Extracted BERT Model Leaks More Information than You Think!

Xuanli He[1]*, **Chen Chen**[2]*, **Lingjuan Lyu**[3]†, **Qiongkai Xu**[4]
[1]University College London, [2]Zhejiang University, [3]Sony AI, [4] The University of Melbourne
zodiac.he@gmail.com, cc33@zju.edu.cn
Lingjuan.Lv@sony.com, Qiongkai.Xu@unimelb.edu.au

Abstract

The collection and availability of big data, combined with advances in pre-trained models (e.g. BERT), have revolutionized the predictive performance of natural language processing tasks. This allows corporations to provide machine learning as a service (MLaaS) by encapsulating fine-tuned BERT-based models as APIs. Due to significant commercial interest, there has been a surge of attempts to steal remote services via model extraction. Although previous works have made progress in defending against model extraction attacks, there has been little discussion on their performance in preventing privacy leakage. This work bridges this gap by launching an attribute inference attack against the extracted BERT model. Our extensive experiments reveal that model extraction can cause severe privacy leakage even when victim models are facilitated with advanced defensive strategies.

1 Introduction

The emergence of pre-trained language models (PLMs) has revolutionized the natural language processing (NLP) research, leading to state-of-the-art (SOTA) performance on a wide range of tasks (Devlin et al., 2018; Yang et al., 2019). This breakthrough has enabled commercial companies to deploy machine learning models as black-box APIs on their cloud platforms to serve millions of users, such as Google Prediction API[1], Microsoft Azure Machine Learning[2], and Amazon Machine Learning[3].

However, recent works have shown that existing NLP APIs are vulnerable to model extraction attack (MEA), which can reconstruct a copy of the remote NLP model based on the carefully-designed queries and outputs of the target API (Krishna et al., 2019; Wallace et al., 2020), causing the financial losses of the target API. Prior to our work, researchers have investigated the hazards of model extraction under various settings, including stealing commercial APIs (Wallace et al., 2020; Xu et al., 2022), ensemble model extraction (Xu et al., 2022), and adversarial examples transfer (Wallace et al., 2020; He et al., 2021).

Previous works have indicated that an adversary can leverage the extracted model to conduct adversarial example transfer, such that these examples can corrupt the predictions of the victim model (Wallace et al., 2020; He et al., 2021). Given the success of MEA and adversarial example transfer, we conjecture that the predictions from a victim model could reveal its private information unconsciously, as victim models can memorize side information in addition to the task-related message (Lyu and Chen, 2020; Lyu et al., 2020; Carlini et al., 2021). Thus, we are interested in examining whether the victim model can leak the private information of its data to the extracted model, which has received little attention in previous research. In addition, a list of defenses against MEA has been devised (Lee et al., 2019; Ma et al., 2021; Xu et al., 2022; He et al., 2022a,b). Although these technologies can alleviate the effects of MEA, it is unknown whether such defenses can prevent private information leakage, *e.g.,* gender, age, identity.

To study the privacy leakage from MEA, we first leverage MEA to obtain a white-box extracted model. Then, we demonstrate that from the extracted model, it is possible to infer sensitive attributes of the data used by the victim model. To the best of our knowledge, this is the first attempt that investigates privacy leakage from the extracted model. Moreover, we demonstrate that the privacy leakage is resilient to advanced defense strategies even though the task utility of the extracted

*Equal contribution. Most of the work was finished when X.H. was at Monash University. Work done during C.C.'s internship at Sony AI.

†Corresponding author.

[1]https://cloud.google.com/prediction

[2]https://studio.azureml.net

[3]https://aws.amazon.com/machine-learning

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1530–1537
December 7-11, 2022

model is significantly diminished, which could motivate further investigation on defense technology in MEA.[4]

2 Related Work

MEA aims to steal an intellectual model from cloud services (Tramèr et al., 2016; Orekondy et al., 2019; Krishna et al., 2019; Wallace et al., 2020). It has been studied both empirically and theoretically, on simple classification tasks (Tramèr et al., 2016), vision tasks (Orekondy et al., 2019), and NLP tasks (Krishna et al., 2019; Wallace et al., 2020). MEA targets at imitating the functionality of a black-box victim model (Krishna et al., 2019; Orekondy et al., 2019), *i.e.,* a model replicating the performance of the victim model.

