Canary Extraction in Natural Language Understanding Models

Rahil Parikh
Institute for Systems Research
University of Maryland

Christophe Dupuy
Amazon Alexa AI

Rahul Gupta
Amazon Alexa AI

Abstract

Natural Language Understanding (NLU) models can be trained on sensitive information such as phone numbers, zip-codes etc. Recent literature has focused on Model Inversion Attacks (ModIvA) that can extract training data from model parameters. In this work, we present a version of such an attack by extracting canaries inserted in NLU training data. In the attack, an adversary with open-box access to the model reconstructs the canaries contained in the model's training set. We evaluate our approach by performing text completion on canaries and demonstrate that by using the prefix (non-sensitive) tokens of the canary, we can generate the full canary. As an example, our attack is able to reconstruct a four digit code in the training dataset of the NLU model with a probability of 0.5 in its best configuration. As countermeasures, we identify several defense mechanisms that, when combined, effectively eliminate the risk of ModIvA in our experiments.

1 Introduction

Natural Language Understanding (NLU) models are used for different tasks such as question-answering (Hirschman and Gaizauskas, 2001), machine translation (Macherey et al., 2001) and text summarization (Tas and Kiyani, 2007). These models are often trained on crowd-sourced data that may contain sensitive information such as phone numbers, contact names and street addresses. Nasr et al. (2019), Shokri et al. (2017) and Carlini et al. (2018) have presented various attacks to demonstrate that neural-networks can leak private information. We focus on one such class of attacks, called Model Inversion Attack (ModIvA) (Fredrikson et al., 2015), where an adversary aims to reconstruct a subset of the data on which the machine-learning model under attack is trained on. We also demonstrate that established ML practices (e.g. dropout) offer strong defense against ModIvA.

In this work, we start with inserting potentially sensitive target utterances called 'canaries'[1] along with their corresponding output labels into the training data. We use this augmented dataset to train an NLU model f_θ. We perform a open-box attack on this model, i.e., we assume that the adversary has access to all the parameters of the model, including the word vocabulary and the corresponding embedding vectors. The attack takes the form of text completion, where the adversary provides the start of a canary sentence (e.g., 'my pin code is') and tries to reconstruct the remaining, private tokens of an inserted canary (e.g., a sequence of 4 digit tokens). A successful attack on f_θ reconstructs all the tokens of an inserted canary. We refer to such a ModIvA as 'Canary Extraction Attack' (CEA). In such an attack, this token reconstruction is cast as an optimization problem where we minimize the loss function of the model f_θ with respect to its inputs (the canary utterance), keeping the model parameters fixed.

Previous ModIvAs were conducted on computer vision tasks where there exists a continuous mapping between input images and their corresponding embeddings. However, in the case of NLU, the discrete mapping of tokens to embeddings makes the token reconstruction from continuous increments in the embedding space challenging. We thus formulate a discrete optimization attack, in which the unknown tokens are eventually represented by a one-hot like vector of the vocabulary length. The token in the vocabulary with the highest softmax activation is expected to be the unknown token of the canary. We demonstrate that in our attack's best configuration, for canaries of type *"my pin code is* $k_1k_2k_3k_4$*"*, $k_i \in \{0, 1, \ldots, 9\}, 1 \leq i \leq 4$, we are able to extract the numeric pin $k_1k_2k_3k_4$ with an accuracy of 0.5 (a lower bound on this accuracy using a naive random guessing strategy for a combination of four digits equals 1×10^{-4}).

[1] Following the terminology in Carlini et al. (2018)

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 552 - 560

Since we present a new application of ModIvA to NLU models, defenses against them are an important ethical consideration to prevent harm and are explored in Section 6. We observe that standard training practices commonly used to regularize NLU models successfully thwart this attack.

2 Related Work

Significant research has been conducted in the field of privacy-preserving machine learning. Shokri et al. (2017) determine whether a particular datapoint belongs to the training set $\boldsymbol{X_{tr}}$. The success of such attacks has prompted research in investigating them (Truex et al., 2019; Hayes et al., 2017; Song and Shmatikov, 2019). Carlini et al. (2018) propose the quantification of unintended memorization in deep networks and presents an extraction algorithm for data that is memorized by generative models. Memorization is further exploited in Carlini et al. (2020) where instances in the training data of very large language-models are extracted by sampling the model. The attacks described above are closed-box in nature where the adversary does not cast the attack as an optimization problem but instead queries the model multiple times.

