BLACK BOX ATTACKS ON TRANSFORMER LANGUAGE MODELS

Vedant Misra
HubSpot
25 First St.
Cambridge, MA, USA
vedant@hubspot.com

ABSTRACT

Language models based on Transformers have proven remarkably effective at producing human-quality text. While such models enable users to quickly generate coherent long-form passages, they are also prone to leaking information about the individual data records used for training. In this work we first implement a Transformer-based architecture for dialog completion. We next investigate the susceptibility of this model to membership inference attack under varying conditions of data and information availability. We show how to train attack models to infer membership in a Transformer language model target even in a query-limited black box setting.

1 INTRODUCTION

Language models have made rapid progress toward producing human quality text. Generative unsupervised pretraining (Radford et al. (2018), Howard & Ruder (2018)) with Transformer models (Vaswani et al. (2017)) can achieve state of the art results after little or even no task-specific tuning (Devlin et al. (2018), Radford et al. (2018), Radford et al. (2019)).

However, because systems developed with such models directly generate and expose text to the user, they invite attacks from adversaries seeking to extract information from or reverse engineer them (Fredrikson et al. (2015), Tramèr et al. (2016), Carlini et al. (2018), Shokri & Shmatikov (2015)).

One method to achieve this is via membership inference attack (Shokri et al. (2016), Hayes et al. (2017)), which considers the scenario in which an adversarial user of such a black box prediction service can provide input messages resembling those of a competitor, and based on the model's output extract information from the model about the user. This information includes whether the competitor's message data was used to train the model, as well as message content from the competitor's private message corpus. Various methods have been proposed to measure the degree to which an adversary can achieve this (Carlini et al. (2018), Tramèr et al. (2016)).

In this work, we train a Transformer-based language model to predict likely message completions given a partial customer message as input. Then, we investigate the susceptibility of the Transformer language model to membership inference attack, specifically under varying conditions of data sparsity, and show that such models are susceptible to attack even in a query-limited black box setting.

2 RELATED WORK

Some existing products that help users reply to messages implement scoring over pre-selected candidates (Kannan et al. (2016), Henderson et al. (2017)). Optimizing for a language modeling objective instead enables the model to dynamically adapt to user input, as in Google *Smart Compose* (Lambert (2018)).

Furthermore, existing products for helping customers reply to messages use LSTM-RNNs (Kannan et al. (2016), Henderson et al. (2017), Lambert (2018)). Such models are efficient at inference, but are believed to be limited in their ability to remember long-range dependencies (Tang et al. (2018)) compared to models based on self-attention instead of recurrence.

3 TRANSFORMER LANGUAGE MODEL

Our implementation leverages self-attention mechanisms and is trained on a language modeling objective. We follow the model architecture and training setup implemented in GPT1 (Radford et al. (2018)), which feeds a context vector of tokens $D = (d_{-k}, ..., d_{-1})$ to a 12-layer stack of Transformer decoders:

$$\begin{aligned} h_0 &= DW_e + W_p \\ h_l &= \texttt{transformer\_decoder}(h_{l-1}) \forall i \in [1, 12] \\ P(d) &= \texttt{softmax}(h_n W_e^T) \end{aligned} \tag{1}$$

where W_e is the token embedding matrix and W_p is the position embedding matrix. We use 12 attention heads and set the input and output dimension of each transformer block d_{model} to 768. We further set $d_{\text{vocab}} = 40734$ and set $k = 256$.

We initialize our implementation to weights learned from the BookCorpus dataset (Zhu et al. (2015)) for the GPT1 model. Given a corpus $\mathcal{D} = \{d_1, ..., d_N\}$ of 10M HubSpot customer chat messages we optimize parameters Θ via unsupervised fine tuning to maximize the language modeling likelihood

$$\mathcal{L}_{\text{LM}}(\mathcal{D}) = \sum_i \log P(d_i | d_{i-k}, ..., d_{i-1}; \Theta).$$

Messages were pre-processed to select agent responses to incoming customer messages. They were further filtered to include English-only messages longer than 5 tokens. We conducted experiments using 8 V100 GPUs.

As shown in Table 1, our model predicts full response completions that capture the gestalt of the target corpus without requiring us to explicitly constrain the space of candidate responses or implement diversity-based sampling.

Input sample	Predicted completion
hi there	, i 'm sorry for the delay .
how can we	help you ?
what is	your email address ?
do you have	a website ?
can we	have your email address ?
the best way to	get a response is to send a message to our support team .
you can find	the link to the page on our website .
there are	no limits on the number of users you can have .
your website has	a great amount of information on our website .

Table 1: Sample dialog completions. Examples shown are generated via multinomial sampling and truncated at the first appearance of terminal punctuation.

