Reconstruction Attack on Instance Encoding for Language Understanding

Shangyu Xie
Illinois Institute of Technology
sxie14@hawk.iit.edu

Yuan Hong
Illinois Institute of Technology
yuan.hong@iit.edu

Abstract

A private learning scheme TextHide was recently proposed to protect the private text data during the training phase via so-called instance encoding. We propose a novel reconstruction attack to break TextHide by recovering the private training data, and thus unveil the privacy risks of instance encoding. We have experimentally validated the effectiveness of the reconstruction attack with two commonly-used datasets for sentence classification. Our attack would advance the development of privacy preserving machine learning in the context of natural language processing.

1 Introduction

With the development of deep learning technologies, a large number of applications in various domains (e.g., image classification and NLP) have been greatly promoted with significantly improved performance. However, this also arouses serious privacy concerns since a large portion of the training data are usually collected from individuals. For instance, the diagnosis systems in hospitals or healthcare institutions will be trained on the patients' private data, such as medical history (Pham et al., 2017), and radiology medical images (Hosny et al., 2018). In addition, it has been reported that the input keyboard prediction model can be trained with the users' data on mobile devices (Hard et al., 2018), and the assisted composing function for emails/texts can be trained with users' personal messages (Chen et al., 2019).

There have been various works on protecting users' data privacy during the training, which are categorized into two main types: 1) composing cryptographic protocols for securely training the data (Bonawitz et al., 2016; Mohassel and Zhang, 2017; Mohassel and Rindal, 2018) which result in high computational and communication costs in general; 2) leveraging the differential privacy techniques (Dwork et al., 2006; Chaudhuri et al., 2011; Hong et al., 2015; Abadi et al., 2016) to prevent the information leakage which typically cause significant accuracy loss. Despite the above demerits, both types of methods can ensure provable privacy guarantees for the training data. This also raises the question: *are there any private learning schemes which can preserve both accuracy and efficiency?*

To this end, there are several techniques (Huang et al., 2020a,b) which privately train the model via the so-called instance encoding scheme, by encoding the local data into a somewhat "encrypted" (encoded) data with a *mixup* scheme (Zhang et al., 2018), and directly training the model on the encoded data. Data privacy is claimed to be well preserved through the encoding method while only causing minor accuracy loss with the merit of the *mixup* scheme. In this paper, we study the privacy risks of the instance encoding scheme, and show that the instance encoding cannot provide sufficient privacy protection as the conventional cryptographic techniques against well-designed attacks. Specifically, we design a reconstruction attack to recover the original data from the privately encoded data. We focus on the instance encoding in language understanding, i.e., TextHide (Huang et al., 2020a) as the state-of-the-art technique.

2 TextHide

The TextHide (Huang et al., 2020a) aims to protect the private text data under the federated learning setting. First, the input text is pre-processed with a BERT transformer encoder to output the corresponding text representation. Then, for "encryption", TextHide will apply the instance encoding to mix up the original text representation with some randomly selected text (representations), which will be fed into the training model of various downstream language understanding tasks, e.g., classification, and question answering. Formally, given the input text x_i with the label y_i, we denote the text representation as $e_i = \phi(x_i)$, where $\phi(\cdot)$ is

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2038–2044
November 7–11, 2021.

a pre-tuned BERT model. The private instance encoded data $\widetilde{e}_i$ can be generated as below:

$$\widetilde{e}_i = \sigma \circ \sum_{j=1}^{K} \lambda_j e_j \quad (1)$$

where λ_j is chosen uniformly at random such that $\sum_j^K \lambda_j = 1$, the sign-flipping mask $\sigma \in \{-1, 1\}^d$ is also chosen uniformly at random, and d denotes the dimension of the encoding vector. $\circ$ represents the Hardamard (element-wise) multiplication, and K is the number of combined mix encoding data (as the security parameter). Therefore, the label (one-hot vector) $\widetilde{y}_i$ of the $\widetilde{e}_i$ is updated as: $\widetilde{y}_i = \sum_{j=1}^{K} \lambda_j y_j$, which is the element-wise addition across y_j. Then, for the training with one data batch $\mathcal{B}$, each data $(x_i, y_i) \in \mathcal{B}$ will be privately encoded as Equation 1, where the K data for mixup are randomly sampled from the batch $\mathcal{B}$. TextHide also specifies another parameter m as the size of the mask pool to facilitate the security of instance encoding against the reconstruction attacks. These formalize the (m, K)-TextHide (Algorithm 1 in (Huang et al., 2020a)), which can be integrated into the language training process to ensure text privacy. For instance, $(m = 0, K = 1)$ is the baseline training setting without protection. A larger K will sacrifice some accuracy while improving the privacy (higher costs on recovering the original data), which reflects the trade-off between privacy and accuracy for private training.