Open-box ModIvA were initially demonstrated on a linear-regression model (Fredrikson et al., 2014) for inferring medical information. It has been extended to computer vision tasks such as facial recognition (Fredrikson et al., 2015) or image classification (Basu et al., 2019). Our work is a first attempt at performing ModIvAs on NLP tasks.

3 Attack Setup

We consider an NLU model $\boldsymbol{f_\theta}$ that takes an utterance $\boldsymbol{x}$ as input and uses the word-embeddings $\boldsymbol{E(x)}$ for the tokens in $\boldsymbol{x}$ to perform a joint intent classification (IC) and named-entity recognition (NER) task. We assume an adversary with open-box access to $\boldsymbol{f_\theta}$, which means that they are aware of the model architecture, trained parameters $\boldsymbol{\theta}$, loss function $\boldsymbol{L(f_\theta(E(x)), y)}$, label set $\boldsymbol{Y}$ of intents and entities supported by the model and vocabulary $\boldsymbol{V}$ which is obtained from the word-embeddings matrix $\boldsymbol{W} \in \mathbb{R}^{|V| \times d}$. However, the adversary does not have access to the training data $\boldsymbol{X_{tr}}$ used to train $\boldsymbol{f_\theta}$. The adversary's goal is to reconstruct a (private) subset $\hat{\boldsymbol{x}} \subseteq \boldsymbol{X_{tr}}$.

To perform a CEA on $\boldsymbol{f_\theta}$, we keep the parameters $\boldsymbol{\theta}$ fixed and minimize the loss function $\boldsymbol{L}$ with respect to the unknown inputs (i.e., tokens) of a given utterance. This is analogous to a traditional learning problem, except with fixed model parameters and a learnable input space. In this work, we use the NLU model architecture described in Section 4.1.

3.1 Canary Extraction Attacks

We consider a canary sentence $\boldsymbol{x_c} = (\boldsymbol{x_p}, \boldsymbol{x_u})$, $\boldsymbol{x_c} \in \boldsymbol{X_{tr}}$ with tokens $(p_1, .., p_m, u_1.., u_n)$ and output label $\boldsymbol{y_c} \in \boldsymbol{Y}$. The first m tokens in $\boldsymbol{x_c}$ represent a known prefix $\boldsymbol{x_p}$ (e.g."my pin code is") and the next n tokens $(u_1, .., u_n)$ represent the unknown tokens that an attacker is interested in reconstructing $\boldsymbol{x_u}$ (e.g."one two three four").
We represent the set of word embeddings of this canary $\boldsymbol{E(x_c)}$ as $\boldsymbol{(e_{p_1}, .., e_{p_m}, e'_{u_1}, .., e'_{u_n})}$.

A trivial attack to identify the n unknown tokens in $\boldsymbol{x_u}$ is by directly optimizing $\boldsymbol{L(f_\theta(E(x_c)), y_c)}$ over $\boldsymbol{(e'_{u_1}, .., e'_{u_n})}$, where $\boldsymbol{(e'_{u_1}, .., e'_{u_n})}$ are randomly initialized. Words corresponding to the optimized values of $\boldsymbol{(e'_{u_1}, .., e'_{u_n})}$ are then assigned by identifying the closest vectors in the embedding matrix $\boldsymbol{W}$ using a distance metric (e.g. Euclidean distance). However, our experiments demonstrate that this strategy is not successful since the updates are performed in a non-discrete fashion, whereas the model $\boldsymbol{f_\theta}$ has a discrete input space. We thus focus on performing a discrete optimization, inspired by works on relaxing categorical variables to facilitate efficient gradient flow (Jang et al., 2016; Song and Raghunathan, 2020), as illustrated in Figure 1.

