng Song and Vitaly Shmatikov. 2019. Auditing Data Provenance in Text-Generation Models. In *KDD*.

[38] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. 2013. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conf. on Signal and Information Processing*. 245–248.

[39] Florian Tramèr and Dan Boneh. 2020. Differentially Private Learning Needs Better Features (or Much More Data). *arXiv preprint arXiv:2011.11660* (2020).

[40] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2018. Towards Demystifying Membership Inference Attacks. *arXiv preprint arXiv:1807.09173* (2018).

[41] White House Office of Science and Technology Policy (OSTP). 2019. Guidance for Regulation of Artificial Intelligence Applications. https://www.whitehouse.gov/wp-content/uploads/2020/01/Draft-OMB-Memo-on-Regulation-of-AI-1-7-19.pdf

[42] Royce Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. 2020. Differentially Private SQL with Bounded User Contribution.

[43] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. 268–282.

[44] Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. 2020. Analyzing Information Leakage of Updates to Natural Language Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 363–375.

[45] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. *ICLR* (2017).

[46] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 21st ACM Conference on Computer and Communications Security*. Scottsdale, Arizona. https://arxiv.org/abs/1407.6981