

Privacy Leakage in Text Classification: A Data Extraction Approach

Adel Elmahdy*

University of Minnesota
adel@umn.edu

Huseyin A. Inan and Robert Sim

Microsoft Research
{huseyin.inan, rsim}@microsoft.com

Abstract

Recent work has demonstrated the successful extraction of training data from generative language models. However, it is not evident whether such extraction is feasible in text classification models since the training objective is to predict the class label as opposed to next-word prediction. This poses an interesting challenge and raises an important question regarding the privacy of training data in text classification settings. Therefore, we study the potential privacy leakage in the text classification domain by investigating the problem of unintended memorization of training data that is not pertinent to the learning task. We propose an algorithm to extract missing tokens of a partial text by exploiting the likelihood of the class label provided by the model. We test the effectiveness of our algorithm by inserting canaries into the training set and attempting to extract tokens in these canaries post-training. In our experiments, we demonstrate that successful extraction is possible to some extent. This can also be used as an auditing strategy to assess any potential unauthorized use of personal data without consent.

1 Introduction

Tremendous progress has recently been made in deep learning with natural language processing (NLP), which has led to significant advances in the model performance of a wide variety of NLP applications. The Transformer model (Vaswani et al., 2017; Wolf et al., 2020) has become the central and dominant architecture of many state-of-the-art NLP models. However, NLP models trained with personal data have also been shown to be vulnerable to fairness (Mehrabi et al., 2021) and privacy

(Mireshghallah et al., 2020) issues, leading to adverse societal and ethical consequences.

One of the prime challenges of training machine learning models is the phenomenon of memorizing unique or rare training data. This may occur via what is called *unintended memorization* (Carlini et al., 2019) where the trained model memorizes out-of-distribution data in the training set that is irrelevant to the learning task. It is known that overfitting is not the cause of such a phenomenon, since the out-of-distribution data can be memorized as long as the model is still learning, making it challenging to mitigate through methods preventing overfitting such as early stopping. This phenomenon raises privacy concerns when the training set includes private data that may be inadvertently leaked, e.g., (Munroe, 2019).

The main focus of our work is to explore the memorization of training data in text classification models, which may contain private information collected from individuals. A motivating example in our study is a topic classification setting in which an individual can have private information, such as “I vote for X party” in the *politics* category, which can lead to a privacy violation if this information is leaked by the model.

We propose a data extraction algorithm to recover missing tokens of a partial text using the target model. The algorithm exploits the likelihood that the model generates for the target label of the text to infer the unknown tokens of the partial input text. To the best of our knowledge, this work is the first to demonstrate privacy leakage in a text classification setting by extracting tokens of canary sequences¹ via access to the underlying classification model. We conduct experiments to evaluate the performance of our extraction algorithm under

*This work was carried out as part of an internship at Microsoft Research (MSR), Redmond, WA. Adel Elmahdy is currently affiliated with the Department of Electrical and Computer Engineering and the Department of Computer Science and Engineering at the University of Minnesota.

¹Canary sequences are out-of-distribution examples inserted into the training data. The trained model is then assessed to measure the degree to which the model has memorized such sequences.

a wide range of parameters such as the number of extracted tokens, the number of canary insertions, and the number of guesses for the extraction.

2 Background: Language Modeling

While this work is about text classification setting, it is built upon language models. In this section, we give a brief overview of language modeling. Language models are one of the pillars of state-of-the-art natural language processing pipelines. It has been well established that training these models at scale on large public corpora makes them adaptable to a wide range of downstream tasks (Bommasani et al., 2021).

Two widely used pre-training objectives are auto-regressive (AR) language modeling (Radford et al., 2018, 2019), and masked language modeling (MLM) (Devlin et al., 2019a; Liu et al., 2019). AR language modeling is based on modeling the probability distribution of a text corpus by decomposing it into conditional probabilities of each token given the previous context. Specifically, the distribution $\mathbb{P}(x_1, x_2, \dots, x_n)$ of a sequence of tokens (x_1, x_2, \dots, x_n) can be factorized as $\mathbb{P}(x_1, x_2, \dots, x_n) = \prod_{i=1}^n \mathbb{P}(x_i | x_1, x_2, \dots, x_{i-1})$ using the Bayes rule. A neural network is then trained to model each conditional distribution. We note that such a decomposition only captures the unidirectional context.

On the other hand, the MLM pre-training objective can utilize the bidirectional context since it is based on replacing a certain portion of tokens by a special symbol [MASK] and the model is trained to recover the original tokens at these corrupted positions. This bidirectional context information often carries useful signal on downstream language understanding tasks such as text classification tasks, leading to improved performance for models trained with MLM pre-training objective.