We define a logit vector $\boldsymbol{z_i} \in \mathbb{R}^{|V|}$ for each token $u_i \in \boldsymbol{x_u}$. We then apply a softmax activation with temperature T to obtain $\boldsymbol{a_i} \in \mathbb{R}^{|V|}$:

$$a_{i,v} = \frac{e^{\frac{z_{i,v}}{T}}}{\sum_{j=1}^{|V|} e^{\frac{z_{j,v}}{T}}} \quad \text{for v =1, 2, ...,}|V| \qquad (1)$$

$\boldsymbol{a_i}$ is a differentiable approximation of the arg-max over the logit vector for low values of T. This vector then selectively attends to the tokens in the embedding matrix, $\boldsymbol{W} \in \mathbb{R}^{|V| \times d}$, resulting in the embeddings $\boldsymbol{(e'_{u_1}, .., e'_{u_n})}$ used as inputs fed to the model during the attack:

$$\boldsymbol{e'_{u_i}} = \boldsymbol{W^T} \cdot \boldsymbol{a_i} \quad \text{for } 1 \leq i \leq n \qquad (2)$$

We then train our attack and optimize for $\boldsymbol{Z} \in \mathbb{R}^{n \times |V|}$, with $\boldsymbol{Z} = (\boldsymbol{z_1}, \ldots, \boldsymbol{z_n})$:

$$\hat{\boldsymbol{Z}} = \arg\min_{\boldsymbol{Z}} \boldsymbol{L(f_\theta(E(x_c)), y_c)} \qquad (3)$$

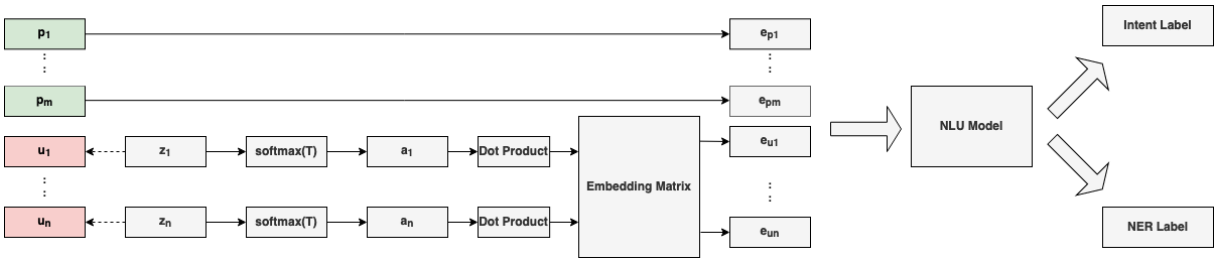

Figure 1: CEA using discrete optimization. The logit vectors $z_1, \ldots, z_n$ are optimized keeping the parameters of the NLU model $\boldsymbol{f_\theta}$ fixed. The unknown tokens $u_i, \ldots, u_n$ are then reconstructed using the logit vectors.

$\boldsymbol{Z}$ is the only trainable parameter in the attack and all parameters of $\boldsymbol{f_\theta}$ remain fixed. Once converged, we identify the token x_i as the one with the highest activation in $\boldsymbol{a_i}$. We decrease the temperature T exponentially to ensure low values of T in Equation (1) and enforce the inputs to $\boldsymbol{f_\theta}$ to be discrete. In our experiments, we define $\boldsymbol{z_i}$ over a subset of candidate words for $\boldsymbol{x_u}$ $V_0, V_0 \subseteq V$ to prevent the logit vector from becoming too sparse.

4 Experiments

4.1 Target Model Description

We attack an NLU model jointly trained to perform IC and NER tagging. This model has a CLC structure (Ma and Hovy, 2016). The input embeddings lead to 2 bi-LSTM layers and a fully-connected layer with softmax activation for the IC task and a Conditional Random Field (CRF) layer for the NER task. The sum of the respective cross-entropy and CRF loss is minimized during training. We use FastText embeddings (Mikolov et al., 2018) as inputs to our model[2].

4.2 Canary Insertion

We inject R repetitions of a single canary with sensitive information and its corresponding intent and NER labels into the training set of the NLU model. We insert three different types of canaries with n unknown tokens, $n \in \{4, 6, 8, 10\}$, described in Table 1. $\mathcal{C}$ is a set of 12 colors[3]. Additional details of the canaries and their output labels are presented in the Appendix A. The adversary aims to reconstruct all the n unknown, sensitive tokens in the canary. The reduced vocabulary V_0 in Equation (1) is the set of all digits for canary *call* and *pin* and the names of 12 colors for canary *color*.

