Proposal for Master Thesis: The Use Of Knowledge Distillation To Limit Unintended Memorization In Fine-tuned Language Models

Andor Diera
Universität Ulm
andor.diera@uni-ulm.de

March 24, 2022

1 Motivation

The field of Machine Learning (ML) has seen an enormous growth in the past decade and revolutionized a wide variety of application domains. One of the main improvements in the area was the emergence of Deep Neural Networks (DNNs). With the rise of computational power and the increase in the amount of data available to researchers, DNNs became the forerunners of the recent successes in many ML domains, such as image recognition and natural language processing (NLP) [26]. Although the use of large volumes of training data is one of the main driving factor behind the great performance of modern ML models, publishing these models to the public raises some serious privacy concerns regarding private and confidential information present in the training data.

These privacy concerns are especially relevant for large Language Models (LMs), which form the basis of most state-of-the-art technologies in many NLP tasks. LMs are statistical models which assign probability to a sequence of word. These models are usually first trained in a task-agnostic self-supervised manner. Latest large LMs use a corpus size ranging from hundreds of gigabytes to several terabytes of text [4, 46] for this self-supervised training. The sheer size of these datasets makes it near impossible for researchers to remove all confidential information which may be present in the corpus, and a recent study has shown that it is possible to extract privacy sensitive information from these large LMs, even if that given information has only appeared once in the training corpus [7].

While the training cost of modern large LMs became so prohibitively expensive that only the biggest tech companies can afford it [53], pre-trained LMs are commonly used in businesses that work with huge amounts of text data. These businesses include banks, telecommunication and insurance companies which often work with privacy sensitive datasets. In practice, pre-trained LMs are usually fine-tuned on a dataset using some specific downstream task (such as text-classification, question-answering or natural language inference) before deployment [14]. The fine-tuning may mitigate some of the unintended memorization of the original data used in pre-training, but raises new concerns regarding the privacy sensitive information in the data used for the fine-tuning process [7].

Privacy Preserving Deep Learning (PPDL) is a common term used for methods aiming to mitigate general privacy concerns present in the use of DNNs. Multiple approaches have been proposed to achieve PPDL [39], but so far there is no perfect solution to this problem, with each method having its own challenges and limitations. The most popular techniques include Federated

Learning [36], the application of Differential Privacy (DP)[1, 69], encryption [20, 3], and data anonymization [55].

The aim of this thesis is to study the privacy preserving effects of Knowledge Distillation (KD) on fine-tuned LMs. Appyling KD is a highly popular way of obtaining efficient and lightweight NLP models that can be deployed with lower memory and computational requirements[51]. Seeing that model overfitting is one of the main indicators of a possible privacy leakage, and that smaller LMs are less prone to unintentionally memorizing training data [7], we hypothesize that the use of KD can also carry privacy preserving benefits. Furthermore, recent Data-Free KD methods[47, 33] introduced new techniques which could further improve the privacy preservation aspects of KD. Since KD is already commonly used as a way to compress models with only a minor loss in performance, we propose that it can be a viable option for privacy preservation that fits well to the standard pipeline of LM deployment and has a smaller trade-off in performance as current PPDL approaches.

2 Related Work

2.1 Modern Language Models

Modern large LMs rely on two core concepts that led to their dominance in most NLP tasks: the focus on the self-attention mechanism in the DNN architecture and the introduction of large scale task-agnostic pre-training to learn general language representation [63].

Self-attention is used for modeling dependencies between different parts of a sequence. A landmark study in 2017 [60] has shown that self-attention was the single most important part of the state-of-the-art NLP models of that time, and introduced a new family of models called Transformers, which rely solely on stacked layers of self-attention and feed-forward layers. Another great advantage of the Transformer architecture is that unlike a recurrent architecture, it allows for training parallelization.