Furthermore, the extracted model could be used as a reconnaissance step to facilitate later attacks (Krishna et al., 2019). For instance, the adversary could construct transferrable adversarial examples over the extracted model to corrupt the predictions of the victim model (Wallace et al., 2020; He et al., 2021). Prior works (Coavoux et al., 2018; Lyu et al., 2020) have shown malicious users can infer confidential attributes based on the interaction with a trained model. However, to the best of our knowledge, none of the previous works investigate whether the extracted model can facilitate privacy leakage of the data used by the black-box victim model.

In conjunction with MEA, a list of avenues has been proposed to defend against MEA. These approaches focus on the perturbation of the posterior prediction. Orekondy et al. (2019) suggested revealing the top-K posterior probabilities only. Lee et al. (2019) demonstrated that API owners could increase the difficulty of MEA by softening the posterior probabilities and imposing a random noise on the non-argmax probabilities. Ma et al. (2021) introduced an adversarial training process to discourage the knowledge distillation from the victim model to the extracted model. However, these approaches are specific to model extraction, which are not effective to defend against attribute inference attack, as shown in Section 5.

3 Attacking BERT-based API

We first describe the process of MEA. Then we detail the proposed attack: *attribute inference attack*

[4]Code and data are available at: https://github.com/xlhex/emnlp2022_aia.git

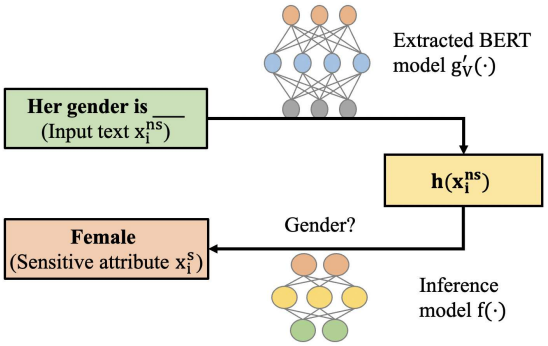


Figure 1: The workflow of attribute inference attack against an extracted BERT model. We use an auxiliary attribute inference model to infer the demographic information of a text.

(AIA). Throughout this paper, we mainly focus on the BERT-based API as the victim model, which is widely used in commercial black-box APIs.

Model Extraction Attack (MEA). To conduct MEA, attackers craft a set of inputs as queries (transfer set), and send them to the target victim model (BERT-based API) to obtain the predicted posterior probability, *i.e.,* the outputs of the softmax layer. Then attackers can reconstruct a copy of the victim model as an "extracted model" by training on query-prediction pairs.

Attribute Inference Attack (AIA). After we derive an extracted model, we now investigate how to infer sensitive information from the extracted model by conducting AIA against the extracted model. Given any record $\boldsymbol{x} = [x^{ns}, x^s]$, AIA aims to reconstruct the sensitive components x^s, based on the hidden representation of x^{ns}, where x^{ns} and x^s represent the non-sensitive information and the target sensitive attribute respectively. The intuition behind AIA is that the representation generated by the extracted model can be used to facilitate the inference of the sensitive information of the data used by the victim model (Coavoux et al., 2018). Note that the **only** explicit information that is accessible to the attacker is the predictions output by the victim model, rather than the raw BERT representations.