Furthermore, TextHide can utilize another dataset X_{public} (usually a large public corpus, e.g., Wikipedia) for mixup, where such mixup works similar to a *random oracle* in the cryptography domain.[1] Specifically, TextHide will mix up about one half $\lfloor K/2 \rfloor$ public data with the private original data, then Equation 1 is updated as:

$$\widetilde{e}_i = \sigma \circ \Big(\sum_{j=1}^{\lfloor K/2 \rfloor} \lambda_j e_j + \sum_{j=\lfloor K/2 \rfloor + 1}^{K} \lambda_j e_j^p \Big) \quad (2)$$

where $e_j^p = \phi(x_j^p), x_j^p \in X_{public}$ (randomly sampled). As a consequence, the mixed label $\widetilde{y}$ is computed by normalization with the labels of the private data (public data usually do not have labels):

$$\widetilde{y}_i = \frac{\sum_{j=1}^{[K/2]} \lambda_j y_j}{\sum_{j=1}^{[K/2]} \lambda_j} \quad (3)$$

[1]The privacy notion provided by mixup in TextHide is based on a *k-vector subset sum* (Abboud and Lewi, 2013) oracle, which would require $O(n^{k/2})$ efforts to break.

In practice, given the original training dataset (denoted as X), each data $(x_i, y_i) \in X$ will be encoded for n times (usually equal to the number of training epochs).

3 Attack Preliminaries

Privacy-Enhancing Schemes. As mentioned before, both cryptographic protocols and differential privacy can provide provable privacy guarantees for protecting the private data. On the one hand, for cryptographic solutions, the data is usually protected by the encryption schemes, e.g., fully homomorphic encryption (FHE) (Gentry, 2009; Cheon et al., 2017), where the security of schemes depends on some hard mathematical problems. Normally, to prove the security of the encryption scheme, we need to formulate a security game, e.g., IND-CPA (Goldreich, 2009), where an attacker with repeating many operations polynomially (w.r.t. the size of the security parameter) cannot do better than randomly guessing. It should be noted that the newly proposed instance encoding schemes are claimed to work as the encryption scheme for privacy protection (Huang et al., 2020a,b), but fail to provide such provable security guarantees.

On the other hand, differential privacy (Dwork et al., 2006, 2014; Mohammady et al., 2020) can statistically protect the individual information from being identified (i.e., against identification attacks (Dinur and Nissim, 2003)) by injecting well-calibrated noise to the original values. For example, differential privacy can help to defend against so-called membership inference attacks (Shokri et al., 2017) in the machine learning such that an attacker cannot determine whether any specific individual information is in the dataset or not.

Privacy Attacks. The attacks on the data privacy in ML training are generally referred to membership inference attacks (Shokri et al., 2017; Salem et al., 2018; Nasr et al., 2019; Hisamoto et al., 2020; Song and Raghunathan, 2020), where an adversary can know whether a given data points was used to train the model or not. In addition, model inversion attacks (Fredrikson et al., 2015; Wu et al., 2016; Zhu et al., 2019) can reconstruct a group of representative data points from the training set, e.g., utilizing gradients (Zhu et al., 2019). Our attack on TextHide works closely as the reconstruction attack (Dinur and Nissim, 2003; Carlini et al., 2020a), which aims to reconstruct the original data/information from the protected data

(privately encoded data). Note that Carlini et al. (Carlini et al., 2020a) attacks the instance encoding on images while we extend this method to the language understanding domain.

Attack Setting. We assume that the attacker have full knowledge of the public dataset X_{public} and the embedding model for downstream ML tasks. Besides, we assume that the attacker can obtain the private dataset (but unaware of the specific data for the training). Note that we need to consider the worst case (attacker) to evaluate the vulnerabilities of the privacy-enhancing schemes. That is, the strong knowledge (e.g., embedding model and private training dataset) can be accessed by a skilled attacker armed with any background knowledge. For instance, such private training dataset can be machine-generated. Specifically, if the dataset involves personal conversations, then the attacker can utilize some language models to generate a large set of commonly-used dialogs as the private training dataset. The attacker can also leverage some advanced inference attacks, e.g., side-channel or public essays to derive some sentences.

