Membership Inference Attacks Against NLP Classification Models

Virat Shejwalkar[1] **Huseyin A. Inan**[2] **Amir Houmansadr**[1] **Robert Sim**[2]
[1]University of Massachusetts, Amherst [2] Microsoft Research
{vshejwalkar,amir}@cs.umass.edu {Huseyin.Inan,rsim}@microsoft.com

Abstract

The success of natural language processing (NLP) is making NLP applications commonplace. Unfortunately, recent research has shown that privacy might be at stake given that these models are often trained on private user data. While privacy risks are demonstrated in text generation settings, privacy risks of the text classification settings, which subsume myriad downstream applications, are largely unexplored. In this work, we study the susceptibility of NLP classification models, used for text classification tasks, to *membership inference* (MI), which is a fundamental type of privacy leakage. We design a comprehensive suite of attacks to assess the risk of *sample-level MI*, as well as that of relatively unexplored *user-level MI*. We introduce novel user-level MI attacks that outperform the existing attacks and conduct experiments on Transformer-based and RNN-based NLP models. Our evaluations show that user-level MI is significantly stronger than sample-level MI. We further perform in-depth analyses showing the effect of various NLP-specific parameters on MI against NLP classification models.

1 Introduction

In the past few years, natural language processing (NLP) has made a tremendous progress [2] and has facilitated multiple NLP applications in our daily lives [10]. However, machine learning models are known to memorize their (potentially private) training data, and NLP models are no exceptions. As a result, NLP models can unintentionally leak private information in various forms, e.g., membership inference attacks (MIAs) [22], text reconstruction [18], and text extraction [3]. MIA is the most fundamental type of privacy leakage, which can represent other, more complex, forms of leakage. Given a target model and a target sample, the MIA adversary aims to infer whether the sample was a member of the private training data of the target model.

While MIAs are extensively studied in the settings of image classification [22, 27, 24, 19, 16, 25, 7, 4, 12] and text generation [23, 3, 18], to the best of our knowledge, they are not well-studied for text classification. Unfortunately, the peculiarities of NLP for text classification do not allow to trivially extend the understanding of MIAs from (the well-explored) image classification and text generation domains to MIAs on text classification domain. *Hence, in this work, we focus on privacy risks due to NLP models in text classification settings*.

To this end, we take the first step towards systematically evaluating the privacy risks due to NLP classification models (here onward, simply called "NLP models") to their private training data. We use MIAs for our privacy risk analyses.

Sample-level versus user-level MIAs: A majority of prior works on MIAs focus on *sample-level MIAs*, where the adversary aims to infer the membership of a single sample. However, in most practical settings, a model is trained on data collected from users where each user may contribute multiple data samples, and the data of different users may overlap. Hence, inferring the membership of a single target sample may not necessarily violate any user's privacy, but inferring membership of

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

a target user can be deemed sensitive (Figure 1). Hence, we argue to analyze NLP models through the lens of *user-level MIAs*, along with sample-level MIAs. Given a target model and a target user's data, *the user-level MIA adversary aims to infer whether the target user's data was part of the training data of the target model*. Intuitively, compared to sample-level MIAs, we expect user-level MIAs to be stronger, as they can combine information from multiple samples of a target user. We will show that this is indeed the case, and user-level MIAs can be as much as 25% more accurate.

Comprehensive suite of MIAs: For a thorough analysis of privacy risks due to NLP models under two practical threat models (Section 2), we design a comprehensive suite of MIAs that contain sample-level MIAs (SMIAs) and our novel user-level MIAs (UMIAs). SMIAs are extensively explored in image classification, and we adopt the simple, yet effective, threshold-based SMIA [24].

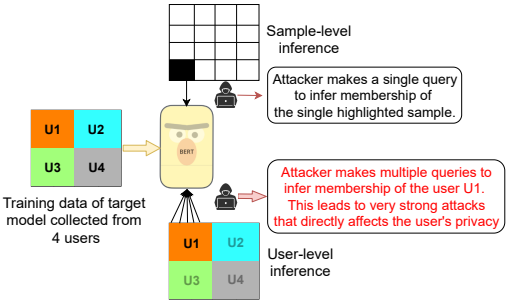

Figure 1: Sample versus user-level membership inference.

For user-level MIAs (UMIAs), we first compute a feature for each of the samples in the target user's data. An example feature is cross-entropy loss of a sample with respect to the target model. Next, for each target user, we compute an aggregate of all features, called *aggregate feature*, and use it to make membership inference. Based on the type of the aggregate feature, we design three user-level MIAs: (1) *Threshold-based* UMIAs use a single statistic, e.g., average, of all the feature as the aggregate feature, (2) *Learning-based* UMIAs first compute multiple statistics of all the features (e.g., average, minimum, maximum, variance). Then they use the vector of these statistics as the aggregate feature. (3) *Sample-to-user-level* UMIAs first use threshold-based SMIAs to infer membership status of each of the samples in target user's data. Then they use the average of all the membership statuses as the aggregate feature. The three key contributions (**C**'s) of our work are:

(C1) We perform *the first* systematic analyses of privacy risks due to NLP classification models.
(C2) We provide a comprehensive suite of membership inference attack frameworks that can be used to mount various *sample-level* and our *novel user-level* MIAs.
(C3) Our extensive evaluation using two real-world text classification datasets provides various interesting insights including, *(1)* depending on the combination of model and data, UMIAs are 11% to 25% more accurate than SMIAs, *(2)* the more the number of classes of data a user contributes to, the more vulnerable they become to UMIAs, *(3)* the pre-training methodology of transformer models [5, 13] can significantly impact the privacy of downstream applications.

2 Membership Inference Attacks Against NLP Classification Models

In this section, we first detail the MIA threat model we consider, then provide a comprehensive suite of sample-level and user-level MIAs for the privacy leakage assessment of NLP classification models.