3 Related Work

The ultimate goal of training language models is to model the underlying distribution of a language, which should not require the memorization of training samples. However, recent results have shown that such memorization occurs in language models (Carlini et al., 2019; Zanella-Béguelin et al., 2020; Carlini et al., 2021; Inan et al., 2021; Miresghallah et al., 2021; Carlini et al., 2022). In fact, when the data distribution is long-tailed, memorization might be necessary to achieve near-optimal accuracy on

the test data (Feldman, 2020; Brown et al., 2021). Leakage of memorized content can cause privacy violations, especially in the case where the content can be linked to an individual (Art. 29 WP, 2014). There is a wide range of data leakage detection and prevention techniques for document classification in the literature, e.g., (Alneyadi et al., 2013; Katz et al., 2014; Alneyadi et al., 2015). However, several challenges and limitations are identified with these techniques (Alneyadi et al., 2016; Cheng et al., 2017).

In the case of language models trained with AR objective, the model learns to predict each and every next token given a sequence of tokens, which can theoretically lead to the leakage of the whole sequence if it is memorized by the model. (Carlini et al., 2021) has shown a successful extraction of memorized data, including various personal information from the GPT-2 model (Radford et al., 2019) belonging to this family.

For language models trained with MLM objective, the story has been different so far. For instance, (Lehman et al., 2021) shows that it is *not* easy to extract sensitive information from the BERT model (Devlin et al., 2019a) trained on private clinical data. This can be attributed to the fact that the MLM objective only targets a small portion of [MASK] tokens randomly replaced in the training set, as opposed to all the tokens in the AR setting.

Other forms of privacy leakage include membership inference, which has been widely explored in vision and text scenarios (Shokri et al., 2017; Yeom et al., 2018; Long et al., 2018; Truex et al., 2018; Song and Shmatikov, 2019; Nasr et al., 2019; Sablayrolles et al., 2019; Hayes et al., 2019; Salem et al., 2019; Leino and Fredrikson, 2020; Choquette-Choo et al., 2021; Shejwalkar et al., 2021), and property inference (Ganju et al., 2018; Zhang et al., 2021; Mahloujifar et al., 2022).

4 This Work: Text Classification

In this work, we turn our attention to the text classification setting, which spans a wide range of downstream applications (Minaee et al., 2021). Often times pre-training a language model is performed on large public datasets while fine-tuning requires a much smaller task-specific dataset whose privacy requirements might be much more strict. To the best of our knowledge, this setting has been largely unexplored and our goal is to understand potential privacy leakage in this setting.

In a text classification problem, the input is a sequence of tokens $\mathbf{x} = (x_1, x_2, \dots, x_n)$ with a corresponding class label $y \in \{1, 2, \dots, C\}$ where C is the number of classes. A model is trained to learn the relation between the input text and the corresponding class label. From a training data extraction perspective, the challenge of this setting is that here the goal is to maximize the log-likelihood of the correct class label (i.e. $\log \mathbb{P}(y|\mathbf{x})$), therefore, there is no language modeling involved among the tokens of the sequence \mathbf{x} . Although we cannot leverage the approaches introduced in prior work, it is also not clear a priori whether one can extract training data given the partial knowledge of the tokens and the label with query access to the model.

5 Threat Model and Testing Methodology

Similar to prior work (Shokri et al., 2017; Carlini et al., 2019), we assume black-box access to the target model, where it receives a sequence of tokens and outputs a class prediction with its corresponding likelihood. Our goal is to investigate whether it is possible to extract the remaining tokens given partial information about a sequence under this black-box access to the target model.

This framework encompasses both a malicious attacker who has partial information about personal data points and aims to fully reconstruct it by fiddling with the target model, and any individual who audits a target model to detect any unauthorized use of personal data (Song and Shmatikov, 2019) (or to check whether a model owner has actually complied with data deletion requests). We choose to focus on the latter case since it allows the data owner to inject “special” sequences into their data that would strongly indicate unauthorized use of personal data if a successful reconstruction is possible through the target model.

Similar to (Thakkar et al., 2021), we inject sequences of randomly selected tokens (with corresponding labels) into the training set. This mimics the existence of out-of-distribution data that is not pertinent to the learning process. We consider a testing procedure in which the goal of the extraction algorithm is to retrieve the last n tokens of a canary², where the sample space for each missing token is the entire tokenizer vocabulary. In the next section, we propose our extraction algorithm.