Given an extracted model g'_V, we first feed a limited amount of the auxiliary data D_{aux} with labelled attribute into g'_V to collect the BERT representation $\boldsymbol{h}(x_i^{ns})$ for each $x_i \in D_{aux}$. Then, we train an inference model $f(\cdot)$, which takes the BERT representation of the extracted model as input and outputs the sensitive attribute of the input, i.e., $\{\boldsymbol{h}(x_i^{ns}), x_i^s\}$. The trained inference model

	AG news	Blog	TP-US
Victim model	79.99	97.07	85.53
$\mathcal{T}_A = \mathcal{T}_V$ (D_Q)	**80.10**	**95.64**	**86.53**
$\mathcal{T}_A \neq \mathcal{T}_V$ (reviews)			
0.1x	50.90	36.83	79.95
1x	69.94	88.16	85.15
5x	75.29	92.75	85.82
$\mathcal{T}_A \neq \mathcal{T}_V$ (news)			
0.1x	61.70	18.04	79.20
1x	71.95	83.13	84.15
5x	75.82	87.64	85.46

Table 1: Performance of MEA across different domains and query sizes on the test set, compared to the victim models. The evaluation metric is accuracy.

$f(\cdot)$ can infer the sensitive attribute; in our case, they are gender, age and named entities (see Section 4.1).

During test time, as illustrated in Figure 1, the attacker can first derive the BERT representation of any record by using the extracted model, then feed the extracted BERT representation into the trained inference model $f(\cdot)$ to infer the sensitive attributes.

4 Experiments and Analysis

4.1 Experimental Setup

Data. We conduct experiments on three datasets: *i)* Trustpilot (TP) (Hovy et al., 2015), *ii)* AG news (Del Corso et al., 2005), and *iii)* Blog posts (Blog) (Schler et al., 2006). To study AIA, we reuse the data pre-processed by Coavoux et al. (2018). For TP, Coavoux et al. (2018) use the subset from US users, *i.e.,* TP-US. The private attributes of TP-US and Blog are *gender* and *age*. The private attributes of AG news are the five most frequent person entities. More details and statistics are provided in Appendix A.

Settings. For each dataset, we randomly split the training data D into two halves D_V and D_Q, where $|D_V| = |D_Q|$. The first half (D_V) is used to train the victim model, whereas the second half (D_Q) is reserved for two folds. The first fold is to train an extracted model, where the data distribution of the victim's training data ($\mathcal{T}_V$) is the same as that of the query ($\mathcal{T}_A$). The second fold is to train $f(\cdot)$ to infer the private attributes, *i.e.,* D_{aux}.

Since API providers tend to use in-house datasets, it is difficult for the attacker to know the target data distribution as prior knowledge. Thus, we sample queries from different distributions but semantically-coherent data as the original data ($\mathcal{T}_A \neq \mathcal{T}_V$). Specifically, we use Amazon reviews dataset (Zhang et al., 2015) (reviews) and CNN/DailyMail dataset (Hermann et al., 2015) (news) as cross-domain queries. Empirically, each query incurs a certain expense. Due to the budget limit, attackers cannot issue massive requests. For the cross-domain case, we vary query size from {0.1x,1x,5x} size of D_V.

	AG news	Blog	TP-US
Majority class	49.94	49.57	38.15
BERT (w/o fine-tuning)	69.39	44.03	49.38
$\mathcal{T}_A = \mathcal{T}_V$ (D_{aux})	15.68	34.41	36.19
$\mathcal{T}_A \neq \mathcal{T}_V$ (reviews)			
0.1x	20.63	35.03	**35.04**
1x	17.93	34.34	35.97
5x	18.31	34.45	36.82
$\mathcal{T}_A \neq \mathcal{T}_V$ (news)			
0.1x	**13.95**	35.60	35.38
1x	15.76	**33.88**	36.92
5x	17.91	35.39	37.68

Table 2: Empirical privacy of baselines and under AIA attack over different datasets and settings. The extracted model is trained on the queries from different distributions. Note higher value means better empirical privacy.

In order to test AIA, we assume D_V is accessible to attackers. We use the non-sensitive attributes of D_V as the test input. If the attacker can successfully infer the sensitive attributes of D_V given its non-sensitive information, then it will cause a privacy leakage of the victim model. Following Coavoux et al. (2018), for demographic variables (*i.e.,* gender and age), we take $1 - X$ as *empirical privacy*, where X is the average prediction accuracy of the attack models on these two variables; for named entities, we take $1 - F$ as *empirical privacy*, where F is the F1 score between the ground truths and the prediction by the attackers on the presence of all the named entities. Higher empirical privacy means lower attack performance.