[2] https://fasttext.cc/docs/en/english-vectors.html

[3] $\mathcal{C}$ = {'red', 'green', 'lilac', 'blue', 'yellow', 'brown', 'cyan', 'magenta', 'orange', 'pink', 'purple', 'mauve'}

Canary Pattern	$\{p_1, ..p_m, \underline{u_1.., u_n}\}$	Unknown tokens set
call	call $\underline{k_1 \ldots k_n}$	$k_i \in \{0, \ldots, 9\}, 1 \leq i \leq n$
pin	my pin code is $\underline{k_1 \ldots k_n}$	$k_i \in \{0, \ldots, 9\}, 1 \leq i \leq n$
color	color $\underline{k_1 \ldots k_n}$	$k_i \in C, 1 \leq i \leq n$

Table 1: Patterns of canaries injected into the dataset. Each token of interest k_i is randomly chosen from the corresponding token set.

4.3 Attack Evaluation

We inject the canary into Snips (Coucke et al., 2018), ATIS (Dahl et al., 1994) and NLU-Evaluation (Xingkun Liu and Rieser, 2019). The canary is repeated with $R \in \{1, 10, 100, 500\}$. For each combination of R, canary type and length n, the experiment is repeated 10 times (trials) with 10 different canaries, to account for variation induced by canary selection. We define the following evaluation metrics averaged across all trials to evaluate the strength of our attack.

Average Accuracy (Acc): Fraction of the trials where the attack correctly reconstructs the *entire* canary sequence in the correct order. A higher Accuracy indicates better reconstruction. Accuracy is 1 if we can reconstruct all n tokens in each of the 10 trials.

Average Hamming Distance per Token (HDT): The Hamming Distance (HD) (Hamming, 1950) is the number of positions at which the reconstructed utterance sequence is different from the inserted canary. Since HD is proportional to the length of the canary, we normalize it by the length of the unknown utterance ($HDT = HD/n$). The HDT can be interpreted as the probability of reconstructing the incorrect token for a given position in the canary, averaged across the 10 trials. A lower HDT indicates better reconstruction.

Accuracy reports our performance on reconstructing *all* n unknown tokens in the correct order and is a conservative metric. HDT quantifies our average performance for reconstructing each po-

Canary	n	R	Attack		Baseline	
			↑Acc	↓HDT	↑Acc	↓HDT
color	4	10	0.40	0.30	4.82e−5	0.92
	6	100	0.30	0.45	3.35e−7	
	8	100	0.10	0.60	2.33e−9	
	10	500	0.00	0.59	1.62e−11	
pin	4	500	0.40	0.27	1e−4	0.90
	6	100	0.10	0.45	1e−6	
	8	100	0.00	0.61	1e−8	
	10	100	0.10	0.43	1e−10	
call	4	10	0.30	0.40	1e−4	0.90
	6	100	0.20	0.50	1e−6	
	8	100	0.00	0.60	1e−8	
	10	500	0.00	0.59	1e−10	

Table 2: Best observed performance metrics for canaries with n unknown tokens and (R) repetitions.

sition in the unknown sequence. We evaluate our attack against randomly choosing a token from the reduced vocabulary V_0. Thus for a given value of n, the expected accuracy and HDT of this baseline are $(\frac{1}{|V_0|})^n$ and $1 - \frac{1}{|V_0|}$ respectively.

5 Results

The trivial attack described in Sec3.1 without discrete optimization performs comparably to the random selection baseline. We thus focus on performing the attack with discrete optimization in this Section. Table 2 shows the best reconstruction metrics for the different values of n and the corresponding repetitions $R \in \{10, 100, 500\}$ at which these metrics are observed in the Snips dataset. In our experiments, our attack consistently outperforms the baseline. For $n = 4, 6$, we reconstruct at least one complete canary for each pattern. The attack also completely reconstructs a 10-digit *pin* for higher values of R, with an accuracy of 0.10. Even when we are unable to reconstruct *every* token in any trial, i.e. accuracy is zero, we still outperform the baseline, as observed from the HDT values.

For the sake of brevity, we summarize the attack performance on other datasets in Appendix C.2. We observe that the attack is dataset-dependent with best performance for the Snips dataset and poorest for the NLU-evaluation dataset.