The ability to parallelize training, alongside the constant increase in computational power allowed these models to train on larger datasets than once was possible. Since supervised training requires labeled data, self-supervised pre-training with supervised fine-tuning became the standard approach when using these models [35]. During pre-training, the model learns to predict the probability of the next word or token given a historical context of an input sequence. After pre-training the model is fine-tuned to a particular downstream task using supervised learning. The first Transformer which achieved great success on a large-range of NLP using this approach was the General Pre-trained Transformer (GPT) [44].

A major limitation of the GPT model is that during pre-training it learns a unidirectional language model. In this model tokens are restricted to only attend other tokens left to them, therefore it can only predict context to the left. A bidirectional approach has been proposed in the Bidirectional Encoder Representation for Transformers (BERT) model [14], which utilizes Masked Language Modeling (MLM) as a pre-training task. In MLM a token is randomly removed from the input sequence and the model is trained to predict the removed token.

Next to other recent large LMs such as XLNet [65] and T5 [46], the developers of GPT and BERT have also released new versions of their models. While the newer GPT models [45, 4], focus mostly on increasing the parameter count of the models, descendants of BERT proposed improved methods to optimize the pre-training procedure [30, 11] and to compress the model size while retaining performance [51, 24].

2.2 Privacy Attacks In Machine Learning

Privacy attacks in ML denote a specific type of adversarial attack, which aim to extract information from a ML model. Based on recent surveys in the field [13, 28, 48], these attacks can be divided into five main categories.

2.2.1 Membership Inference Attacks

The goal of a membership inference attack is to determine whether or not an individual data instance is part of the training dataset for a given model. This attack typically assumes a black-box query access to the model. The common approach to this type of attack is to use a shadow training technique to imitate the behavior of a specific target model. The trained inference model is then used to recognize differences on the target model predictions between inputs used for training and inputs not present in the training data [54]. Membership inference attacks have been shown to work on models used for supervised classification tasks [54] , Generative Adversarial Networks (GANs)[29], Variational Autoencoders(VAEs) [29], and the embedding layers of LMs [34].

2.2.2 Model Extraction Attacks

The adversarial aim of a model extraction attack is to duplicate (i.e., "steal") a given ML model. It achieves this by approximating a function f' that is the same as the function f of the target model [28]. A shadow training scheme has been shown to successfully extract popular ML models such as logistic regression, decision trees, and neural networks, using only a black-box query access [58]. Other works have proposed methods to extract information about hyperparameters [61] and properties of the architecture [40] in neural networks.

2.2.3 Model Inversion Attacks

The idea behind model inversion attacks states that an adversary can infer sensitive information about the input data using a target model's output. These attacks can be used to extract input features and/or reconstruct prototypes of a class in case the inferred feature characterize an entire class[13]. The first model inversion attack [18] was based on the assumption that the adversary has white-box access to a linear regression model, with some prior knowledge about the features of the training data. With the use of the output labels and known values of non-sensitive features this attack is capable of estimating values of a sensitive feature. This work was later extended to neural networks with a new type of model inversion attack [17], which reformulated the attack as an optimization problem where the objective function is based on the target model output and uses gradient descent in the input space to recover the input data point. This technique allows the adversary to reconstruct class prototypes (i.e faces in a facial recognition model) given a white-box access to the model and target labels with some auxiliary information of the training data.

2.2.4 Property Inference Attacks

The goal of property inference attacks is to infer some hidden property of a training dataset that the owner of the target model does not intend to share (such as feature distribution or training bias). Initially, property inference attacks were applied on discriminative models with white-box access [37, 42]. A more recent work has extended the method to work on generative models with black-box access [42].