Training Details. Victim and extracted models are *BERT-base* (Devlin et al., 2018), trained for 5 epochs with the Adam optimizer (Kingma and Ba, 2014) using a learning rate of 2×10^{-5}. We use the codebase from Transformers library (Wolf et al., 2020). Attribute inference models are 2-layer MLP, trained for 3 epochs with the same optimizer and learning rate. All experiments are run with one Nvidia V100 gpu.

		AG news			BLOG			TP-US		
		Utility ↑	MEA ↓	AIA ↑	Utility ↑	MEA ↓	AIA ↑	Utility ↑	MEA ↓	AIA ↑
No Defense		79.99	71.95	15.76	**97.07**	88.16	34.34	85.53	85.33	36.92
SOFT.	$\tau = 0.0$	79.99	69.11	**22.47**	97.07	85.57	**35.19**	85.53	84.60	37.62
	$\tau = 0.5$	79.99	72.32	20.78	97.07	85.68	34.91	85.53	85.10	**37.69**
	$\tau = 5$	79.99	72.48	11.32	97.07	86.73	33.80	85.53	85.33	33.18
PERT.	$\sigma = 0.05$	**80.03**	71.47	14.46	96.17	85.87	34.75	**85.83**	85.09	37.43
	$\sigma = 0.2$	79.41	71.61	12.58	95.38	85.31	34.97	85.65	84.98	36.90
	$\sigma = 0.5$	65.13	69.05	11.66	62.23	81.77	33.79	63.21	83.88	35.90
Reverse Sigmoid		79.99	71.59	12.17	97.07	85.08	33.09	85.53	85.34	32.81
NASTY		79.90	71.33	17.00	96.05	85.61	34.24	85.15	84.40	36.77
MOSTLEAST		79.99	**47.98**	17.86	97.07	**48.29**	34.44	85.53	**39.40**	37.60

Table 3: Attack performance under different defenses on AG News, BLOG and TP-US. τ is temperature parameter on softmax. σ is the variance of Gaussian noise. Utility means the accuracy of the victim model after adopting defense. For MEA, **lower** scores indicate better defenses. conversely for AIA. All experiments are conducted on datasets with 1x queries.

Baselines. To gauge the private information leakage, we consider a majority value for each discrete attribute as a baseline. To evaluate how the extracted model suffers from AIA, we also take the pretrained BERT without (w/o) fine-tuning as a baseline model to extract representation. Note that BERT (w/o fine-tuning) is a plain model that does not contain any information about the training data of the target model.

4.2 Experimental Results

MEA results. We present the performance of MEA for the same domain querying and cross-domain querying in Table 1. Due to the domain mismatch, the cross-domain querying underperforms the same-domain querying. Although increasing the cross-domain query size can boost the accuracy of the extracted model, it is still inferior to the same-domain competitor with fewer data. In addition, we notice that AG news prefers *news* data, while TP-US and Blog favor *reviews* data. Intuitively, one can attribute this preference to the genre similarity, *i.e., news* data is close to AG news, while distant from TP-US and Blog. To verify this phenomenon, we calculate the uni-gram and 5-gram overlapping between test sets and different queries in Appendix A.

Since we do not have access to the training data of the victim model, we will use *news* data as queries for AG news, and *reviews* data as queries for TP-US and Blog, unless otherwise mentioned.

AIA results. We show AIA results using the same-domain and cross-domain queries in Table 2. Table 2 shows that compared to the BERT (w/o fine-tuning) and majority baselines, the attack model built on the BERT representation of the extracted model indeed essentially enhances the attribute inference for the victim training data, *i.e.,* more than 3.57-4.97x effective for AG news compared with the baselines, even when using cross-domain queries. The majority baseline is merely a random guess, while BERT (w/o fine-tuning) is a plain model that **did not** contain any information about the victim training data. However, the extracted model is trained on the queries and the returned predictions from the victim model. This implies that the target model predictions inadvertently capture sensitive information about users, such as their gender, age, and other important attributes, apart from the useful information for the main task.