Attack Goal. Given a privately encoded dataset $\widetilde{\mathcal{E}}$ (including the mixed label $\widetilde{y}$), the attacker aims to reconstruct the original data vector $e \in \mathcal{E}$, where $\mathcal{E}$ is the set of the original data vectors. W.l.o.g., we consider the basic mixup case that the two original data vectors are used for private encoding, i.e., for one encoded data $\widetilde{e}_i$, it will be constructed on two original data e_{j_1} and e_{j_2}. Then, we denote a mapping function for the attack as $\mathcal{A}_m : \widetilde{e}_i \in \widetilde{\mathcal{E}} \rightarrow \{e_{j_1}, e_{j_2}\} \in \mathcal{E} \times \mathcal{E}$. Thus, given $\mathcal{A}_m(\widetilde{e}_i) = \{e_{j_1}, e_{j_2}\}$, the attacker seeks to derive such mapping function. Note that our attack focuses on reconstructing the text representation vectors (processed by the language understanding model, e.g., BERT) and then we can utilize the model inversion attack (Zhu et al., 2019) to recover the raw text, i.e., $x_i = \phi^{-1}(e_i)$.

4 Attack Methodology

4.1 Overview of The Attack

Our proposed attack consists of three main steps:

1. Removing the sign-flipping mask σ. We first nullify the sign-flipping step for encoding by taking the absolute value of the encoded data $\widetilde{e} \in \widetilde{\mathcal{E}}$ as: $\widetilde{\mathcal{E}} \leftarrow \{abs(\widetilde{e}), \widetilde{e} \in \widetilde{\mathcal{E}}\}$.

2. Revealing the mapping function $\mathcal{A}_m$ to map the encoded data vector $\widetilde{\mathcal{E}}$ to the original data vector via clustering (Section 4.2).

3. Reconstructing the original text representation vector e_i (by computing the λ_i) given the mapping function $\mathcal{A}_m$ (Section 4.3).

4.2 Revealing Mapping Function

The main procedure of this step is clustering the encoded text vectors and mapping the clusters back to the original text vectors. Given a set of original data instances $|X|$ and every data instance will be encoded n times. Since each encoded text vector $\widetilde{e}_i$ is corresponding to the two original data (i.e., $\mathcal{A}_m(\widetilde{e}_i) = \{e_{j_1}, e_{j_2}\}$), the clustering result would expect to be $|X|$ clusters of size $2 * n$ encoded data vectors (the size of encoded data $\widetilde{\mathcal{E}}$ is $|X| * n$).

1) Compute Similarity Score. For the cluster of $\widetilde{\mathcal{E}}$, we first compute a similarity score $s \in [0, 1]$ among the two privately encoded data $\widetilde{e}_i$ and $\widetilde{e}_j$: if $\mathcal{A}_m(\widetilde{e}_i) \cap \mathcal{A}_m(\widetilde{e}_j) \neq \emptyset$, $s = 1$ (or close to 1), otherwise 0 (or close to 0). To compute the similarity score s, we train a neural network model $f(\cdot)$ by inputting two privately encoded vectors $(\widetilde{e}_i, \widetilde{e}_j)$, and $f(\widetilde{e}_i, \widetilde{e}_j) = \{0, 1\}$. The two vectors will be stacked together (e.g., for $d \times 1$ encoded vector, the input will be $d \times 2$).

Specifically, we utilize a vanilla MLP model trained with Adam (learning rate 0.01) on the cross-entropy loss. We use the MNLI dataset (around 393k examples with all labels removed) (Williams et al., 2018) as the public dataset , and Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019), and Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) as the private dataset. Then, we construct a large-scale training data pairs encoded with the above datasets by TextHide, which are labeled accordingly (1 if encoded with the same original text data; otherwise 0). The final model can achieve 94% accuracy.