Threat model. Given a *target model* M_θ trained on a private training data D_{tr} and a *target sample* $(\mathbf{x}, y)$, the goal of sample-level MIAs (SMIAs) is to infer whether the target sample is a part (member) of the private training data of the target model M_θ. Following previous works [22, 27, 24, 19], we assume *gray-box* (also known as *black-box*) access to the target model, i.e., access to output prediction vector $M_\theta(\mathbf{x})$ for $(\mathbf{x}, y)$. As discussed in Section 1, our aim is to understand privacy risks in practical settings where text classification models are deployed as an oracle. Furthermore, we also evaluate our attacks under the more practical and difficult setting of *label-only* access to the target model. Similar to previous works, we assume access to some *shadow data*, i.e., data drawn from the same distribution as D_{tr}, but disjoint from D_{tr}.

Threat model of user-level MIAs (UMIAs) is the same as that of sample-level MIAs, except for their goal. Given a *target model* M_θ and the data D_* of a *target user*, the goal of UMIAs is to infer whether D_* is a part of the private training data of M_θ.

2.1 Threshold-based Sample-level MIAs

There are numerous SMIAs in the literature. However [27, 24, 21] observe that the basic threshold-based SMIAs (T-SMIAs) perform as well as the more sophisticated neural networks based gray-box or white-box SMIAs [11, 16]. Hence, we only adopt T-SMIAs in our attack suite. There are a number of T-SMIAs based on different features (e.g., loss of the target model on a target sample) that

T-SMIAs use to differentiate between member and non-member samples. We explore the following five features in our work: **(1)** cross-entropy loss of M_θ on $(\mathbf{x}, y)$ [16], **(2)** modified entropy of M_θ on $(\mathbf{x}, y)$ [24], **(3)** rank of y in $M_\theta(\mathbf{x})$ [23], **(4)** confidence of $M_\theta(\mathbf{x})$ on y [22], **(5)** correctness of M_θ on $(\mathbf{x}, y)$ [27]. We formally define each feature and give details of T-SMIAs to Appendix A.1.

2.2 Baseline User-Level MIAs

In Section 1, we argued for privacy risk analyses of NLP models using user-level MIAs. As a baseline for user-level MIAs (UMIAs), we build upon the T-SMIAs from Section 2.1 and generalize the approach in [9]. Specifically, we compute an aggregate statistic (e.g., average) of all of the feature values of a user and use this statistic, called *aggregate feature*, to make the final inference. Unless specified otherwise, we use *average of feature values* as the aggregate feature for each user. It is worth noting that we choose this statistic after extensively exploring various statistics, including variance and minimum; we provide their results in Appendix A.2. Finally, we compute a threshold τ that separates the aggregate features for members and non-members in shadow data, and use τ to decide membership of the target user. We formalize this in Algorithm 2.

2.3 Our Novel User-Level MIAs

We next propose novel UMIAs that outperform the baseline UMIAs, as we show in our experiments.

Learning based User-level MIAs: This class of UMIAs uses a machine learning model to learn to differentiate between the behavior of the shadow model (trained on the shadow member users' data) towards member and non-member users in the shadow data. These attacks are motivated from a long line of research [22, 19, 14] on learning based MIAs. Algorithm 3 describes a general framework to mount UMIAs for any of the aforementioned features (e.g. loss). We first compute a single vector by concatenating four statistics (average, minimum, maximum, variance) of the set of feature values for each user in shadow data. We label the vector as a member or a non-member depending on the true membership of the corresponding user in the shadow data. This gives us training data to train our UMIA model; we use logistic regression as our UMIA model. Then, we use this UMIA model to infer the membership of target users using the vector of the statistics of their features values.

Sample-to-User-Level MIAs: Note that, sample-level MIAs become stronger when mounted for each class separately [22, 19, 24]. This is because, the difficulty of learning a particular class can vary and result in significantly different generalization errors (i.e., difference between the model's performance on train and test data). Generalization error strongly correlates with MIAs' success, hence per-class MIAs are stronger than a single, class-independent MIA. Unfortunately, the previous UMIAs cannot fully exploit the susceptibility of each class in the data, as they compute aggregate statistics irrespective of the classes in target user's data. Therefore, in this section, we improve UMIAs by fully exploiting the susceptibility of each class in the data.

Our intuition is to first mount a T-SMIA from Section 2.1 on each sample in a target user's data and then aggregate the results to infer the membership of that user. Therefore, we call them *Sample-to-user level MIAs (S2U-MIAs)*. The method of our S2U-MIAs (Algorithm 4) is: **(1)** we obtain class-wise thresholds $\tau_{y \in [C]}$ using Algorithm 1, **(2)** infer the membership status (MS) of each sample $(\mathbf{x}, y^*)$ in the target user's data using τ_{y^*} (MS=1 if member and 0 otherwise), and **(3)** infer the user as a member if the average of MS of all samples in the user's data is more than 0.5 and a non-member otherwise (i.e. majority vote over samples).

3 Empirical results

We use Reddit and Amazon datasets for our evaluations. Each sample of both the datasets has a user id, text (comment or review), class (subreddit or product). We use area under ROC curve and accuracy to measure attack performances. We defer further experimental setup details in Appendix B.

Table 1 presents AUCs and accuracies of all MIAs (Section 2) for our default settings (Appendix A). The two key takeaways from comparisons between SMIAs and UMIAs are as follows: **(1)** ***UMIAs are significantly stronger, hence, pose more severe threat than SMIAs***. Under the gray-box threat model, AUC of the strongest UMIA exceed that of the strongest SMIA by 0.19 for Reddit with BERT and 0.27 for Amazon with BERT. We note that, the successes of SMIAs and UMIAs strongly correlate with E_{gen}, hence for Reddit (Amazon), MIAs are stronger on BERT (LSTM) than on LSTM (BERT). **(2)** ***UMIAs perform very well even with just the label-only access to the target model*** with AUC above 80 for BERT and above 68 for LSTM models. We observe that, under label-only access,