2.2.5 Training Data Extraction Attacks

Training data extraction attacks aim to reconstruct training datapoints, but unlike model inversion attacks, the goal is to retrieve verbatim training examples and not just "fuzzy" class representatives

[7]. These attacks are best suited for generative sequence models such as LMs. Initially these attacks have been designed for small LMs using academic datasets [6, 66, 57]. The aim of these studies was to measure the presence of training datapoints when generating sequences that are irrelevant to the learning task and are unhelpful to improving model performance. A common approach to measure the extent of this *unintended memorization* is to insert so-called "canaries" (artificial datapoints) into the training datasets and quantify their occurrence during sequence completion [6]. Since these initial studies were based on smaller models trained with a high number of epochs, it was assumed that this kind of privacy leakage must be correlated with overfitting [66]. However, a follow-up study [7] using the GPT-2 model (which is trained on a very large corpus for only a few epochs) showed that even state-of-the-art large LMs are susceptible to these kinds of attacks. Using the pre-trained GPT-2 model, Carlini et al. were able to generate and select sequence samples which contained low *k-eidetic* data-points (data points that have a frequency of k in the training corpus).

2.3 Privacy Preserving Deep Learning

Since deep learning is a subfield of ML, most methods developed for privacy preserving ML can be also adapted to PPDL. Based on the literature [13, 28, 48] these methods can be divided into four main categories.

2.3.1 Encryption

Cryptography-based methods can be divided in to two subcategories, depending whether the target of the encryption is the training data [20] or the model[3]. Regardless of the target, existing approaches use homomorphic encryption, which is a special kind of encryption scheme that allows computations to be performed on encrypted data without decrypting it in advance [2]. Since training a DNN is already computationally expensive, adding homomorphic encryption to the process raises major challenges, as it increases training times by at least an order of magnitude [28]

2.3.2 Data Anonymization

Data Anonymization techniques aim to remove all Personally Identifiable Information (PII) from a dataset. The common approach to achieve this is to remove attributes that are identifiers and mask quasi-identifier attributes [64]. The popular k-anonimity algorithm [55] works by suppressing identifiers (i.e. replacing them with an asterisk) and generalizing quasi-identifiers with a broader category which has a frequency of at least k in the dataset. Although data anonymization techniques were developed for structured data, it is possible to adapt them to unstructured text data as well [22].

2.3.3 Differentially Private Learning

Differential Privacy (DP) is a rigorous mathematical definition of privacy in the context of statistical and machine learning analysis. It addresses the paradox of "learning nothing about an individual while learning useful information about the population" [16]. In ML, DP algorithms aim to obfuscate either the training data [68] or the model [50], by adding noise. Since DNN parameters are highly sensitive to noise, the best place for applying DP in deep learning is the gradients. Abadi et al. [1] proposed an efficient training algorithm with a modest privacy budget called Differentially Private Stochastic Gradient Descent (DPSGD). DPSGD ensures DP by cutting the gradients to a maximum l2 norm for each layer and then adding noise to the gradients bounded by the "l2 norm-clipping-bound". Although DPSGD comes with increased computational cost and performance loss, variations of this algorithm [15, 12] still belong to the cutting-edge of PPDL research.

2.3.4 Aggregation

Aggregation methods are generally used along with distributed/collaborated learning, in which multiple parties join a ML task while aiming to keep their respective datasets private [28]. The most popular collaborative framework for privacy preservation is Federated Learning introduced by Google [36]. Although aggregation methods can provide data security during distributed training, their privacy preserving aspects are more limited than other PPDL approaches.

2.3.5 Combined Approaches

The four main categories of PPDL methods are not mutually exclusive. DP is often used in collaborated learning where it is combined with aggregation techniques [62]. A promising framework called Private Aggregation of Teacher Ensembles (PATE) [41], proposes improved privacy preservation with the use of an ensemble of teacher models (which have been trained on non-overlapping datasets), and a differentially private aggregation mechanism. The knowledge of the aggregated model is then transferred into a student model, resulting in a model with strong privacy guarantees.