Notice that reconstructing model $f(\cdot)$ by computing the similarity scores between two privately encoded data is based on a key hypothesis: *given any instance encoding scheme which achieves a high accuracy, the privacy guarantee would be somewhat weak (since the original information should be preserved with high accuracy)*. In other words, if TextHide ensures high accuracy in the downstream tasks (e.g., sentence classification), then the instance encoded data can also be "learned" to recover the original text data (model $f(\cdot)$ can be viewed as a downstream task in NLP). We identify

this as an intrinsic vulnerability of such instance encoding schemes, which can be exploited to launch the reconstruction attack.

2) Clustering. Given the similarity model, we can compute the similarity scores on all pairs of the encoded data $(\widetilde{e}_i, \widetilde{e}_j)$ ($|\widetilde{\mathcal{E}}|^2$ pairs in total). This procedure can be computationally efficient. To find $|X|$ clusters (exclusive), denoted the cluster set as $\{C_p, p \in [1, |X|]\}$ w.r.t. $|X|$ original text vectors, we formulate the objective function as:

$$\max \sum_{p=1}^{|X|} \sum_{\widetilde{e}_i, \widetilde{e}_j \in C_p} f(\widetilde{e}_i, \widetilde{e}_j) \tag{4}$$

Ideally, the size of each cluster should be exactly $2n$, and any two encoded data $(\widetilde{e}_i, \widetilde{e}_j)$ in every cluster C_p should satisfy $f(\widetilde{e}_i, \widetilde{e}_j) = 1$ (or close to 1). Following K-NN, we can design a greedy method to iteratively update $|X|$ clusters by selecting the encoded data which has the maximum average similarity score of all the data in the cluster. Furthermore, we can audit each cluster by checking the similarity scores among the encoded data and finally partition $\widetilde{\mathcal{E}}$ into $|X|$ clusters.

4.3 Reconstructing Original Text Vectors

After deriving the mapping function from the encoded data to the original data, we can reconstruct the original data. Roughly we can sum up the absolute values of all the encoded vectors mapping to one given original data vector e and average it: $e' = \frac{1}{n} \sum abs(\widetilde{e}_i)$. The vector e' is approximately close to the original e based on two aspects: 1) the sign-flipping mask σ is removed by taking the absolute values; 2) the values of other irrelevant mixup text vectors can be "cancelled out" by the averaging (could also result in some noises added into the vector). Thus, we need to ensure that the recovered result is close to the original result with tolerable noises.

We first recover the values of the mix-up coefficients λ via the mix-up labels. Specifically, we can get the list of λ with the mix-up labels since TextHide utilizes one-hot vector labels. For example, given one TextHide label $(0.4, 0, 0, 0.6)$, we can directly derive λ_i, λ_j as $0.4, 0.6$ (Figure 1 in (Huang et al., 2020a)). Then, the attacker can directly retrieve the values of λ. Note that there exists one special case: the mixed two data could belong to the same class (the mixed label will only have one non-zero entry), and thus we can consider $\lambda_i = \lambda_j$.

After we compute the value of λ, we can reconstruct the original vector e by trying to inverse the mixup operation (Equation 2). Specifically, we denote Λ as an $|\mathcal{E}| \times |X|$ matrix. For each row of Λ, there are two non-zero entries i, j corresponding to the two mixup values λ_i and λ_j (other entries are 0). Denote the original text vectors as $\mathcal{X} = [e_1, \cdots, e_{|X|}]^T$ (with dimension $|X| \times d$), and the privately encoded vectors as $\mathcal{Y} = [\widetilde{e}_1, \cdots, \widetilde{e}_{|\mathcal{E}|}]^T$ (with dimension $|\mathcal{E}| \times d$). Then, Equation 2 can be updated as:

$$\Lambda \cdot \mathcal{X} = \mathcal{Y} + \epsilon \tag{5}$$

where ϵ denotes the potential introduced noises ($\mathcal{X}$ may not be exactly the original one). To compute $\mathcal{X}$, we can directly solve the above equation:

$$\mathcal{X} = \Lambda^{-1} \cdot \mathcal{Y} + \Lambda^{-1} \cdot \epsilon \tag{6}$$

Since the noise could subject to Gaussian distribution, the component $\Lambda^{-1} \cdot \epsilon \approx 0$ (the mean value would be close to 0, then we can average it). Furthermore, we can formulate another optimization to minimize the "extra" noise ϵ:

$$\min_{\mathcal{X}} ||\epsilon||_2^2 \quad s.t. \;\; \epsilon = \mathcal{Y} - \lambda \cdot \mathcal{X} \tag{7}$$