Dataset	Model A_{test} (E_{gen})	Threat Model	Feature	Sample-level MIA Algorithm 1 $A_{\mathsf{s-th}}$	User-level MIAs Algorithm 2 $A^{(\mathsf{baseline})}_{\mathsf{u-th}}$	Algorithm 3 $A_{\mathsf{u-nn}}$	Algorithm 4 A_{s2u}
Reddit	BERT 18.7 (33.5)	Gray-box	loss	0.745 (68.9)	0.894 (82.3)	0.916 (85.5)	**0.935 (85.6)**
			mentropy	0.744 (69.0)	0.886 (83.1)	0.874 (84.4)	0.891 (85.5)
			rank	0.734 (68.8)	0.903 (85.1)	0.829 (81.1)	0.909 (84.9)
			confidence	**0.745 (69.2)**	0.822 (73.8)	0.829 (74.3)	0.890 (83.7)
		Label-only	correctness	0.652 (62.3)	0.835 (77.1)	0.833 (77.3)	0.811 (76.6)
	LSTM 13.2 (12.5)	Gray-box	loss	0.682 (63.2)	0.777 (69.7)	0.816 (73.7)	**0.831 (74.2)**
			mentropy	**0.683 (63.4)**	0.779 (69.7)	0.824 (73.4)	0.819 (74.2)
			rank	0.673 (63.0)	0.772 (70.4)	0.751 (68.6)	0.781 (72.0)
			confidence	0.682 (62.2)	0.716 (62.4)	0.719 (65.5)	0.774 (73.2)
		Label-only	correctness	0.567 (55.1)	0.688 (63.5)	0.640 (63.0)	0.595 (56.6)
Amazon	BERT 75.4 (15.6)	Gray-box	loss	0.597 (58.7)	0.827 (73.7)	0.857 (81.6)	0.849 (74.2)
			mentropy	**0.598 (58.7)**	0.859 (76.7)	**0.870 (85.1)**	0.862 (75.8)
			rank	0.587 (58.4)	0.792 (71.2)	0.600 (70.1)	0.812 (68.6)
			confidence	0.597 (58.7)	0.759 (65.4)	0.814 (72.9)	0.813 (74.6)
		Label-only	correctness	0.585 (58.5)	0.800 (69.5)	0.800 (72.5)	0.822 (75.2)
	LSTM 58.7 (20.7)	Gray-box	loss	0.632 (61.6)	0.799 (70.7)	0.925 (86.1)	0.831 (75.4)
			mentropy	**0.635 (61.7)**	0.843 (74.9)	**0.940 (88.2)**	0.835 (74.8)
			rank	0.611 (60.8)	0.788 (70.5)	0.819 (75.1)	0.820 (63.1)
			confidence	0.632 (61.6)	0.724 (63.4)	0.921 (83.9)	0.831 (75.4)
		Label-only	correctness	0.603 (60.2)	0.765 (67.5)	0.732 (64.1)	0.766 (65.6)

Table 1: Comparison of AUCs (accuracy in parentheses) of SMIAs and UMIAs for gray-box and label-only threat models. We bold the results of the best of SMIAs and the best of UMIAs.

any of UMIAs performs better than the best of SMIAs. This implies that an adversary can correctly infer the presence of a target user in the private training data with high confidence by just looking at the output label of the target model, which highlights the severity of our UMIAs.

Our UMIAs outperform the state-of-the-art UMIAs [9, 23]. Algorithm 2 with loss feature corresponds to the UMIA of [9]; $A^{\mathsf{baseline}}_{\mathsf{u-th}}$ column of Table 1 shows its performance. We note that our novel UMIAs ($A_{\mathsf{u-nn}}$ and $A_{\mathsf{u-s2u}}$ columns of Table 1) from Section 2.3 outperform the above baseline UMIAs. Song et al. [23] propose a learning based attack that uses histogram of ranks of the tokens in target user's data to decide the user's membership. The attack is devised for text generation setting, but it can be trivially extended to our text classification setting by using histogram of ranks of the correct labels in target user's data. We note in our experiments that, for Reddit + BERT (LSTM), AUC of their attack is 0.907 (0.812) while AUC of our best UMIA is 0.935 (0.831); for Amazon + BERT (LSTM), AUC of their attack is 0.831 (0.878) while AUC of our best UMIA is 0.857 (0.94).

Key takeaways from our experiments with varying the learning parameters Due to space restrictions, we defer detailed experimental results to Appendix C and provide key takeaways below.
(1) Increasing the number of samples per user, at first increases the susceptibility of NLP models to user-level MIAs, but as the number of samples/user become very large, the the susceptibility reduces. However, increasing the number of samples/user always weakens the sample-level MIAs.
(2) Increasing the number of users reduces the susceptibility to both user- and sample-level MIAs. While increasing the number of classes in data, increases the susceptibility to both types of MIAs.
(3) Increasing the number of parameters (capacity) of a transformer model need not always increase their susceptibility to MIAs. Instead we observe that, the pre-training methodology of transformer models [5, 13] can significantly impact the privacy of downstream applications.
(4) Tokenization algorithm used to preprocess the text can significantly impact the privacy of downstream applications. For instance, we observe that the models which use word-piece tokenization are *less* susceptible to MIAs than the models which use word-level tokenization.
(5) Increasing the size of vocabulary during tokenization increases the susceptibility of downstream models to both types of MIAs, because of reduced number of out-of-vocabulary tokens.
(6) The more the number of classes of data a user contributes to, the more vulnerable they become to UMIAs.

We hope that our comprehensive suite of membership inference attacks will help practitioners understand a lower bound on the privacy leakage due to NLP classification models. Furthermore, our work opens up a few interesting avenues for future works, e.g., 1) using user-level MIAs to audit fairness of data collection process, 2) using text perturbations to improve MIAs.

References

[1] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839, 2020.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, and Melanie Subbiah. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[3] 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. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association, August 2021.

[4] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International Conference on Machine Learning*, pages 1964–1974. PMLR, 2021.

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

[6] Alexej Gossmann. Probabilistic interpretation of auc. `https://www.alexejgossmann.com/auc/`, 2018. [Online; accessed 7-September-2021].