5.1 Discussion

The training data of NLU models may potentially contain sensitive utterances such as *"call* $k_1 \dots k_{10}$*"*, $k_{1 \le i \le 10} \in \{0, 1, \dots, 9\}$. An adversary who wishes to extract the phone-number can assume the prefix *"call"*, along with the output labels of the utterance which are also trivial to guess, given access to the label set Y. Our canaries act as a placeholder for such utterances. We choose to insert the canary *color* since the names of colors appear infrequently in the datasets mentioned in Section 4.3, allowing us to evaluate the attack on *'out-of-distribution'* data which is more likely to be memorized by deep networks (Carlini et al., 2018).

For $n = 4$ and $R = 1$ (i.e., the canary only appears once in the train set), our attack has an accuracy of 0.33 for canary *color* and 0.10 for *pin*. This suggests that the attack could potentially reconstruct sensitive information from short rare utterances in real-world scenarios. For a special case when the adversary attempts to reconstruct a ten digit phone-number in canary *call* with a three digit area-code of their choosing, the attack can reconstruct the remaining seven digits of the number with an accuracy of 0.1 when $R = 1$. For conciseness, we show these results in Appendix C.1. We observe that our model is more effective and with fewer repeats for the canary *color* than canaries *pin* and *call* of the same length. Our empirical analysis indicates the attack is more successful in extracting tokens that are relatively infrequent in the training data and in reconstructing shorter canaries. As shown in Appendix C.1, the attack performs best for $R = 1000$. However, this trend of improved reconstruction for larger values of R is not monotonic and we observe a general decline in reconstruction for $R > 1000$. We are unsure of the vulnerabilities that facilitate CEA. While unintended memorization is a likely explanation, we note that our attack performs best on the Snips data, although the smaller ATIS data should be easier to memorize (Zhang et al., 2016).

6 Proposed Defenses against ModIvA

We propose three commonly used modeling techniques as defense mechanisms- Dropout (D), Early Stopping (ES) (Arpit et al., 2017) and including a Character Embeddings layer in the NLU model (CE). D and ES are regularization techniques to reduce memorization and overfitting. CE makes the problem in 3 more difficult to optimize, by concatenating the embeddings of each input token with a character level representation. This character level representation is obtained using a convolution layer on the input sentence (Ma and Hovy, 2016).

For defense using D, we use a dropout of 20% and 10% while training the NLU model. For ES, we stop training the NLU model under attack if the

validation loss does not decrease for 20 consecutive epochs to prevent over-training.

6.1 Efficacy of Defenses

In this section we present the performance of the proposed defenses against ModIvA. To do so, we evaluate the attack on NLU models trained with each defense mechanism individually, and in all combinations. The canaries are inserted into the Snips dataset and repeated 10, 500 and 1000 times. The results are summarized in Table 3. We observe that the attack accuracy for each defense (used individually and in combination) is nearly zero for all canaries and is thus omitted in the table. We also note that the HDT approaches the random baseline for most defense mechanisms. The attack performance is comparable to a random-guess when the three mechanisms are combined. However, when dropout or character embedding is used alone, HDT values are lower than the baseline, indicating the importance of combining multiple defense mechanisms. Additionally, training with defenses do not have any significant impact on the performance of the NLU model under attack. The defenses thus successfully thwart the proposed attack without impacting the performance of the NLU models.

R	Defense Mechanism	↓HDT		
		Color	Pin	Call
	Baseline	0.916	0.90	0.90
10	**No defense**	**0.30**	**0.33**	**0.40**
	Dropout (D)	0.85	0.80	0.76
	Early Stopping (ES)	0.80	0.93	0.95
	Char. Emb. (CE)	0.65	0.75	0.90
	D + ES	0.98	0.90	0.95
	ES + CE	0.90	0.83	0.90
	D + ES + CE	0.90	0.90	0.90
500	**No defense**	**0.39**	**0.27**	**0.38**
	Dropout (D)	0.65	0.54	0.83
	Early Stopping (ES)	0.85	1.00	0.75
	Char. Emb. (CE)	0.58	0.93	0.68
	D + ES	0.85	0.93	0.98
	ES + CE	0.93	0.98	0.78
	D + ES + CE	0.95	0.88	1.00
1000	**No defense**	**0.35**	**0.18**	**0.48**
	Dropout (D)	0.35	0.78	0.58
	Early Stopping (ES)	0.90	0.83	0.85
	Char. Emb. (CE)	0.70	0.68	0.78
	D + ES	0.88	0.98	0.90
	ES + CE	0.88	1.00	0.95
	D + ES + CE	0.95	0.93	0.95