Thus, with the minimization of the noise, we can accurately derive $\mathcal{X}$ (close to the true value). It is worth noting that $\mathcal{X}$ includes the sign-flipping mask σ. Recall that we nullify the mask σ by taking the absolute value, then Equation 8 can be updated:

$$\min_{\mathcal{X}} ||\epsilon||_2^2 \quad s.t. \;\; \epsilon = abs(\mathcal{Y}) - \lambda \cdot abs(\mathcal{X}) \tag{8}$$

where abs is the element-wise absolute value function of the matrix $\mathcal{X}$ or $\mathcal{Y}$. To solve Equation 8, we can utilize the gradient descent to search the value of $\mathcal{X}$, and thus compute the ϵ based a fit solution of $\mathcal{X}$ (w.r.t. the objective function $||\epsilon||_2^2$). Note that there may exist several values of ϵ to satisfy the constraints, then we can heuristically search the value of ϵ entry by entry to get the smallest $||\epsilon||_2^2$. Since the attackers have the full knowledge of the pre-trained language model $\phi(\cdot)$, we can directly utilize model inversion attacks (Song and Raghunathan, 2020) to recover the original text.

5 Results and Analysis

We utilize the pre-trained BERT$_{base}$ model by (Devlin et al., 2019) (`https://github.com/`

K	1	2	4	6
CoLA	100%	88%	91%	93%
SST-2	100%	92%	95%	88%

Table 1: Attack success rate on the two datasets.

google-research/bert) as the language model to generate the text representations (the dimensionality d is 768). We evaluate our attack on two datasets for sentence classification: 1) Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019); 2) Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) (the private datasets). For the "public dataset", we use MNLI daset (Huang et al., 2020a). We utilize the open source code of TextHide (https://github.com/Hazelsuko07/TextHide) to construct the private dataset. We vary the parameter $k \in [1, 2, 4, 6]$ (the number of data for mixup). We keep the size of mask pool $m = 1$. Also, we evaluate the attack performance on varying the size of mask pool $m = [1, 16, 64, 256, 1024, 4096]$. For each dataset, we randomly select 100 data points and generate 5000 encoded data via TextHide. In our attack, we will try to reconstruct the original data from such 5000 encoded data by instance encoding. We report the attack success rate (the percentage of reconstructed data out of the original data). Note that our attack is independent of datasets/applications and hyper-parameter free.

Table 1 illustrates the attack results (the percentage of recovering original data) on the two datasets. We can observe that our proposed attack can almost recover the text vectors (high success rate). Moreover, while TextHide claims that the privacy will increase as K increases (while losing accuracy), the results show that the value of K does not impact privacy much. Similarly, Figure 1 shows that the mask cannot ensure privacy (but only increasing computational costs instead). Above all, the text vectors cannot be simply viewed as "real-number" vectors since they may still contain semantic meanings (features), which may help the attacker break the security oracle more efficiently.

6 Discussion

Privacy preserving machine learning (PPML) has been popular in industries under more and more restrictive data actions or regulations, e.g., General Data Protection Regulation (GDPR) in European Union. PPML could help the corporations improve business continuity while machine learning-

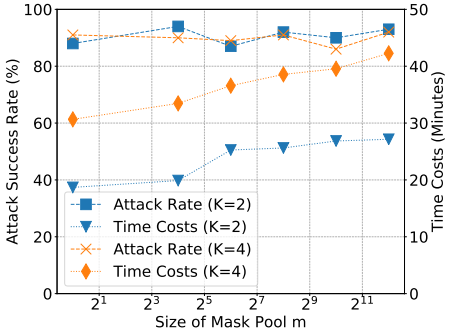

Figure 1: Attack success rate vs. size of mask pool m

based services deal with large amounts personal data/information, including text data-based applications such as the keyboard input prediction (Hard et al., 2018). Private instance encoding (e.g., TextHide) has been proposed to address privacy risks in such application scenarios. However, weak privacy guarantees provided by TextHide (e.g., against our proposed attack) may leak the personal information, and also violate privacy regulations and laws. This would cause severe sanctions and lose enterprise reputation from their customers.