[7] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152, 2019.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Abhyuday Jagannatha, Bhanu Pratap Singh Rawat, and Hong Yu. Membership inference attack susceptibility of clinical language models. *arXiv preprint arXiv:2104.08305*, 2021.

[10] Daniel Jurafsky and James Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 2. 02 2008.

[11] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1605–1622, 2020.

[12] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[14] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 521–534. IEEE, 2020.

[15] Edward Loper and Michael Heilman. Nltk 3.6 documentation. `https://www.nltk.org/_modules/nltk/tokenize/treebank.html`, 2021. [Online; accessed 22-July-2021].

[16] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. *Security and Privacy (SP), 2019 IEEE Symposium on*, 2019.

[17] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

[18] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331, 2020.

[19] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *In NDSS*, 2019.

[20] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[21] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9549–9557, 2021.

[22] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, 2017.

[23] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

[24] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[25] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Towards demystifying membership inference attacks. *arXiv preprint arXiv:1807.09173*, 2018.

[26] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[27] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing connection to overfitting. *In 2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 2018.

A Missing Details of Membership Inference Attacks Against NLP Classification Models

In this section, we provide a missing details of our comprehensive suite of sample-level and user-level MIAs for the privacy leakage assessment of NLP classification models.

Notations: Let M_θ be the target model, $(\mathbf{x}, y)$ be a target sample (for sample-level MIAs), and D_* be a target user's data (for user-level MIAs). $\bar{D}$ ($\bar{D}'$) is the union of the data of n (n') shadow member (non-member) users. $M_{\bar{\theta}}$ is the shadow model trained on $\bar{D}$. D^j denotes j^{th} sample in a sample-level MIA, while D_i^j denotes j^{th} sample of user i in a user-level MIA. C is the number of classes in data. We use F to represent different features (e.g. loss, confidence, rank, etc.) that we use in our attacks. For an integer p, $[p]$ denotes integers in range $[1, p]$. $|D|$ denotes the size of D. $\mathbb{1}\{g\}$ is an indicator function which outputs 1 if g is true and 0 otherwise. For a set of values T, T_{avg}, T_{max}, T_{min}, and T_{var} denote average, maximum, minimum, and variance of T, respectively.

Algorithm 1 Threshold-based Sample-level MIA ($A_{\mathsf{s-th}}$)

1: **Input**: $\bar{D}$, $\bar{D}'$, $M_{\bar{\theta}}$, M_θ, $(\mathbf{x}, y)$
2: Obtain class-wise shadow data: $\bar{D}_{y\in[C]}$, $\bar{D}'_{y\in[C]}$
3: **for** $y \in [C]$ **do**
4: $\quad \{\bar{f}_y^i = F(M_{\bar{\theta}}, \bar{D}_y^i)\}_{i\in[|\bar{D}_y|]}$; $\{\bar{f}_y'^i = F(M_{\bar{\theta}}, \bar{D}_y'^i)\}_{i\in[|\bar{D}'_y|]}$
5: $\quad \tau_y = \underset{\tau\in\mathbb{R}}{\text{argmax}} \left(\sum_{i\in[|\bar{D}_y|]} \mathbb{1}\{\bar{f}_y^i \geq \tau\} + \sum_{i\in[|\bar{D}'_y|]} \mathbb{1}\{\bar{f}_y'^i < \tau\} \right)$
6: **end for**
7: **Output** $\mathbb{1}\{F(M_\theta, (\mathbf{x}, y)) \geq \tau_y\}$

A.1 Threshold-based Sample-level MIAs

There are numerous SMIAs in the literature. However [27, 24, 21] observe that the basic threshold-based SMIAs (T-SMIAs) perform as well as the more sophisticated neural networks based gray-box or white-box SMIAs [11, 16]. Hence, we only adopt T-SMIAs in our attack suite. There are a number of T-SMIAs based on different features (e.g., loss of the target model on a target sample) that T-SMIAs use to differentiate between member and non-member samples. Here, we give a general methodology and intuition behind these attacks.

Method: As in the previous work [24], we compute separate thresholds for each class in the data to make the attacks more effective. T-SMIAs in Algorithm 1 use shadow member and non-member data ($\bar{D}$ and $\bar{D}'$) to compute the class-specific thresholds. First, T-SMIAs train a shadow model $M_{\bar{\theta}}$ on $\bar{D}$. Then, for each shadow member sample $\bar{D}_y^i$ and non-member sample $\bar{D}_y'^i$ of class y, T-SMIAs compute a feature (e.g. model loss) using the sample and $M_{\bar{\theta}}$ (line 4 of Algorithm 1). Then, for each class y, they compute a threshold τ_y that best separates the features of shadow members and non-members (line 5 of Algorithm 1). Finally, T-SMIAs compute the feature value for the target sample $(\mathbf{x}, y)$, and use τ_y to make the final inference.

One can use different features (F in Algorithm 1) to mount T-SMIAs; we explore the following five features in our work:

- (1) Cross-entropy loss [16]

$$L_{CE}(M_\theta(\mathbf{x}), y) = -\log(M_\theta(\mathbf{x})_y).$$

- (2) Modified entropy of [24]

$$\text{Mentr}(M_\theta(\mathbf{x}), y) = -(1 - M_\theta(\mathbf{x})_y)\log(M_\theta(\mathbf{x})_y) - \sum_{c\neq y} M_\theta(\mathbf{x})_c \log(M_\theta(\mathbf{x})_c).$$

- (3) Rank of correct label [23]

$$\text{Rank}(M_\theta(\mathbf{x}), y) = r \quad \text{s.t.} \quad M_\theta(\mathbf{x})_y = M_\theta^s(\mathbf{x})[r]$$

where $M_\theta^s(\mathbf{x})$ is $M_\theta(\mathbf{x})$ sorted from high to low values and $v[r]$ is the r^{th} index of v.

- (4) Confidence of the correct label [22]: $M_\theta(\mathbf{x})_y$.
- (5) Correctness [27]: $\mathbb{1}\{y = \underset{c}{\text{argmax}}\ M_\theta(\mathbf{x})_c\}$.