Table 3: Attack performance for the canary *color*, *pin* and *call* after incorporating defenses while training the target NLU model, with $R \in \{10, 500, 1000\}$.

7 Conclusion

We formulate and present the first open-box ModIvA in a form of a CEA to perform text completion on NLU tasks. Our attack performs discrete optimization to select unknown tokens by optimizing over a set of continuous variables. We demonstrate our attack on three patterns of canaries and reconstruct their unknown tokens by significantly outperforming the 'chance' baseline.

To ensure that the proposed attack is not misused by an adversary, we propose training NLU models with three commonplace modelling practices–dropout, early-stopping and including character level embeddings. We observe that the above practices are successful in defending against the attack as its accuracy and HDT values approach the random baseline. Future directions include *'demystifying'* such attacks, and strengthening the attack for longer sequences with fewer repeats and a larger V_0 and investigating additional defense mechanisms, such as those based on differential privacy, and their effect on the model performance.

8 Ethical Considerations

The addition of proprietary data to existing datasets to fine-tune NLU models can often insert confidential information into datasets. The proposed attack could be misused to extract private information from such datasets by an adversary with open-box access to the model. The objectives of this work are to (1) study and document the actual vulnerability of NLU models against this attack, which shares similarities with existing approaches (Fredrikson et al., 2014; Song and Raghunathan, 2020); (2) warn NLU researchers against the possibility of such attacks; and (3) propose effective defense mechanisms to avoid misuse and help NLU researchers protect their models.

Our work demonstrates that private information such as phone-numbers and zip-codes can be extracted from a discriminative text-based model, and not only from generative models as previously demonstrated (Carlini et al., 2020). We advocate for the necessity to privatize such data using anonymization (Ghinita et al., 2007) or differential privacy (Feyisetan et al., 2020). Additionally, in case the training data continues to contain some private information, practitioners can prevent the extraction of sensitive data by using the defense mechanisms described in Section 6, which reduces the attack performance to a random guess.

References

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR.

Samyadeep Basu, Rauf Izmailov, and Chris Mesterharm. 2019. Membership model inversion attacks for deep networks. *arXiv preprint arXiv:1910.04257*.

Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. 2018. The secret sharer: Evaluating and testing unintended memorization in neural networks. *arXiv preprint arXiv:1802.08232*.

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

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, pages 12–16.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriber. 1994. Expanding the scope of the atis task: The atis-3 corpus. *Proceedings of the workshop on Human Language Technology*, pages 43–48.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 178–186.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333.

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32.

Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. 2007. Fast data anonymization with low information loss. In *Proceedings of the 33rd international conference on Very large data bases*, pages 758–769.

Richard W Hamming. 1950. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160.

Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2017. Logan: evaluating privacy leakage of generative models using generative adversarial networks. *arXiv preprint arXiv:1705.07663*.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: the view from here. *natural language engineering*, 7(4):275.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074. Association for Computational Linguistics.

Klaus Macherey, Franz Josef Och, and Hermann Ney. 2001. Natural language understanding using statistical machine translation. In *Seventh European Conference on Speech Communication and Technology*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753. IEEE.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. *arXiv preprint arXiv:2004.00053*.

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.

Oguzhan Tas and Farzad Kiyani. 2007. A survey automatic text summarization. *PressAcademia Procedia*, 5(1):205–213.

Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2019. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*.

Pawel Swietojanski Xingkun Liu, Arash Eshghi and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, pages xxx–xxx, Ortigia, Siracusa (SR), Italy. Springer.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

A Inserted Canary Information

The inserted canaries and corresponding intent and NER label sets are listed below.