2.4 Knowledge Distillation

KD refers to the process of transferring the knowledge of an ML model (or an ensemble of models) to a single, smaller model. At its core, it is a model compression technique that has seen a great increase in popularity, since modern DNNs are often too large for practical deployment on edge devices. The earliest version of this technique was developed to compress the knowledge of an ensemble of classifier models to one fast model [5], but as DNNs increased in size Hinton et al. [23] generalized this method for knowledge transfer between neural networks, and gave the name Knowledge Distillation.

A KD system consists of three core components: the knowledge, the distillation algorithm, and a teacher-student architecture. The knowledge in a DNN refers to the learned weights and biases of the network. A teacher is a network which already possesses the knowledge, while the student is the target network of the distillation process. The knowledge of a teacher model can be categorized in to three types: Response-based Knowledge, Feature-based Knowledge and Relation-based Knowledge [21].

In the context of supervised learning, distillation algorithms based on Response-based Knowledge are the most common. These algorithms focus on the output layer of the teacher model. The student model learns to mimic the predictions of the teacher by minimizing a loss that captures the difference between the logits of student and teacher model [23]. Feature-based Knowledge captures the information stored in the intermediate layers of a teacher model. The main idea behind the distillation algorithms utilizing this knowledge is to match the feature activations of the student and teacher. This can improve the training of the student, especially for deep networks with many hidden layers [49]. Algorithms using Relation-based Knowledge utilize the knowledge represented in the relationships between between the feature maps of different layers. These relationships can be modeled in multiple ways such as using the correlation [27], graphs [67], similarity matrices [59], learned feature embeddings [9], or the probabilistic distribution based on feature representations [43].

2.4.1 Knowledge Distillation in NLP

KD is extensively studied in the field of NLP, since modern LMs can be immensely resource consuming. Considering most LMs follow the ”pretrain then fine-tune” deployment pipeline, the distillation process can also differ depending on which phase of the pipeline it is implemented at. The popular DistillBERT model [51] uses general purpose pre-training distillation where the student model remains task-agnostic and can be further fine-tuned. Tang et al. [56] proposed a task-specific distillation method, where they distilled a pre-trained BERT model in to a small LSTM-based classifier. Chatterjee [8] distilled an already fine-tuned BERT model in to a smaller Transformer architecture. The authors of TinyBERT [24] proposed a distillation process which encompasses both phases: a general-purpose distillation is followed by a task-specific one.

2.4.2 Data-Free Knowledge Distillation

While most KD methods use the same dataset for training the teacher model and distilling the student, there have been a rising interest in developing Data-Free KD techniques. Using the original training data for distillation may not be always feasible in many practical cases: sometimes the data is unavailable or is withhold due to privacy or copyright reasons. Methods for Data-Free KD can be divided in to three main categories [31]. Noise Optimization techniques aim to reconstruct realistic training samples by optimizing noise samples until they meet certain constraints [32]. Generative Reconstruction methods utilize GANs to synthesize desired data impressions, that can be used as a replacement for training data [10]. Adversarial Exploration uses an iterative process where concurrently to the distillation, a GAN is optimized to generate samples that yield the greatest discrepancy between the outputs of the teacher and student [38]. While most of the current research on Data-Free KD focus on applications for computer vision, a few methods have been adapted to the NLP domain as well [47, 33].

3 Methods

The experimental setup consist of four main steps with different configurations. The pipeline of the setup is shown in Figure 1.