Note that, features (1)-(4) are *gray-box features*, i.e., for sample $(\mathbf{x}, y)$, adversary needs the prediction vector $M_\theta(\mathbf{x})$ to compute them, while feature (5) is a *label-only feature*, i.e., just $\arg\max M_\theta(\mathbf{x})$ is required to compute it.

Intuition: The intuition behind T-SMIAs is that the target model M_θ is trained on the training data (members) and hence, performs well on the training data, but this may not equally apply to the test data (non-members). This in turn creates a gap between how the model performs on members and non-members. Therefore, various (aforementioned) features of M_θ become sufficiently different for members and non-members. Hence, the adversary can find a threshold that separates the member from non-member features and use it to infer the membership of any target sample.

Algorithm 2 Baseline User-level MIA ($A^{(\mathsf{baseline})}_{\mathsf{u-th}}$)

Input: $\bar{D}$, n, $\bar{D}'$, n', $M_{\bar{\theta}}$, M_θ, D_*

$\bar{F}^i = \{F(M_{\bar{\theta}}, \bar{D}_i^j)\}_{j\in[|\bar{D}_i|]} \quad \forall i \in [n]$

$\bar{F}'^i = \{F(M_{\bar{\theta}}, \bar{D}_i'^j)\}_{j\in[|\bar{D}'_i|]} \quad \forall i \in [n']$

$$\tau = \underset{\tau\in\mathbb{R}}{\text{argmax}} \left(\sum_{i\in[n]} \mathbb{1}\{\bar{F}^i_{\mathsf{avg}} \geq \tau\} + \sum_{i\in[n']} \mathbb{1}\{\bar{F}'^i_{\mathsf{avg}} < \tau\} \right)$$

Output $\mathbb{1}\{F^*_{\mathsf{avg}} \geq \tau\}$ where $F^* = \{F(M_\theta, D_*^j)\}_{j\in[|D_*|]}$

A.2 Baseline User-Level MIAs

In Section 1, we argued for privacy risk analyses of NLP models using user-level MIAs, because the training data of modern NLP models often contains multiple samples from a user. As a baseline for user-level MIAs (UMIAs), we build upon the T-SMIAs from Section 2.1 and generalize the approach in [9]. Specifically, we compute an aggregate statistic (e.g., average) of all of the feature values of a user and use this statistic, called *aggregate feature*, to make the final inference. Unless specified otherwise, we use *average of feature values* as the aggregate feature for each user. It is worth noting that we choose this statistic after extensively exploring various statistics, including variance and minimum; we provide their results in Table 2.

Finally, we compute a threshold τ that separates the aggregate features for members and non-members in shadow data, and use τ to decide membership of the target user. We formalize this in Algorithm 2.

A.3 Our Novel User-Level MIAs

We next propose novel UMIAs that outperform the baseline UMIAs, as we will show in our experiments.

Learning based User-level MIAs: This class of UMIAs uses a machine learning model to learn to differentiate between the behavior of the shadow model (trained on the shadow member users' data) towards member and non-member users in the shadow data. These attacks are motivated from a long line of research [22, 19, 14] on learning based MIAs.

Algorithm 3 describes a general framework to mount UMIAs for any of the aforementioned features (e.g. loss). We first compute a single vector by concatenating four statistics (average, minimum, maximum, variance) of the set of feature values for each user in shadow data. We label the vector as a member or a non-member depending on the true membership of the corresponding user in the shadow data. This gives us training data to train our UMIA model; we use logistic regression as our UMIA model. Then, we use this UMIA model to infer the membership of target users using the vector of the statistics of their features values.

Sample-to-User-Level MIAs: Note that, sample-level MIAs become stronger when mounted for each class separately [22, 19, 24]. This is because, the difficulty of learning a particular class can vary and result in significantly different generalization errors (i.e., difference between the model's performance on train and test data). Generalization error strongly correlates with MIAs' success, hence per-class MIAs are stronger than a single, class-independent MIA.

Algorithm 3 Learning based User-level MIA ($A_{\mathsf{u-nn}}$)

Input: $\bar{D}$, n, $\bar{D}'$, n', $M_{\bar{\theta}}$, M_θ, D_*
Compute sample-level features for shadow users:
$\bar{F}^i = \{F(M_{\bar{\theta}}, \bar{D}_i^j)\}_{j\in[|\bar{D}_i|]} \quad \forall i \in [n]$ ▷*member users*
$\bar{F}'^i = \{F(M_{\bar{\theta}}, \bar{D}_i'^j)\}_{j\in[|\bar{D}'_i|]} \quad \forall i \in [n']$ ▷*non-member users*
Compute feature-vector for shadow users
$\bar{f}_u^i = (\bar{F}^i_{\mathsf{avg}}, \bar{F}^i_{\mathsf{min}}, \bar{F}^i_{\mathsf{max}}, \bar{F}^i_{\mathsf{var}}) \quad \forall i \in [n]$ ▷*member users*
$\bar{f}_u'^i = (\bar{F}'^i_{\mathsf{avg}}, \bar{F}'^i_{\mathsf{min}}, \bar{F}'^i_{\mathsf{max}}, \bar{F}'^i_{\mathsf{var}}) \quad \forall i \in [n']$ ▷*non-member users*
Train attack model $\mathcal{A}$ on $(\{\bar{f}_u^i, 1\}_{i\in[n]} \cup \{\bar{f}_u'^i, 0\}_{i\in[n']})$
$F^* = \{F(M_\theta, D_*^j)\}_{j\in[|D_*|]}$ ▷*features of target user*
$F_u^* = (F^*_{\mathsf{avg}}, F^*_{\mathsf{min}}, F^*_{\mathsf{max}}, F^*_{\mathsf{var}})$ ▷*feature-vector of target user*
Output $\mathcal{A}(F_u^*)$

Algorithm 4 Sample to User level MIA (A_{s2u})

Input: $\bar{D}$, $\bar{D}'$, $M_{\bar{\theta}}$, M_θ, D_*
Compute per-class thresholds τ_y using Algorithm 1 lines 2-6
Set membership status $\text{MS}(\mathbf{x}, y) = \mathbb{1}\{F(M_\theta, (\mathbf{x}, y)) \geq \tau_y\}$
Set $F^* = \{\text{MS}(\mathbf{x}, y)\}_{(\mathbf{x},y)\in D_*}$
Output $\mathbb{1}\{F^*_{\mathsf{avg}} \geq 0.5\}$ ▷*majority vote over MS of samples*

Unfortunately, the previous UMIAs cannot fully exploit the susceptibility of each class in the data, as they compute aggregate statistics irrespective of the classes in target user's data. Therefore, in this section, we improve UMIAs by fully exploiting the susceptibility of each class in the data.