As depicted earlier, a well-designed privacy-enhancing scheme must ensure provable privacy guarantee, and show its performance on data protection. Since TextHide is based on such mixup encoding method, it would be possible to apply differential privacy (Dwork et al., 2006) to the mixup encoding and thus to show similar indistinguishability of the privately encoded instances. This can defend against our reconstruction attacks to some extent (at least reducing the information disclosure). Another possible defense method is to filter sensitive data from the training data. However, this might degrade the model performance.

It is also worth noting that the intrinsic property of DNN model (i.e., memorization) can also be utilized to extract/recover training data from model itself, especially for language models (Carlini et al., 2020b; Lehman et al., 2021). Such works are orthogonal with our proposed attack since we focus more on the encoded data. Nevertheless, our attack can be integrated with such attacks to be more powerful on instance encoding schemes.

7 Conclusion

We proposed a novel reconstruction attack on a recent private learning scheme, TextHide in the NLP domain. We have experimentally shown that such scheme cannot provide rigorous privacy guarantee even though it obtains good accuracy.

Acknowledgements

This work is partially supported by the National Science Foundation (NSF) under the Grants No. CNS-1745894 and CNS-2046335. We are also grateful to the anonymous reviewers for their very constructive comments.

Ethical Impact

Data privacy topics (including privacy-enhancing technologies or attacks to breach data privacy) have been widely investigated in the machine learning-based applications. Such works should be carefully considered to be more ethical rather than harmful, especially for the attacks on breaking privacy-enhancing technologies. We think this matches with our case. Even though it is possible that our proposed attack can be further utilized to attack, we firmly believe that our attack can call more attention on the privacy-enhancing works and then motivate more advanced defense schemes in the language understanding domain.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Amir Abboud and Kevin Lewi. 2013. Exact weight subgraphs and the k-sum conjecture. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2016. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*.

Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Shuang Song, Abhradeep Thakurta, and Florian Tramer. 2020a. An attack on instahide: Is private learning possible with instance encoding? *arXiv preprint arXiv:2011.05315*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2020b. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3).

M. Chen, B. Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Y. Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Z. Chen, Timothy Sohn, and Yonghui Wu. 2019. Gmail smart compose: Real-time assisted writing. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic encryption for arithmetic of approximate numbers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 409–437. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, page 202–210, New York, NY, USA. Association for Computing Machinery.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA. Association for Computing Machinery.

Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178.

Oded Goldreich. 2009. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

Sorami Hisamoto, Matt Post, and Kevin Duh. 2020. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, 8:49–63.

Yuan Hong, Jaideep Vaidya, Haibing Lu, Panagiotis Karras, and Sanjay Goel. 2015. Collaborative search log sanitization: Toward differential privacy and boosted utility. *IEEE Trans. Dependable Secur. Comput.*, 12(5):504–518.

Ahmed Hosny, Chintan Parmar, John Quackenbush, Lawrence H Schwartz, and Hugo JWL Aerts. 2018. Artificial intelligence in radiology. *Nature Reviews Cancer*, 18(8):500–510.

Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, and Sanjeev Arora. 2020a. TextHide: Tackling data privacy in language understanding tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1368–1382, Online. Association for Computational Linguistics.

Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. 2020b. InstaHide: Instance-hiding schemes for private distributed learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4507–4518. PMLR.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron Wallace. 2021. Does BERT pretrained on clinical notes reveal sensitive data? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 946–959, Online. Association for Computational Linguistics.

Meisam Mohammady, Shangyu Xie, Yuan Hong, Mengyuan Zhang, Lingyu Wang, Makan Pourzandi, and Mourad Debbabi. 2020. R2dp: A universal and automated approach to optimizing the randomization mechanisms of differential privacy for utility metrics with no known optimal distributions. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 677–696, New York, NY, USA. Association for Computing Machinery.

Payman Mohassel and Peter Rindal. 2018. Aby3: A mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 35–52.

Payman Mohassel and Yupeng Zhang. 2017. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38.

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)*, pages 739–753. IEEE.

Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. 2017. Predicting healthcare trajectories from medical records: A deep learning approach. *Journal of biomedical informatics*, 69:218–229.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, 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*.

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)*, pages 3–18. IEEE.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 377–390.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F Naughton. 2016. A methodology for formalizing model-inversion attacks. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 355–370. IEEE.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.