4 BLACK BOX ATTACK

4.1 THREAT MODEL

Because modern web applications leverage rate limiting on APIs that inhibit any attacker's ability to quickly gather data, our threat model considers primarily the black box query-limited environment, where the attacker does not have access to model weights or hyper-parameters, and can collect only a limited sample of data from an oracle (Papernot et al. (2016a), Ilyas et al. (2018), Papernot et al. (2016b)). We assume this oracle f_{oracle} was trained via multi-party collaborative learning such that one participant is an adversary.

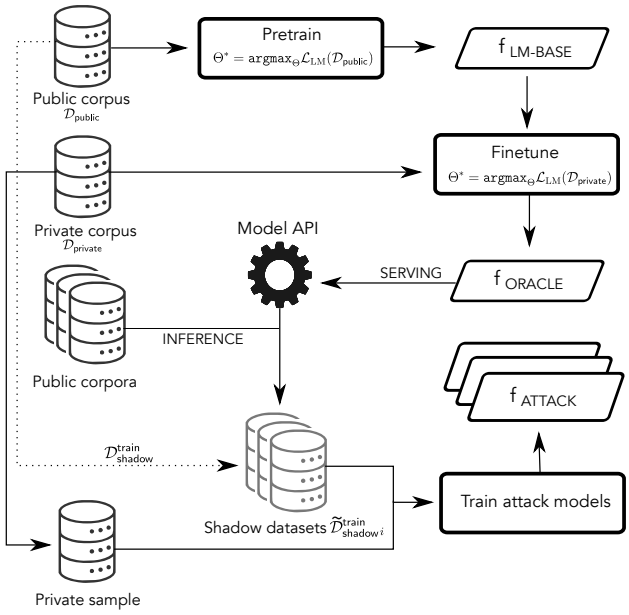

Figure 1: Black box membership inference attack procedure for a trained language model behind a service API

4.2 MEMBERSHIP INFERENCE ATTACK

Membership inference is the problem of assessing given a model and a data record whether that record was used in the training set of the model (Shokri et al. (2016), Hayes et al. (2017)). We focus on the case of inferring the membership of a sample of customer message data in the training set of a language model. Given our target language model f_{oracle} fine tuned from $f_{\text{lm-base}}$ and a sample of the private corpus $\mathcal{D}^{\text{test}}_{\text{target}}$ of tokens $\{d_1, ..., d_N\}_{\text{test}}$, we generate shadow datasets (Shokri et al. (2016)) $\mathcal{D}^{\text{train}}_{\text{shadow}^i}$ and $\tilde{\mathcal{D}}^{\text{train}}_{\text{shadow}^i}$ distributed similarly to $\mathcal{D}^{\text{train}}_{\text{target}}$ with the goal of assessing the utility of publicly available corpora for training shadow models in the black box setting, as illustrated in Figure 1. We sample $\tilde{\mathcal{D}}^{\text{train}}_{\text{shadow}^i}$ from public corpora (Zhu et al. (2015), Danescu-Niculescu-Mizil & Lee (2011)), fixing $|\tilde{\mathcal{D}}^{\text{train}}_{\text{shadow}^i}|$ and $|\mathcal{D}^{\text{train}}_{\text{shadow}^i}| \in \{1000, 5000, 10000, 50000\}$. We sample $\mathcal{D}^{\text{train}}_{\text{shadow}^i}$ from $\mathcal{D}^{\text{test}}_{\text{target}}$, a portion of the HubSpot message corpus held out during language modeling.

4.3 ATTACK SIMULATION

In the black box environment, we assume an adversary does not have any real training data or statistics about the distribution of that data. As such, we implement model-based synthesis under the assumption that the adversary has access to the prediction vector of a machine learning API service.

We require $\mathcal{D}^{\text{train}}_{\text{shadow}^i} \cap \mathcal{D}^{\text{train}}_{\text{target}^i} = \emptyset \; \forall i$ to simulate the environment in which an adversary might sample from a private message corpus, however, this is not strictly necessary (Shokri et al. (2016)). Given a sample $\{d_1, d_2, ..., d_n\} \in \mathcal{D}^{\text{train}}_{\text{shadow}} \cup \tilde{\mathcal{D}}^{\text{train}}_{\text{shadow}}$ we set $x^{\text{train}}_{\text{shadow}} = d_1, \ldots, d_{\lfloor n/2 \rfloor}$, requiring $n \geq 8$. This implicitly sets $y_{\text{shadow}} = d_{\lfloor n/2 \rfloor + 1}$, for which we collect the prediction vector $y^{\text{train}}_{\text{shadow}} = f_{\text{oracle}}(x^{\text{train}}_{\text{shadow}})$ via service requests to the machine learning API. This is equivalently $\texttt{softmax}(h_n W_e^T) = P(d_{\lfloor n/2 \rfloor + 1}) \in \mathbb{R}^{d_{\text{vocab}}}$.