Our intuition is to first mount a T-SMIA from Section 2.1 on each sample in a target user's data and then aggregate the results to infer the membership of that user. Therefore, we call them *Sample-to-user level MIAs (S2U-MIAs)*. The method of our S2U-MIAs (Algorithm 4) is: **(1)** we obtain class-wise thresholds $\tau_{y\in[C]}$ using Algorithm 1, **(2)** infer the membership status (MS) of each sample $(\mathbf{x}, y^*)$ in the target user's data using τ_{y^*} (MS=1 if member and 0 otherwise), and **(3)** infer the user as a member if the average of MS of all samples in the user's data is more than 0.5 and a non-member otherwise (i.e. majority vote over samples).

Dataset	Model A_{test} (E_{gen})	Threat Model	Feature	Threshold-based User-level MIAs (Algorithm 2) Statistic used for attack			
				Average	Minimum	Maximum	Variance
Reddit	BERT 18.7 (33.5)	Gray-box	loss	**0.894 (82.3)**	0.658 (58.4)	0.882 (77.0)	0.861 (79.2)
			mentropy	**0.886 (83.1)**	0.658 (58.5)	0.869 (75.2)	0.868 (80.7)
			rank	**0.903 (85.1)**	0.536 (53.6)	0.895 (83.5)	0.901 (85.1)
			confidence	0.822 (73.8)	**0.882 (77.0)**	0.658 (58.4)	0.679 (63.8)
		Label-only	correctness	**0.835 (77.1)**	0.501 (50.0)	0.536 (53.6)	0.692 (62.5)
	LSTM 13.2 (12.5)	Gray-box	loss	**0.777 (69.7)**	0.639 (56.8)	0.754 (68.6)	0.751 (68.9)
			mentropy	0.779 (69.7)	0.640 (57.0)	**0.810 (74.2)**	0.751 (68.6)
			rank	**0.772 (70.4)**	0.559 (55.1)	0.743 (68.2)	0.739 (69.0)
			confidence	0.716 (62.4)	**0.754 (68.6)**	0.639 (56.8)	0.669 (58.9)
		Label-only	correctness	**0.688 (63.5)**	0.501 (50.0)	0.555 (55.5)	0.674 (61.9)
Amazon	BERT 75.4 (15.6)	Gray-box	loss	0.827 (73.7)	0.486 (51.3)	**0.856 (80.0)**	0.805 (76.8)
			mentropy	0.859 (76.7)	0.486 (51.3)	0.859 (73.8)	**0.955 (89.0)**
			rank	0.792 (71.2)	0.500 (50.0)	0.782 (70.0)	**0.811 (70.2)**
			confidence	0.759 (65.4)	**0.856 (80.0)**	0.486 (51.3)	0.122 (49.7)
		Label-only	correctness	**0.800 (69.5)**	0.530 (53.0)	0.500 (50.0)	0.144 (50.0)
	LSTM 58.7 (20.7)	Gray-box	loss	0.799 (70.7)	0.534 (52.2)	0.777 (70.2)	**0.809 (72.9)**
			mentropy	0.843 (74.9)	0.534 (52.2)	0.883 (79.3)	**0.933 (87.9)**
			rank	0.788 (70.5)	0.503 (50.2)	**0.819 (76.5)**	0.806 (72.9)
			confidence	0.724 (63.4)	**0.777 (70.2)**	0.534 (52.2)	0.265 (50.0)
		Label-only	correctness	**0.765 (67.5)**	0.504 (50.4)	0.503 (50.2)	0.310 (49.1)

Table 2: We design our threshold-based user-level MIA attack framework such that one can plug in various statistics of the feature values of samples from a target user's data. Here, we provide all of the results from our explorations of four statistics: average, minimum, maximum, and variance. We observe the average-based UMIAs perform better for majority of the features, hence, in main paper, we use average-based results.

B Missing Details of Experimental Setup

Datasets: **(1) Reddit** [1]: We use Reddit comments from a randomly chosen month (November 2017). Each sample of the dataset contains user id, comment, and subreddit (i.e., the comment topic). Here the labels are subreddits and the classification task is to predict the subreddit for a given comment. For our default setting of Table 1, we take the users with minimum 150 and maximum 500 comments from the top 256 subreddits; randomly sample 1,200 users and divide them equally in target member, non-member, and shadow member, non-member datasets; we use two disjoint sets of 30 users each for test and validation datasets. **(2) Amazon** [17]: We concatenate Amazon reviews data from 25 product categories to generate a dataset whose each sample contains user id, comment on a product, and category of the product. We train our models to predict a product category for a given comment. For our default setting of Table 1, we take the users with minimum 100 and maximum 300 comments from all 25 categories; rest of the setting is the same as that of Reddit.

Models: In our experiments, we use the BERT-base [5] and a 2-layer LSTM model [8].

Measurement metrics: We measure the success of MIAs using *area under ROC curve (AUC)* and *accuracy*. MIA AUC measures the *probability* that for a randomly selected pair of a member and a non-member user/sample, MIA assigns higher rank to the member than the non-member [6]. Note that, unlike accuracy, AUC is a more robust, threshold-independent metric; below, we provide MIA AUCs in most of the experiments. MIA accuracy is the *percentage* of correct membership predictions of the target samples/users. We maintain equal number of member and non-member users in the target data, hence accuracy and AUC of a random guess are 50% and 0.5, respectively.