1. Canary *call*: "call $k_1 \dots k_n$", $k_i \in \{0, 1, \dots, 9\}$, for $1 \leq i \leq n$.
   - Sequence NER tags: "O B-canary $\underbrace{\text{I-canary} \dots \text{I-canary}}_{n-1 \text{ times}}$"
   - Intent: "CallIntent"
2. Canary 2: "my pin code is $k_1 \dots k_n$", k_i, for $1 \leq i \leq n$.
   - Sequence NER tags: "O O O O B-canary $\underbrace{\text{I-canary} \dots \text{I-canary}}_{n-1 \text{ times}}$"
   - Intent: "PinIntent"
3. Canary 3: "color $k_1 \dots k_n$", $k_i \in$ {'red', 'green', 'lilac', 'blue', 'yellow', 'brown', 'cyan', 'magenta', 'orange', 'pink', 'purple', 'mauve'} for $1 \leq i \leq n$.
   - Sequence NER tags: "O B-canary $\underbrace{\text{I-canary} \dots \text{I-canary}}_{n-1 \text{ times}}$"
   - Intent: "ColorIntent"

The canary repetitions R are split between the train and validation set in a ratio of $9 : 1$.

B Training Parameters

We decrease the temperature T exponentially after each iteration t. The temperature at the t^{th} iteration T_t is given by $T_t = 0.997^t \times 10^{-1}$.

We use the Adam optimizer and train our attack for 250 epochs. We begin with an initial learning rate of 6.5×10^{-3} for our attack with a decay rate of 9.95×10^{-1}.

C Results

C.1 Attack Performance Across Canary Repetitions

Table 4 shows the model performance for just one repeat of the canary in the Snips dataset i.e. $R = 1$. The $n = 7$ example for the *call* canary refers to the special case when the adversary is trying to reconstruct a 10-digit phone number beginning with a three digit area code of their choice.

Table 5 illustrates the best reconstruction metrics for different values on n and with $R \in \{10, 100, 500, 1000, 2000\}$. We observe an accuracy of 0.5 for the canary *pin* when $n = 4$ and $R = 1000$. Figure 2 illustrates the model performance across canaries in the Snips dataset with varying number of repetitions R. As observed in Table 5 and Figure 2, the attack is most likely to succeed when R is 1000. However, the attack weakens for higher values of R.

n	Canary	Attack Metrics		Baseline Metrics	
		Accuracy	HDT	Accuracy	HDT
4	color	0.33	0.43	4.8×10^{-5}	0.92
4	pin	0.10	0.60	1×10^{-4}	0.90
4	call	0.10	0.58	1×10^{-4}	0.90
10	call	0.00	0.68	1×10^{-10}	0.90
7	call	0.10	0.70	1×10^{-7}	0.90

Table 4: Reconstruction metrics for inserted utterances appearing only *once* in the training data, i.e $R = 1$. The attack accuracy is much higher and HDT is much lower than that of a randomly chosen sequence of tokens.

Canary	n	R	Attack		Baseline	
			↑Acc	↓HDT	↑Acc	↓HDT
color	4	10	0.40	0.30	4.82e−5	0.92
	6	100	0.30	0.45	3.35e−7	
	8	1000	0.10	0.48	2.33e−9	
	10	1000	0.00	0.59	1.62e−11	
pin	4	1000	0.50	0.18	1e−4	0.90
	6	1000	0.10	0.43	1e−6	
	8	1000	0.00	0.57	1e−8	
	10	100	0.10	0.43	1e−10	
call	4	10	0.30	0.40	1e−4	0.90
	6	100	0.20	0.50	1e−6	
	8	1000	0.00	0.58	1e−8	
	10	2000	0.00	0.59	1e−10	

Table 5: Best observed performance metrics for canaries with n unknown tokens and $R \in \{10, 100, 500, 1000, 2000\}$.

C.2 Attack Performance Across Datasets

We evaluate our attack on the ATIS and NLU-Evaluation Datasets, for canaries *color* and *pin* with $n = 4$ and canary *call* with $n = 10$. To ensure that we maintain a comparable number or repeats with respect to the size of the dataset, $R \in \{10, 100, 200, 500\}$ for the ATIS dataset and $R \in \{100, 500, 1000, 5000, 10000\}$ for the NLU-Evaluation dataset. As shown in Figure 3, the attack performance is almost comparable for shorter sequences in Snips and ATIS but under-performs for the NLU-Evaluation data. Figure 4 and Figure 5 illustrate the HDT for the ATIS and NLU Evaluation datasets for R canary repetitions respectively.