²Since the model is bidirectional, this could be any arbitrary n tokens in the sequence in general.

6 Proposed Extraction Algorithm

Given a partial sequence with missing tokens, the core idea of the proposed extraction algorithm is to choose the tokens such that the corresponding class label achieves the highest likelihood under the target model. Consider a canary sequence $\mathbf{x} = (x_1, x_2, \dots, x_n)$ with a corresponding label y . Given a partial input, we iteratively query the underlying classification model to reconstruct the missing tokens. In particular, for a partial sequence $(x_1, x_2, \dots, x_{t-1})$, the extraction algorithm enumerates all possible tokens from the vocabulary \mathcal{V} , evaluates the corresponding likelihood of the label y for each token by querying the classification model, and then returns the token that achieves the maximum likelihood. Formally, x_t is evaluated using the following optimization problem:

$$x_t = \arg \max_{v \in \mathcal{V}} \mathbb{P}(y|(x_1, x_2, \dots, x_{t-1}, v)). \quad (1)$$

When a canary is repeated a few times in the training set, the extraction criterion in (1) may not yield a successful reconstruction of the canary sequence. In order to boost the performance of token extraction, we propose a data-dependent regularizer to penalize the tokens with the highest number of occurrences in the training set, counteracting the model’s bias towards these tokens. Let $C(v)$ be the normalized number of occurrences of token v in the training data³ for $v \in \mathcal{V}$. Consequently, the optimization problem with the regularized objective function is given by

$$x_t = \arg \max_{v \in \mathcal{V}} \mathbb{P}(y|(x_1, x_2, \dots, x_{t-1}, v)) - \lambda \cdot C(v),$$

where λ is the regularization coefficient that controls the amount of penalization imposed on the tokens with frequent occurrences in the training data.

7 Experimental Evaluation

Dataset: We use the Reddit dataset⁴. We select the top 100 subreddits with largest number of reddit posts. We randomly sample 10000 and 2500 posts for the training and validation sets, respectively. The task is topic classification. In particular, given a user comment, the model is trained to predict the corresponding subreddit.

³This may be a strong requirement but approximations can be made via publicly available datasets. However, the extraction performance does not degrade much by setting $\lambda = 0$ (see Table 2 in Section 7).

⁴<https://huggingface.co/datasets/reddit>

	Original Canary		Supporting Canary	
	Subreddit	Repetitions	Subreddits	Repetitions
Table 2	Rarest	100	All Other Subreddits	1
Table 3	Rarest	Varying (1st Column)	All Other Subreddits	1
Table 4	Rarest	100	One Other Subreddit	Varying (1st Column)

Table 1: Information about the subreddits as well as numbers of repetitions of the original and supporting canaries for each experiment.

Model: We use the pre-trained BERT base model (Devlin et al., 2019b). We fine-tune the model for 10 epochs using AdamW optimizer (Loshchilov and Hutter, 2018) with weight decay 0.01, learning rate 1e-6, and batch size 32. We apply early stopping and take the snapshot that achieves the best validation performance to avoid overfitting. The average performance of the model over 10 runs with different random seeds is as follows:

- The average training accuracy is 47.69% for a training set size of 10k samples.
- The average validation accuracy is 42.94% for a validation set size of 2.5k samples.

Canary Construction: A canary sequence consists of a number of tokens and an associated class label. Each token in a canary is sampled uniformly at random from the BERT tokenizer vocabulary. We exclude subwords and sample from the remaining 17k whole words in the vocabulary. The reason for random sampling of tokens is to construct out-of-distribution posts with very high probability. For instance, an example of a randomly generated canary is “expected Disney activated Fulton rebel scalp Stark fraud myths Palestine.” Finally, a canary sequence is inserted into the training set and repeated multiple times. This construction of canary sequences enables us to evaluate the model’s unintended memorization of training data.