Note that, *we use early stopping in all of our experiments to prevent overfitting to training data and ensure good generalization performance of models*. We report the test accuracy A_{test} of the snapshot of the model that achieves the best performance on validation data in the course of training We define generalization error E_{gen} of a model as the difference between its train and test accuracies, i.e., $E_{\mathsf{gen}} = A_{\mathsf{train}} - A_{\mathsf{test}}$. For each experiment, we present the average of three runs with three different seeds.

C Missing Experimental Results

Below, we study the impacts of various parameters of learning setting, including the ones that are specific to the training of NLP models. We use Reddit dataset for the following ablation studies. We only vary the parameter of interest, and keep the others as in the default setting of Reddit from Section B. For clarity, we present AUCs only of our S2U-MIAs, as they perform the best in most of the settings.

Number of samples per user: Figure 2-(a) shows the results when we vary the number of samples per user. The increase in the samples improve the test accuracy, A_{test}, and the generalization error, E_{gen}, as expected. Hence, SMIA performance, which strongly correlates with generalization error, E_{gen}, reduces for both BERT and LSTM.

Note that, for UMIAs, increasing the number of samples per user allows to make more queries to the target model and glean more information about target users, and hence, we expect UMIAs to be more effective. Interestingly, we observe this behavior only in the low data regime (up to on average 475 samples/user). But, when we further increase the number of samples/user, the UMIA AUC reduces. This is because, in high data regimes, the E_{gen} of model reduces significantly and this reduces UMIA performances.

Number of member users: Figure 2-(b) shows the results when we vary the number of users in the training data. Increasing the number of users increases the amount of training data. As expected, this increases the test accuracy, A_{test}, and reduces generalization error, E_{gen}. This in turn reduces the performances of both SMIAs and UMIAs. We observe the same behaviors for BERT and LSTM models.

Number of classes in dataset: Figure 2-(c) shows the results when we vary the number of classes in the data. With all the other parameters constant, increasing the number of classes increases the difficulty of the classification task. Hence, A_{test} reduces and E_{gen} increases, which makes both SMIAs and UMIAs more effective.

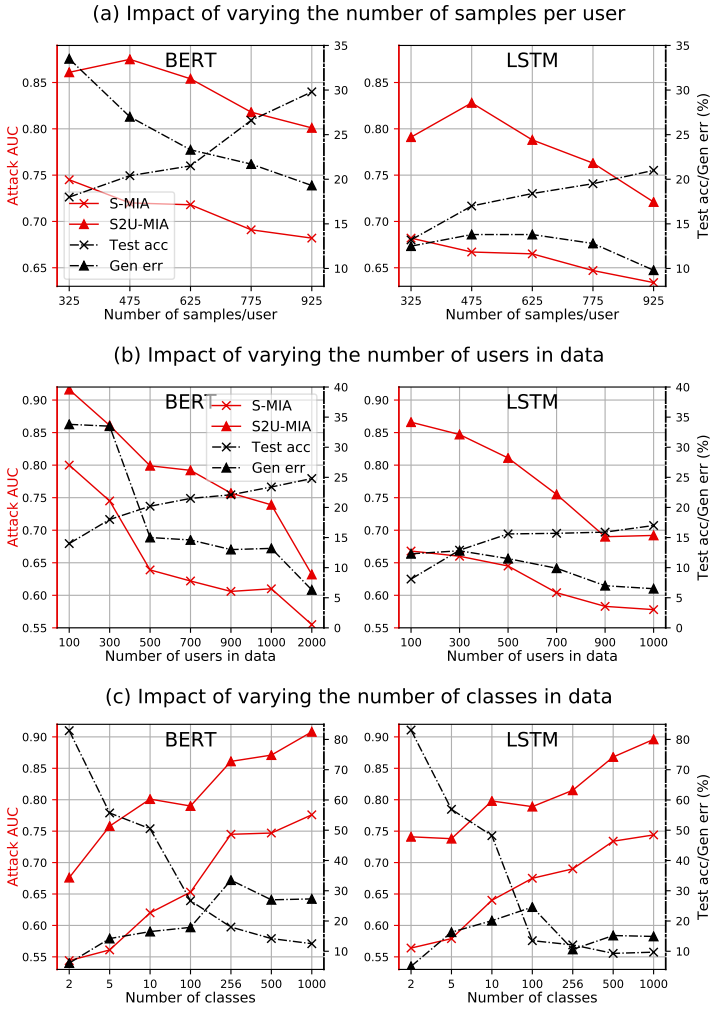

Figure 2: Impacts of varying the learning parameters on AUC of SMIAs and UMIAs, and on test accuracy (A_{test}) and generalization error (E_{gen}) of the target model.

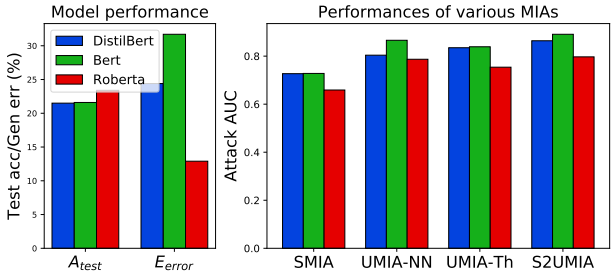

Figure 3: Impact of varying the architecture and number of trainable parameters of transformer-based models on MIAs.

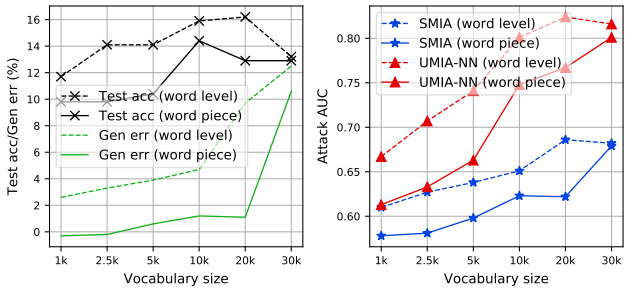

Figure 4: Impact of varying the size of vocabulary and the tokenization algorithm on MIAs against LSTM model.