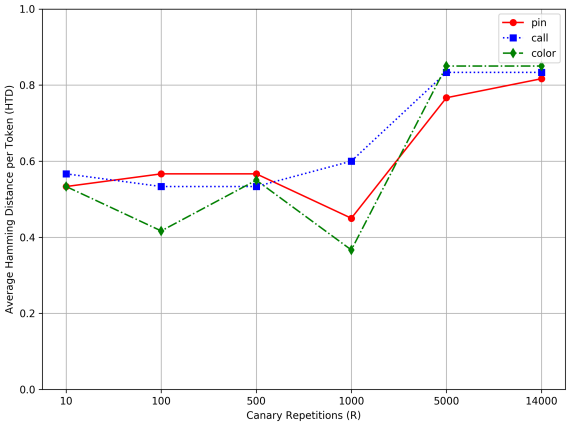

Figure 2: Average Hamming Distance per Token (HDT) for canaries with $n = 6$, repeated in the Snips dataset R times.

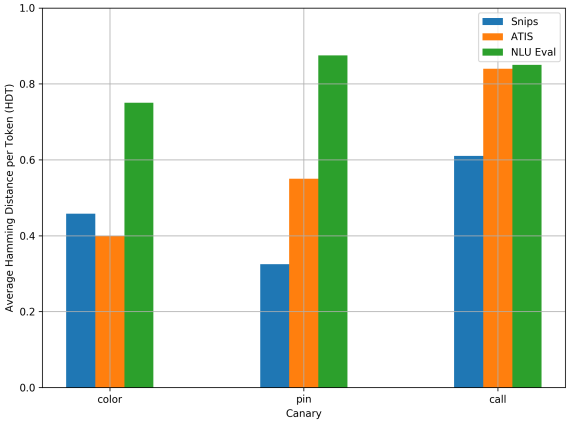

Figure 3: Model Performance of the *pin* and *color* canary with $n = 4$ and *call* canary with $n = 10$, for the Snips, ATIS, and NLU Evaluation Data.

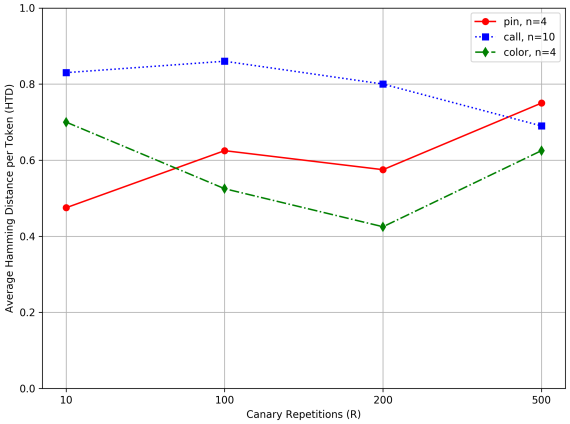

Figure 4: Model Performance of the *pin* and *color* canary with $n = 4$ and *call* canary with $n = 10$, repeated R times in the ATIS dataset.

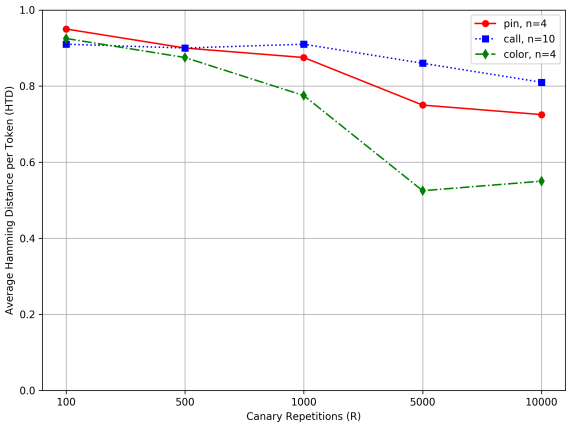

Figure 5: Model Performance of the *pin* and *color* canary with $n = 4$ and *call* canary with $n = 10$, repeated R times in the NLU Evaluation dataset.