Interestingly, compared with the queries from the same distribution, Table 2 also shows that queries from different distributions make AIA **easier** (see the best results corresponding to the lower privacy in bold in Table 2). We provide a detailed study of this phenomenon in Appendix B.1.

5 Defense

Although we primarily focus on the privacy vulnerability of BERT-based APIs in this work, we also test four representative defenses: *i)*: **Softening predictions:** Using τ on softmax layer to scale probability vector (Xu et al., 2022). *ii)*: **Prediction perturbation:** Adding Gaussian noises with a variance of σ to the probability vector (Xu et al., 2022). *iii)*. **Reverse sigmoid:** Softening the posterior probabilities and injecting a random noise on the non-argmax probabilities (Lee et al., 2019). *iv)*.

Nasty teacher: Using an adversarial loss to discourage the knowledge distillation from the victim model to the extracted model (Ma et al., 2021). We also propose a new defense called **Most Least**, in which we set the predicted probabilities of the most and least likely categories to $0.5+\epsilon$ and $0.5-\epsilon$, and zero out others. ϵ could be set as small as possible. For defense experiment, we set ϵ to 10^{-5}.

According to Table 3, except MOSTLEAST, none of the defense avenues can well defend against MEA, unless we significantly compromise the utility (or accuracy) of the victim model. However, such degradation is more detrimental to the victim model than the extracted model. Consequently, the extracted model may surpass the victim model.

Regarding AIA, although MOSTLEAST manages to defend against MEA, it still falls short of preventing privacy leakage from the extracted model (*c.f.,* Table 2 and 3). Among these defenses, merely the hard-labeling ($\tau = 0.0$) can slightly mitigate the information leakage caused by AIA. In addition, some defenses, such as prediction perturbation and reverse sigmoid, can even exacerbate privacy leakage. Given that these methods have been used to defend against MEA, such a side effect requires more investigation before it causes a severe implication. We leave this for future study.

6 Conclusions

This work reveals that the hazards of the extracted model have been underestimated. In addition to the violation of IP, it can vastly exacerbate the privacy leakage even under challenging scenarios (*e.g.,* limited query budget; queries from distributions that are different from that of the training data used by the victim APIs). Such a vulnerability cannot be alleviated by the strong defensive strategies against model extraction. We hope our work can raise the alarm for more investigations to the vulnerability of model extraction attack.

Limitations

Although our work has revealed the vulnerability of model extraction through a lens of privacy leakage, we have not proposed an effective enough defense approach for AIA. Thus, we encourage the community to investigate this direction to mitigate the adverse social impacts caused by this attack.

Statement of Ethics

There are two major ethical issues in this work. The first one is the violation of intellectual property, as model extraction attacks can illegally replicate commercial APIs. The second relates to privacy leakage in model extraction attacks. Both can bring severe ethical concerns to the community when deploying machine learning services on the cloud platform. Although we have shown that some defensive avenues can partly mitigate their vulnerabilities, more efforts should be dedicated to them in future work.

References

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. 2018. Privacy-preserving neural representations of text. In *EMNLP*, pages 1–10.

Gianna M Del Corso, Antonio Gulli, and Francesco Romani. 2005. Ranking a stream of news. In *WWW*, pages 97–106.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xuanli He, Lingjuan Lyu, Qiongkai Xu, and Lichao Sun. 2021. Model extraction and adversarial transferability, your bert is vulnerable! In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2006–2012.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. 2022a. Protecting intellectual property of language generation apis with lexical watermark. In *Proceedings of the 36th AAAI Conference on Artificial Intelligenc*.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. 2022b. Cater: Intellectual property protection on text generation apis via conditional watermarks. In *NeurIPS*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NeurIPS*, pages 1693–1701.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *WWW*, pages 452–461.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *arXiv*, pages arXiv–1907.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2019. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*.