Intuitively, the most successful extraction is likely to occur within the rarest subreddit because there is more capacity for memorization. Hence, we insert a canary sequence of 10 randomly selected tokens into the rarest subreddit with 100 repetitions. This will be the original canary for which we would like to perform token extractions. Our first observation is that given the first 7 to 9

λ	Success Rate		
	Last Token	Last 2 Tokens	Last 3 Tokens
0	0.8	0.2	0
0.01	0.9	0.3	0
0.1	0.7	0.1	0
1	0.3	0.1	0
10	0.1	0	0

Table 2: Successful extraction rates of the proposed algorithm on the last 1 to 3 tokens for different values of the regularization parameter λ . The original canary is inserted 100 times in the rarest subreddit, while the supporting canary is inserted only once in all other subreddits. Random guess rate is only 0.0058 for the last token and 3.4e-5 for last 2 tokens.

tokens, the model is already confident in the corresponding label, and hence the missing token(s) do not exhibit themselves in our optimization. In particular, $\mathbb{P}(y|(x_1, x_2, \dots, x_9, v))$ has similar values for all $v \in \mathcal{V}$. Therefore, we inject one sequence into all other subreddits where the first 7 to 9 tokens are fixed, and the missing token(s) are chosen differently at random. These are called *supporting canaries* since they are not meant to be extracted, but enable the missing token(s) in the original canary to be crucial for maximizing the likelihood of the corresponding label, and hence the performance of the reconstruction is significantly boosted. Table 1 shows detailed information about the original and supporting canaries for each experiment whose results are presented next. The success of reconstruction is defined by the appearance of the missing token(s) in the top- k generation of the algorithm for a beam size k . Note that each experiment is run 10 times and the average success rate is reported. In Table 2, we present the results of the aforementioned experiment with $k = 100$. It is evident that the proposed algorithm achieves significant success rates for the extraction of a few tokens. However, it fails to reconstruct beyond more than two tokens since the search space becomes exponentially larger.

Table 3 presents the extraction results for the last token for various repetitions of the original canary and beam sizes. The supporting canary is inserted only once in all subreddits except the rarest. Although high repetition improves the success rate of our algorithm, which aligns well with the findings that memorization is exacerbated by duplication of a sequence (Kandpal et al., 2022; Carlini et al., 2022), low repetition still resurfaces the missing

Original Canary		Success Rate	
Repetitions	Beam Size	Our Algo.	Random Guess
100	50	0.7	0.0029
50	50	0.5	0.0029
25	50	0.1	0.0029
10	50	0	0.0029
100	100	0.9	0.0058
50	100	0.5	0.0058
25	100	0.3	0.0058
10	100	0.1	0.0058
100	200	1	0.0117
50	200	0.9	0.0117
25	200	0.4	0.0117
10	200	0.2	0.0117

Table 3: Successful extraction rates of the proposed algorithm compared to random guessing on the last token for various repetitions of the original canary and beam sizes. The supporting canary is inserted only once in all subreddits except the rarest. We set $\lambda = 0.01$.

token if the algorithm generates a larger number of candidates (i.e., larger beam size).

Instead of inserting one supporting canary into all subreddits except the rarest, we next investigate the insertion of a supporting canary into only one other arbitrarily chosen subreddit. Here we fix 100 repetitions of the original canary in the rarest subreddit and vary the repetition of the supporting canary in a different subreddit. Table 4 shows the extraction results for this experiment for various repetitions of the supporting canary and beam sizes. We can see that extraction is possible even when a canary is inserted into the rarest subreddit only, as shown in the last part of Table 4. However, the success rate improves greatly when we inject a supporting canary into another subreddit. The repetition we use for the subreddit does not seem to have an effect on the success rate of the extraction of the original canary.

8 Conclusion and Future Work

In this work, we studied the problem of unintentional memorization in a text classification setting. We developed an algorithm to extract unknown tokens of a partial text via access to the underlying classification model. Through experimental studies, we demonstrated the efficacy of the proposed extraction algorithm over random guessing.

Our experimental setting provides preliminary results and is subject to further exploration in future

Supporting Canary		Success Rate	
Repetitions	Beam Size	Our Algo.	Random Guess
99	50	0.5	0.0029
99	100	0.5	0.0058
99	200	0.5	0.0117
50	50	0.4	0.0029
50	100	0.4	0.0058
50	200	0.5	0.0117
25	50	0.4	0.0029
25	100	0.5	0.0058
25	200	0.5	0.0117
0	50	0.1	0.0029
0	100	0.1	0.0058
0	200	0.1	0.0117

Table 4: Success rates of extracting the last token under the proposed algorithm and random guess for various repetitions of the supporting canary and beam sizes. The original canary is inserted 100 times in the rarest subreddit. We set $\lambda = 0$.

work. In particular, we injected the original canary into the rarest subreddit. In general, it would be interesting to range from the rarest to the most popular subreddit. We also used random tokens for canary construction, and it is of importance to extend it to more organic canaries. Finally, we leave investigating the effect of formal privacy guarantees, such as differentially private model training (Abadi et al., 2016), to future work.