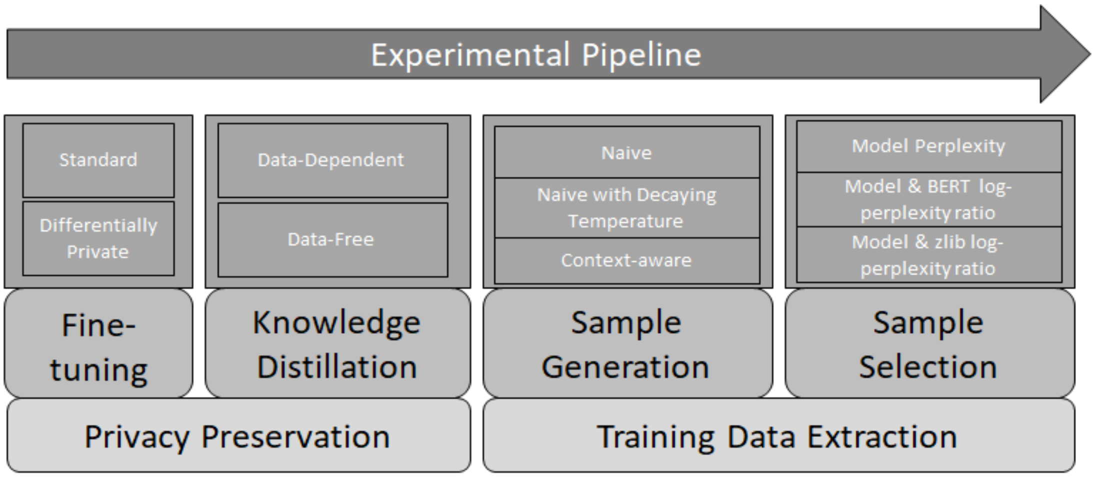

Figure 1: Overview of the pipeline for privacy preservation and training data extraction

Fine-tuning In the first step We take a pre-trained large BERT model with 24 layers from huggingface [63] and fine-tune it on a text-classification task using our own dataset. The fine-tuning is done in two different ways: we use the standard approach for models used for distillation, and the DPSGD algorithm [1] as a direct privacy preserving approach.

Knowledge Distillation In the second part we apply KD on the conventionally fine-tuned models. We use the pre-trained base BERT model with 12 layers from huggingface [63] as the student together with the previously fine-tuned large BERT model as the teacher, and apply a response-based distilliation algorithm [23]. Optionally, we also experiment with a Data-Free KD algorithm [47].

Sample Generation For training data extraction, we follow the best practices found by Carlini et al. [7]. We generate token sequences from the models after detaching the top of the model, using top-n sampling with three different techniques. In the naive approach, we use an empty special token as prompt and sample a sequence of 256 tokens from the results. As a second method, we apply a decaying temperature to the naive sampling to gather more diverse output. Lastly, we gather our own prefixes for prompting, to feed context to our models at the beginning of the token sequence.

Sample Selection From the samples samples we generated in the last step, we select the most likely candidates of unintentionally memorized training data based on the methods described by Carlini et al. [7]. To identify these samples, we we apply some basic preprocessing where we discard duplications, repeated substrings and trivial memorizations and then utilize three metrics to select the best candidates. These metrics include the model perplexity, the ratio of log-perplexity between the pre-trained BERT model and the distilled fine-tuned model, and the ratio between the log perplexity of the model and zlib entropy [19].

4 Research Questions

4.1 Evaluate Training Data Extraction Techniques

The goal of the Training Data Extraction is to find low k-eidetic instances of the training data used for fine-tuning. In the process of fine-tuning, the weights of the encoder part of the models are also updated, therefore it is likely that unintended memorization also happens during fine-tuning. We aim to generate and select samples in a standardized way from our models, and validate our findings by comparing the sample selections to the training data used for fine-tuning.

4.2 Compare Knowledge Distillation with Differentially Private Fine-tuning

The main goal of the thesis is to investigate whether KD could be a valid alternative to DP techniques for privacy preservation. Our hypothesis states that KD achieves comparable privacy preservation as training with DPSGD, while attaining significantly better performance metrics on the downstream task. To test this hypothesis, we evaluate and compare the number of successfully extracted low k-eidetic instances of training data, and the performance metrics of each model.

4.3 Compare Data-Free Knowledge Distillation with Data-Dependent Knowledge Distillation (optional)

Authors of Data-Free KD techniques cite improved privacy preservation as one of the main benefits, but so far there is no study where they have been tested against privacy attacks. As part of the thesis, we aim to compare the performance and privacy preservation of Data-Free KD and Data-Dependent KD.