Algorithm 1 BLACK BOX MEMBERSHIP INFERENCE ATTACK

Input: A trained model f_{oracle}
Output: $\tilde{\mathcal{D}}^{\text{train}}_{\text{shadow}^i}, \mathcal{D}^{\text{train}}_{\text{shadow}^i}, f_{\text{attack}\,i}$
$f_{\text{attack}} \leftarrow \emptyset$
$c \leftarrow \mid \{\mathcal{D}_{\text{public}\,i}\} \mid$
for $i \in \{1, ...c\}$ **do**
  $\tilde{\mathcal{D}}^{\text{train}}_{\text{shadow}} \leftarrow \text{SAMPLE}(\mathcal{D}^{\text{train}}_{\text{public}\,i})$
  $\mathcal{D}^{\text{train}}_{\text{shadow}} \leftarrow \text{SAMPLE}(\mathcal{D}^{\text{train}}_{\text{target}})$
  $\mathcal{X}^{\text{train}}_{\text{shadow}} \leftarrow \{\, \text{d}_1, \ldots, d_{\lfloor n/2 \rfloor}\} \in \tilde{\mathcal{D}}^{\text{train}}_{\text{shadow}} \cup \mathcal{D}^{\text{train}}_{\text{shadow}}$
  $y^{\text{train}}_{\text{shadow}} \leftarrow \text{f}_{\text{oracle}}(\mathcal{X}^{\text{train}}_{\text{shadow}})$
  $y^{\text{train}}_{\text{attack}} \leftarrow \mathbf{1}_{\mathcal{D}^{\text{train}}_{\text{shadow}}}(\mathcal{X}^{\text{train}}_{\text{shadow}})$
  $f_{\text{attack}} \leftarrow \text{TRAINATTACKMODEL}(\mathcal{X}^{\text{train}}_{\text{shadow}}, y^{\text{train}}_{\text{attack}})$

As indicated in algorithm 1, we then train attack models f_{attack} to differentiate the distributions generating $\mathcal{D}^{\text{train}}_{\text{shadow}^i}$ from those generating $\tilde{\mathcal{D}}^{\text{train}}_{\text{shadow}^i}$. We do this by first projecting the prediction vectors in our shadow datasets to a space of dimensionality $\lfloor \sqrt{d_{\text{vocab}}} \rfloor$ via SVD. We then provide this denser representation as input to models that learn f_{attack}. We investigate the performance of various attack model architectures under varying conditions of data sparsity.

As baseline models we consider random forest and a feedforward neural network with hidden layers of sizes $(100, 50, 25)$, relu non-linearities, and a sigmoid head. We learn the function f_{attack}, which classifies samples from the f_{shadow_i}, by optimizing a binary cross-entropy objective using the Adam optimizer.

As shown in Table 2, the attack models begin to learn a signal even in conditions of data sparsity. We further see that it is not necessary for the adversary to know which data were used to train $f_{\text{lm-base}}$. This indicates that even without access to white-box implementation details, it is possible to extract information from a transformer language model via membership inference attack.

	$\tilde{\mathcal{D}}^{\text{shadow}}_{\text{train}} \sim \mathcal{D}^{\text{LM-Base}}_{\text{train}}$		$\tilde{\mathcal{D}}^{\text{shadow}}_{\text{train}} \sim \mathcal{D}^{\text{CornellDialog}}_{\text{train}}$	
Target Size	DNN-FF	RF	DNN-FF	RF
1000	**0.638**	0.623	0.562	**0.600**
5000	**0.690**	0.677	0.615	**0.616**
10000	**0.700**	0.668	**0.644**	0.636
50000	**0.694**	0.676	**0.635**	0.623

Table 2: Attack model AUC ROC

5 CONCLUSION

We implement shadow training to perform a membership inference attack on a Transformer-based language model, specifically investigating the relationship between attack model performance, data availability, and shadow dataset hypotheses. Future work should consider larger and more complex language models that implement conditional control or leverage multi-phased pretraining curricula. It should also aim to study the size of the attack surface by considering additional shadow models, and performing architecture search across attack models.

REFERENCES

Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *arXiv preprint arXiv:1802.08232*, 2018.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

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

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

Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. 2017.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply, 2017.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information, 2018.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufman, Balint Miklos, Greg Corrado, Andrew Tomkins, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. Smart reply: Automated response suggestion for email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2016).*, 2016. URL `https://arxiv.org/pdf/1606.04870v1.pdf`.

Paul Lambert. Subject: Write emails faster with smart compose in gmail, May 2018. URL `https://www.blog.google/products/gmail/subject-write-emails-faster-smart-compose-gmail/`.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, 2016a.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning, 2016b.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321. ACM, 2015.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2016.

Gongbo Tang, Mathias Mller, Annette Rios, and Rico Sennrich. Why self-attention? a targeted evaluation of neural machine translation architectures, 2018.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pp. 601–618, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, 2015.