Taesung Lee, Benjamin Edwards, Ian Molloy, and Dong Su. 2019. Defending against neural network model stealing attacks using deceptive perturbations. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 43–49. IEEE.

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.

Lingjuan Lyu and Chi-Hua Chen. 2020. Differentially private knowledge distillation for mobile analytics. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1809–1812.

Lingjuan Lyu, Xuanli He, and Yitong Li. 2020. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. In *EMNLP Findings*.

Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. 2021. Undistillable: Making a nasty teacher that {cannot} teach students. In *International Conference on Learning Representations*.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *CVPR*, pages 4954–4963.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In *USENIX*, pages 601–618.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. *arXiv preprint arXiv:2004.15015*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Qiongkai Xu, Xuanli He, Lingjuan Lyu, Lizhen Qu, and Gholamreza Haffari. 2022. Student surpasses teacher: Imitation attack for black-box NLP APIs. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2849–2860, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5753–5763.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NeurIPS*, pages 649–657.

A Datasets

This section first details the construction of each dataset.

Trustpilot (TP) . Trustpilot sentiment dataset (Hovy et al., 2015) contains reviews associated with a sentiment score on a five point scale, and each review is associated with 3 attributes: gender, age and location, which are self-reported by users. The original dataset is comprised of reviews from different locations, however, in this paper, we only derive TP-US for study. Following (Coavoux et al., 2018), we extract examples containing information of both gender and age, and treat them as the private information.

AG news . AG news corpus (Del Corso et al., 2005) aims to predict the topic label of the document, with four different topics in total. Following (Zhang et al., 2015; Jin et al., 2019), we use both "title" and "description" fields as the input document.

Dataset	Private Variable	#Train	#Dev	#Test
TP-US	age, gender	22,142	2,767	2,767
AG	named entity	11,657	1,457	1,457
Blog	age, gender	7,098	887	887

Table 4: Summary of the studied NLP datasets.

We use full AG news dataset for MEA, which we call AG news (full). As AIA takes the entity as the sensitive information, we use the corpus filtered by (Coavoux et al., 2018), which we call AG news. The resultant AG news merely includes sentences with the five most frequent person entities, and each sentence contains at least one of these named entities. Thus, the attacker can identify these five entities as five independent binary classification tasks.

Blog posts (Blog) . We derive a blog posts dataset from the blog authorship corpus (Schler et al., 2006). We recycle the corpus preprocessed by (Coavoux et al., 2018), which covers 10 different topics. Similar to TP-US, the private variables are comprised of the age and gender of the author.

We provide the statistics of all datasets in Table 4. Table 5 presents the uni-gram and 5-gram overlapping between test sets and different queries. According to Table 5, AG news is closer to news data, whereas Blog and TP-US are more similar to reviews data, which validates our claim in Section 4.2.

B Supplementary Studies

B.1 Impact of Prediction Sharpness

Interestingly, compared with the queries from the same distribution, Table 2 also shows that queries from different distributions make AIA **easier** (see the best results corresponding to the lower privacy in bold in Table 2). We hypothesize this counter-intuitive phenomenon is due to that the posterior probability of the same distribution is sharper than that of the different distribution. This argument can be further strengthened in Section 5, in which we use a temperature coefficient τ at the softmax layer to control the sharpness of the posterior probability. We conjecture that if the model is *less confident* on its most likely prediction, then AIA is more likely to be successful. This speculation is confirmed by Figure 2, where the higher posterior probability leads to a higher empirical privacy.

B.2 Impact of Attribute Distribution

We further investigate which attribute is more vulnerable, *i.e.,* the relationship between attribute distribution (histogram variance) and privacy leakage. Table 6 empirically indicates that attributes with higher variances cause more information leakage or a lower empirical privacy. For example, for AG-news, entity 2-4 with higher variances result in lower empirical privacy, while entity 0-1 are more resistant to AIA. For TP-US and Blog, as age and gender exhibit similar distribution, AIA performance gap across these two attributes is less obvious.