5 Datasets

For selecting the datasets used for fine-tuning, the following criteria was considered:

- We are interested in datasets which contain information that is low k-eidetic and potentially privacy sensitive. For ethical considerations, we only use publicly available datasets Therefore we count any PII as privacy sensitive information.
- We are only interested in datasets where the raw data is available or at least reconstructable. This is important for the Training Data Extraction evaluation.
- We favor datasets which have been used in privacy related research before.

5.1 Enron Email Dataset

The raw Enron Email corpus [25] consists of 619,446 email messages from 158 employees of the Enron corporation, made public due to legal investigations. The cleaned version has 200,399 messages and is commonly used for various NLP tasks. Since this dataset contains full emails of real users, it includes naturally occurring privacy sensitive information (such as social security numbers, credit card numbers etc), which makes it a perfect fit for our research. We adapt the dataset for text-classification by preprocessing the raw corpus and labeling the emails by folders names shared by the users (i.e. ”sent-mail”, ”corporate” , ”junk”, ”proposals” etc.)

5.2 Blog Authorship Corpus

The Blog Authorship Corpus [52] contains text from blogs written in 2004 and before, with each blog being the work of a single user. The data was collected from blogger.com in 2004. The corpus incorporates a total of 681,288 posts from 19,320 users. Alongside the blogposts, the dataset includes a topic label and demographic information about the writer, including gender, age and zodiac sign. Although the blogposts were written for the public, they contain some PIIs, such as names and postal addresses. We adapt the dataset for text-classification by labeling the posts by the topics.

5.3 Other dataset canditates

- Insurance QA Dataset `https://github.com/shuzi/insuranceQA`
- TAB Dataset `https://github.com/NorskRegnesentral/text-anonymisation-benchmark`

6 Schedule

Task	Time Span
Data Preparation	2022.04.01 - 2022.04.15 (2 weeks)
Implementation of BERT and Downstream Task	2022.04.15 - 2022.04.29 (2 weeks)
Implementation of Training Data Extraction	2022.04.29 - 2022.06.01 (1 month)
Implementation of Knowledge Distillation	2022.06.01 - 2022.06.15 (2 weeks)
Implementation of Differentially Private Fine-tuning	2022.06.15 - 2022.07.01 (2 weeks)
Implementation of Optional Goals	2022.07.01 - 2022.08.01 (1 month)
Evaluation	2022.08.01 - 2022.09.30 (2 months)
Data Collection	2022.05.01 - 2022.09.30 (5 months)
Writing	2022.04.01 - 2022.09.30 (6 months)