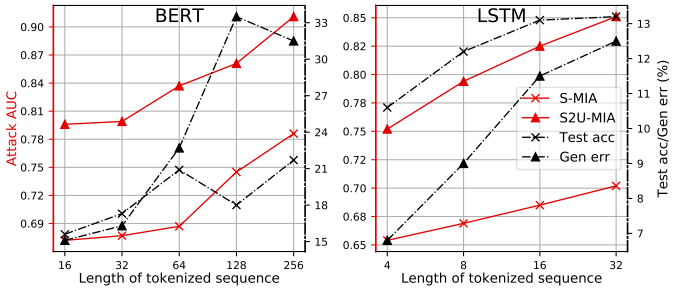

Figure 5: Impact of varying the maximum length of the tokenized input sequences on MIAs.

Number of trainable parameters: In this experiment we finetune three pre-trained models: DistilBERT [20], BERT-base [5], and Roberta [13], with 66 million, 108 million, and 148 million parameters, respectively. We mount all of our SMIAs and UMIAs from Section 2 and present the results in Figure 3.

Larger models tend to memorize their training data better than smaller models. Hence, as expected, we observe that BERT is more susceptible to membership inferences than DistilBert. Roberta is the largest among the three models and achieves the highest test accuracy. Interestingly, *it achieves the lowest generalization error, and hence, it is least susceptible to SMIAs and UMIAs.* This experiment highlights that *the pre-training methodology of transformer models can significantly impact the privacy of downstream applications.*

Impact of text tokenization parameters: Tokenization is an important pre-processing step in any NLP task. There are three important parameters of tokenization: **1)** tokenization algorithm (e.g., word-level [15] versus word-piece [26]), **2)** vocabulary size, and **3)** maximum length (number of tokens) of the tokenized model inputs.

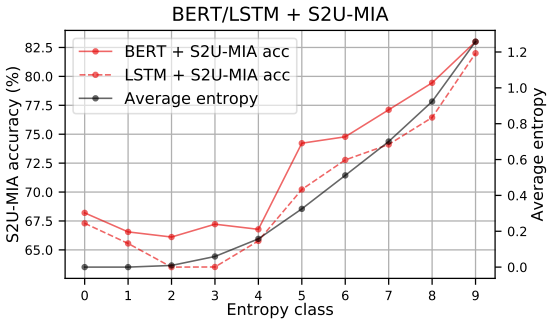

Figure 6: Understanding user-level MIAs

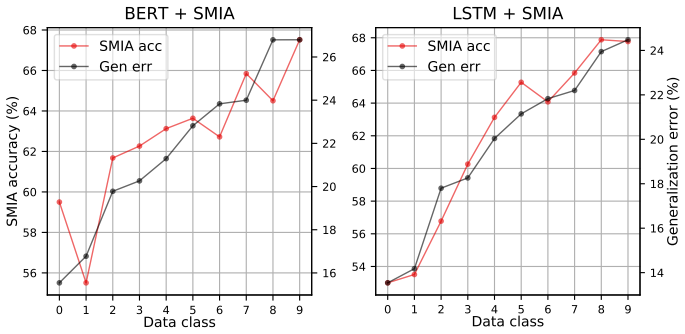

Figure 7: Understanding sample-level MIAs

Figure 4 shows the impacts of the first two parameters. We only use the LSTM model here, because the other model we use is pre-trained BERT-base whose tokenization and vocabulary are fixed. As the vocabulary size increases, the number of out-of-vocabulary words reduce, which improves the performance of MIAs. Interestingly, *word-piece tokenization weakens MIAs*, probably because, it splits words into sub-words which reduces specificity of the words and obfuscates the signal required for membership inference.

Figure 5 shows the impact of the maximum length of tokenized model inputs (i.e., sequence). As we increase the maximum length, the resulting models become more accurate but also more susceptible to MIAs. This is because, longer tokenized sequences expose more specific context of a given text. When tokenization truncates the text to obtain shorter sequences, this obfuscates the membership signal.

C.1 Understanding Sample and User-level MIAs

Below, we try to understand the sources of the susceptibility of NLP models to sample-level and user-level MIAs.

Sample-level MIAs: Figure 7 shows results plot for understanding sample-level MIAs, where we plot the generalization error and accuracy of sample-level MIA for each of 10 classes of data. Recall that we use 10 classes for this experiment to ensure sufficient number of samples in each class. Similar to previous works [24, 22], we observe that classes with higher E_{gen} are more susceptible to SMIAs. This is because, higher E_{gen} widens the gap between model's behaviour on members and non-members.

User-level MIAs: Here, we try to understand the effect of the distribution of classes in a target user's data on their susceptibility to UMIAs. We use entropy of the empirical distribution of classes (H_c) in a user's data as the proxy for the distribution of classes in the user's data. We compute H_c for all users, sort them based on H_c, and divide them in 10 buckets. Finally, for the users in each bucket, we plot the average H_c and the average of our S2U-MIA accuracy in Figure 6.

We observe in Figure 6 that the susceptibility to UMIAs increases with the increase in the entropy of class distribution. In other words, *those users who contribute to different classes (e.g., by participating in various subreddits) are more susceptible to user-level membership inferences.*

For Reddit, this is because: the users with low entropy (i.e., those with very few classes) tend to have only the top classes (i.e., the classes with larger number of samples) in their data. The top classes tend to have lower generalization error because the model trains on larger number of samples. Hence, such users are less susceptible to user-level MIAs. But, users with more number of classes in their data have the classes with low number of samples (along with the top classes). Due to the presence of such classes, the generalization error of such users is higher, and therefore, their susceptibility to UMIAs is also higher. Interestingly, UMIAs are stronger against users with a single class (zero entropy) than those with a few classes , probably because having just a single class makes these users more unique.