B.3 Architecture Mismatch

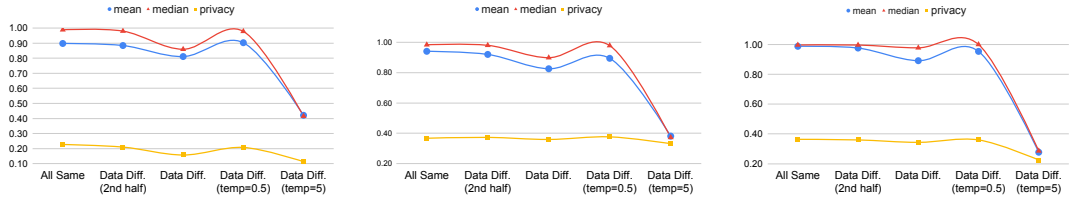
In practice, it is more likely that the adversary does not know the victim's model architecture. A natural question is whether our attacks are still possible when the extracted models and the victim models have different architectures, *i.e.,* architectural mismatch. To study the influence of the architectural mismatch, we fix the architecture of the extracted model, while varying the victim model from BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) to XLNET (Yang et al., 2019). According to Table 7, when there is an architecture mismatch between the victim model and the extracted model, the efficacy of AIA is alleviated as expected. However, the leakage of the private information is still severe (compare to the majority class in Table 2). Surprisingly, we observe that MEA cannot benefit from a more accurate victim, which is different from the findings in (Hinton et al., 2015). For example, the victim model performs best using XLNET-large, while MEA shows best performance when the victim model uses XLNET-base. We conjecture such difference is ascribed to the distribution mismatch between the training data of the victim model and the queries. We will conduct an in-depth study on this in the future.

Query	AG news		Blog		TP-US	
	uni-gram	5-gram	uni-gram	5-gram	uni-gram	5-gram
reviews	68.22%	0.53%	47.21%	0.73%	60.86%	2.57%
news	72.13%	1.24%	44.76%	0.06%	51.28%	0.12%

Table 5: Percentage of uni-gram and 5-gram recall-based overlap between different queries and test sets.

	AG news				
	entity 0 (std=310.0)	entity 1 (std=1568.5)	entity 2 (std=2095.5)	entity 3 (std=2640.5)	entity 4 (std=2615.5)
$\mathcal{T}_A = \mathcal{T}_V$	15.61	15.10	7.71	6.95	5.49
$\mathcal{T}_A \neq \mathcal{T}_V$ (news)	14.79	12.38	3.84	5.33	2.02
	TP-US		Blog		
	gender (std=1512.0)	age (std=1440.0)	age (std=28.0)	gender (std=6.0)	
$\mathcal{T}_A = \mathcal{T}_V$	36.40	37.12	32.18	39.02	
$\mathcal{T}_A \neq \mathcal{T}_V$ (reviews)	36.44	37.40	31.20	38.01	

Table 6: AIA performance on attributes of different datasets. All experiments are based on 1x queries. std is the standard deviation of attribute distribution.



(a) AG news (b) TP-US (c) Blog

Figure 2: The correlation between the empirical privacy of AIA and the maximum posterior probability. **mean** and **median** denote the mean and median of the maximum posterior probability of the queries.

Victim	**Extracted**	**TP-US**		
		victim ↑	MEA ↑	AIA ↓
BERT-large	BERT-base	86.82	85.36	36.65
RoBERTa-large	BERT-base	87.20	85.72	37.33
RoBERTa-base	BERT-base	86.66	85.40	37.52
XLNET-large	BERT-base	87.21	85.99	37.68
XLNET-base	BERT-base	86.91	**86.13**	38.09
BERT-base	BERT-base	85.53	85.15	**35.97**

Table 7: Attack performance on TP-US with mismatched architectures between the victim model and the extracted model.