Table 1: Planned Schedule

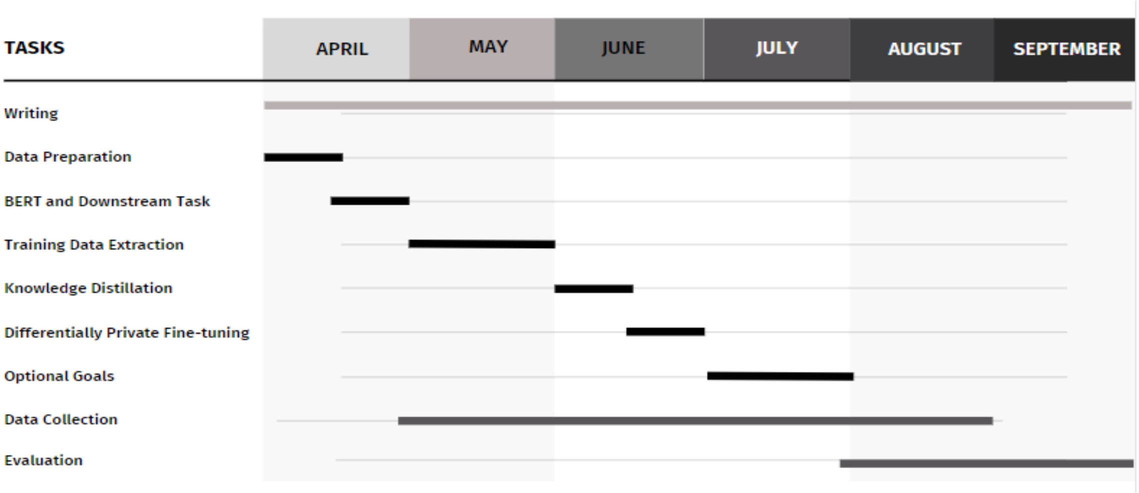

Figure 2: GANNT chart of thesis plan

7 Acronyms

BERT - Bidirectional Encoder Representation for Transformers

DNN - Deep Neural Network

DP - Differential Privacy

GAN - Generative Adversarial Networks

GPT - General Pre-trained Transformer

KD - Knowledge Distillation

LM - Language Model

ML - Machine Learning

MLM - Masked Language Modeling

NLP - Natural Language Processing

PPDL - Privacy Preserving Deep Learning

VAE - Variational Autoencoder

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[2] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–35, 2018.

[3] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[5] C. Buciluă, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

[6] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 267–284.

[7] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[8] D. Chatterjee, "Making neural machine reading comprehension faster," *arXiv preprint arXiv:1904.00796*, 2019.

[9] H. Chen, Y. Wang, C. Xu, C. Xu, and D. Tao, "Learning student networks via feature embedding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 25–35, 2020.

[10] H. Chen, Y. Wang, C. Xu, Z. Yang, C. Liu, B. Shi, C. Xu, C. Xu, and Q. Tian, "Data-free learning of student networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3514–3522.

[11] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[12] A. Davody, D. I. Adelani, T. Kleinbauer, and D. Klakow, "Robust differentially private training of deep neural networks," *arXiv preprint arXiv:2006.10919*, 2020.

[13] E. De Cristofaro, "An overview of privacy in machine learning," *arXiv preprint arXiv:2005.08679*, 2020.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[15] C. Dupuy, R. Arava, R. Gupta, and A. Rumshisky, "An efficient DP-SGD mechanism for large scale NLP models," *arXiv preprint arXiv:2107.14586*, 2021.

[16] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[17] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[18] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 17–32.

[19] J.-l. Gailly and M. Adler, "Zlib compression library," 2004.

[20] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International conference on machine learning.* PMLR, 2016, pp. 201–210.

[21] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[22] F. Hassan, J. Domingo-Ferrer, and J. Soria-Comas, "Anonymization of unstructured data via named-entity recognition," in *International conference on modeling decisions for artificial intelligence.* Springer, 2018, pp. 296–305.

[23] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[24] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[25] B. Klimt and Y. Yang, "Introducing the enron corpus." in *CEAS*, 2004.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[27] S. H. Lee, D. H. Kim, and B. C. Song, "Self-supervised knowledge distillation using singular value decomposition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 335–350.

[28] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–36, 2021.

[29] K. S. Liu, B. Li, and J. Gao, "Generative model: Membership attack, generalization and diversity," *CoRR, abs/1805.09898*, 2018.

[30] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[31] Y. Liu, W. Zhang, J. Wang, and J. Wang, "Data-free knowledge transfer: A survey," *arXiv preprint arXiv:2112.15278*, 2021.

[32] R. G. Lopes, S. Fenu, and T. Starner, "Data-free knowledge distillation for deep neural networks," *arXiv preprint arXiv:1710.07535*, 2017.

[33] X. Ma, Y. Shen, G. Fang, C. Chen, C. Jia, and W. Lu, "Adversarial self-supervised data-free distillation for text classification," *arXiv preprint arXiv:2010.04883*, 2020.

[34] S. Mahloujifar, H. A. Inan, M. Chase, E. Ghosh, and M. Hasegawa, "Membership inference on word embedding and beyond," *arXiv preprint arXiv:2106.11384*, 2021.