9 Ethical Impact

This work explores the privacy implications of a text classification setting in which training is performed on sensitive and private data. We investigate whether data leakage is feasible under this setting. We believe that this work is a first step in determining the susceptibility of the underlying text classification model to privacy leakage and detecting unauthorized use of personal data. Both the dataset and the model are publicly available.

Acknowledgements

We thank members of Microsoft’s Privacy in AI (PAI) research group for the valuable feedback in the early stages of this work. Special thanks to Fatemehsadat Mireshghallah for the insightful discussions, and Mashaal Musleh for the support in setting up the experiment environment. Finally, we thank the anonymous reviewers for their helpful comments toward improving the paper.

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.
- Sultan Alneyadi, Elankayer Sithirasenan, and Vallipuram Muthukkumarasamy. 2013. Adaptable N-gram classification model for data leakage prevention. In *7th International Conference on Signal Processing and Communication Systems (ICSPCS)*.
- Sultan Alneyadi, Elankayer Sithirasenan, and Vallipuram Muthukkumarasamy. 2015. Detecting data semantic: a data leakage prevention approach. In *IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 910–917.
- Sultan Alneyadi, Elankayer Sithirasenan, and Vallipuram Muthukkumarasamy. 2016. A survey on data leakage prevention systems. *Journal of Network and Computer Applications*, 62:137–152.
- Art. 29 WP. 2014. [Opinion 05/2014 on “Anonymisation Techniques”](#).
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. 2021. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 123–132.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arxiv.2202.07646*.
- Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.
- Long Cheng, Fang Liu, and Danfeng Yao. 2017. Enterprise data breach: causes, challenges, prevention, and future directions. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(5):e1211.
- Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *International Conference on Machine Learning*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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)*, NAACL-HLT ’19, pages 4171–4186.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.
- 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*, STOC 2020, page 954–959.
- Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS ’18, page 619–633. Association for Computing Machinery.
- Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2019. LOGAN: Membership inference attacks against generative models. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*, volume 2019, pages 133–152.
- Huseyin A Inan, Osman Ramadan, Lukas Wutschitz, Daniel Jones, Victor Rühle, James Withers, and Robert Sim. 2021. Training data leakage analysis in language models. *arXiv preprint arXiv:2101.05405*.
- Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. *arXiv preprint arXiv:2202.06539*.
- Gilad Katz, Yuval Elovici, and Bracha Shapira. 2014. Coban: A context based model for data leakage prevention. *Information sciences*, 262:137–158.
- Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron C Wallace. 2021. Does bert pre-trained 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.

- Klas Leino and Matt Fredrikson. 2020. Stolen Memories: Leveraging model memorization for calibrated white-box membership inference. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1605–1622.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- 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*.
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.
- Saeed Mahloujifar, Esha Ghosh, and Melissa Chase. 2022. Property inference from poisoning. In *IEEE Symposium on Security and Privacy (SP)*, pages 1569–1586.
- Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.
- Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.
- Fatemehsadat Mireshghallah, Huseyin Inan, Marcello Hasegawa, Victor Rühle, Taylor Berg-Kirkpatrick, and Robert Sim. 2021. Privacy regularization: Joint privacy-utility optimization in languagemodels. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3799–3807.
- Fatemehsadat Mireshghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, and Hadi Esmailzadeh. 2020. Privacy in deep learning: A survey. *arXiv preprint arXiv:2004.12254*.
- Randall Munroe. 2019. xkcd: Predictive models. <https://xkcd.com/2169>.
- 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 *IEEE Symposium on Security and Privacy (SP)*, pages 739–753.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5558–5567.
- Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Network and Distributed Systems Security Symposium*.
- Virat Shejwalkar, Huseyin A Inan, Amir Houmansadr, and Robert Sim. 2021. Membership inference attacks against NLP classification models. In *NeurIPS 2021 Workshop Privacy in Machine Learning*.
- Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, pages 3–18.
- Congzheng Song and Vitaly Shmatikov. 2019. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206.
- Om Dipakbhai Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Francoise Beaufays. 2021. Understanding unintended memorization in language models under federated learning. In *Third Workshop on Privacy in Natural Language Processing*. Association for Computational Linguistics.
- Stacey Truex, Ling Liu, Mehmet Emre Guroy, Lei Yu, and Wenqi Wei. 2018. Towards demystifying membership inference attacks. *arXiv preprint arXiv:1807.09173*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 38–45.
- Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282.

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*, page 363–375.

Wanrong Zhang, Shruti Tople, and Olga Ohrimenko. 2021. Leakage of dataset properties in Multi-Party machine learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2687–2704.