[35] H. H. Mao, "A survey on self-supervised pre-training for sequential transfer learning in neural networks," *arXiv preprint arXiv:2007.00800*, 2020.

[36] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, 2016.

[37] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 691–706.

[38] P. Micaelli and A. J. Storkey, "Zero-shot knowledge transfer via adversarial belief matching," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[39] F. Mireshghallah, M. Taram, P. Vepakomma, A. Singh, R. Raskar, and H. Esmaeilzadeh, "Privacy in deep learning: A survey," *arXiv preprint arXiv:2004.12254*, 2020.

[40] S. J. Oh, B. Schiele, and M. Fritz, "Towards reverse-engineering black-box neural networks," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019, pp. 121–144.

[41] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016.

[42] M. P. Parisot, B. Pejo, and D. Spagnuelo, "Property inference attacks on convolutional neural networks: Influence and implications of target model's complexity," *arXiv preprint arXiv:2104.13061*, 2021.

[43] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 268–284.

[44] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[45] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[46] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[47] A. Rashid, V. Lioutas, A. Ghaddar, and M. Rezagholizadeh, "Towards zero-shot knowledge distillation for natural language processing," *arXiv preprint arXiv:2012.15495*, 2020.

[48] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *arXiv preprint arXiv:2007.07646*, 2020.

[49] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[50] B. I. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, “Learning in a large function space: Privacy-preserving mechanisms for SVM learning,” *arXiv preprint arXiv:0911.5708*, 2009.

[51] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” *arXiv preprint arXiv:1910.01108*, 2019.

[52] J. Schler, M. Koppel, S. Argamon, and J. Pennebaker, “Effects of age and gender on blogging in proceedings of 2006 aaai spring symposium on computational approaches for analyzing weblogs,” in *Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*, 2006.

[53] O. Sharir, B. Peleg, and Y. Shoham, “The cost of training NLP models: A concise overview,” *arXiv preprint arXiv:2004.08900*, 2020.

[54] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models,” in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[55] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[56] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin, “Distilling task-specific knowledge from BERT into simple neural networks,” *arXiv preprint arXiv:1903.12136*, 2019.

[57] O. Thakkar, S. Ramaswamy, R. Mathews, and F. Beaufays, “Understanding unintended memorization in federated learning,” *arXiv preprint arXiv:2006.07490*, 2020.

[58] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, “Stealing machine learning models via prediction {APIs},” in *25th USENIX security symposium (USENIX Security 16)*, 2016, pp. 601–618.

[59] F. Tung and G. Mori, “Similarity-preserving knowledge distillation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1365–1374.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[61] B. Wang and N. Z. Gong, “Stealing hyperparameters in machine learning,” in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 36–52.

[62] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[63] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[64] Y. Xu, T. Ma, M. Tang, and W. Tian, “A survey of privacy preserving data publishing using generalization and suppression,” *Applied Mathematics & Information Sciences*, vol. 8, no. 3, p. 1103, 2014.

[65] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *Advances in neural information processing systems*, vol. 32, 2019.

[66] S. Zanella-Béguelin, L. Wutschitz, S. Tople, V. Rühle, A. Paverd, O. Ohrimenko, B. Köpf, and M. Brockschmidt, "Analyzing information leakage of updates to natural language models," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 363–375.

[67] C. Zhang and Y. Peng, "Better and faster: knowledge transfer from multiple self-supervised learning tasks via graph distillation for video classification," *arXiv preprint arXiv:1804.10069*, 2018.

[68] T. Zhang, Z. He, and R. B. Lee, "Privacy-preserving machine learning through data obfuscation," *arXiv preprint arXiv:1807.01860*, 2018.

[69] J. Zhao, Y. Chen, and W. Zhang, "Differential privacy preservation in deep learning: Challenges, opportunities and solutions," *IEEE Access*, vol. 7, pp. 48 901–48 911, 2019.