Domain Private Transformers for Multi-Domain Dialog Systems

Anmol Kabra [1] Ethan R. Elenberg [2]

[1]TTI Chicago, anmol@ttic.edu [2]ASAPP, eelenberg@asapp.com

Safe generation in multi-domain dialogs

Large Language Models perform well when little training data is available for new domains. Through transfer learning, one can finetune a pretrained model on new data from specialized domains. This leads to a novel privacy scenario:

When prompted with text from one domain, contextual language models should not generate sensitive text of other domains.

Say a company trains dialog generative models on a dataset pooled from many clients. Two clients in `AIRLINE` and `INSURANCE` industries contribute data in exchange for a generative model. Each client desires that the model not generate their sensitive (proprietary) text when prompted with text of other clients. Clients are interested in safe generation.

Tools needed to measure safe generation

Let $\{d_1, \ldots, d_N\}$ be domains from which datasets are created. Let M_D be a model trained on dataset D. For each domain d_i, we can prompt M_D with contexts $\{c_i\}$ from domain d_i. We then check if M_D generates sensitive text of domains d_j for $j \neq i$.

Hence, we need (1) a tool to tell us if a text token is sensitive for a domain, and (2) a procedure to check if a model generates sensitive tokens.

Policy functions

Previous work by Shi et al. (2022a) introduced policy functions to mark sensitive tokens.

Let $\tau_n = (t_1, \ldots, t_n)$ be a sequence of tokens. A policy function F annotates τ with 0-1 labels; $F(\tau_n)_i = 1$ if the i^{th} token is sensitive and 0 if not. For each domain $d_i \in \{d_1, \ldots, d_n\}$, we use a policy F_i that marks a sequence with sensitive tokens from d_i.

Membership inference attacks

Carlini et al. (2021, 2022) introduced membership inference attacks to check if models leaked training data.

We use Likelihood Ratio (LiRa) membership inference attacks to prompt models and check if the generated text contains sensitive text for a domain. We compare the leakage of a **target** with a **reference** model as follows:

1. Prompt target model with contexts $\{c_i\}$ to generate text $\{x_i\}$.
2. Select $\{x_i\}$ most likely to be generated by **target** w.r.t. **reference**.
3. If selected x_i contains sensitive text of other domains, then **target** model leaks (success).

We can compare several **target** models by success rate:

LiRa attack success rate = #success / #non-empty-generations

Defining safe generation as Domain Privacy

The goal of domain privacy is to estimate how likely a **target** model M_D trained on dataset D generates sensitive text of domain d_j when prompted with text from domain d_i, where $j \neq i$.

To check if text contains sensitive tokens of domain d_j, we can use a policy function F_j. We can empirically estimate domain privacy using LiRa inference attacks.

Domains d_i and d_j could have inherent overlap, e.g. `AIRLINE` and `INSURANCE` text overlapping due to air travel insurance. So, use $M_{\overline{D}_j}$ as a **reference** model, where $\overline{D}_j = D \setminus d_j$ is the dataset obtained by removing text of domain d_j from D. We can compare the leakage of M_D w.r.t. $M_{\overline{D}_j}$ on the domain d_j.

D and $\overline{D}_j$ are neighbors **at domain level** w.r.t. F_j as they differ in one domain.

Note: Domain privacy only cares about cross-domain leakage. So, domain privacy captures the need for safe generation: **inter-domain private generation** and **intra-domain public generation**.

Domain Privacy

Let $C > 0$ be a parameter. A model M_D is C-domain-private for D, if for all $i, j \in [N]$ where $j \neq i$, contexts $\{c_i\}$ from domain d_i,

$$\Pr[M_D(c_i) \in d_j] \leq C \cdot \Pr\left[M_{\overline{D}_j}(c_i) \in d_j\right].$$

Methodology

We compare the domain privacy of several *target* models in dialog generation, for two policy functions.

Two policies for experiments

1. **Keyword Detection** policy marks a token as sensitive if the token is in a hand-crafted list of keywords. We create a list for each domain, and thus a policy for each domain.
2. **Sequence Classification** policy marks a token as sensitive if a finetuned RoBERTa model classifies the token as belonging to a domain.

Target models for experiments

Using both policies, we create a **redacted** version of D for each domain. Then we experiment with these target models:

1. `DOMAIN`$_i$ Only. Baseline target, finetuned only on $D \cap d_i$ **non-redacted** data.
2. Public. Finetuned on **non-redacted** data with AdamW optimizer.
3. Pub+Redacted. Finetuned on **redacted** data with AdamW optimizer.
4. Private. Finetuned on **non-redacted** with DP-AdamW optimizer (Differentially-Private).
5. JFT. "Just Fine-tune Twice" procedure: finetune on **redacted**, then use weights to initialize and finetune on **non-redacted** Shi et al. (2022b).
6. Ours: Redaction Schedule. Initially finetune on **redacted** and gradually transition to **non-redacted** – transition according to probability p, controlled by a decaying schedule.

Experiments

Setup

MultiDoGo dataset consists of task-oriented dialogs of user-agent customer service simulation from 6 domains. We use the 3 largest domains: `AIRLINE` , `MEDIA` , and `INSURANCE` .

We finetune a pretrained GPT2 checkpoint on data from all 3 domains.

We conduct LiRa attacks on each target model to test for domain privacy. Here we give results for domain `AIRLINE` . Into each model, we feed 100 prompts from the `AIRLINE` domain and generate 10 different outputs for each prompt. We compare LiRa attack success rates and test set perplexity for target models.

Results

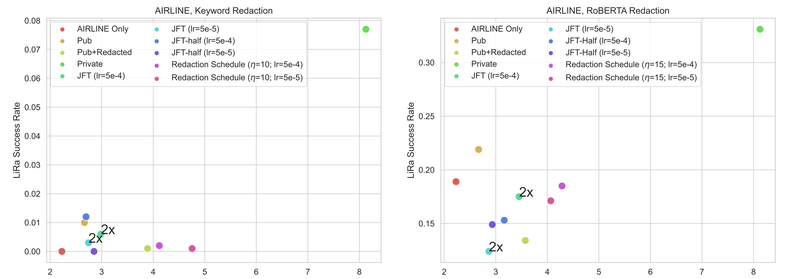

Figure 1. LiRa success rate vs test perplexity. Lower is better for both axes. 2x indicates double training cost.

- LiRa attacks are more successful w.r.t. RoBERTa policy compared to the keyword, because the former has higher recall and lower precision.
- For RoBERTa, all but Private and Public have LiRa success rate lower than the `AIRLINE` Only baseline.
- While having comparable domain privacy, JFT has better perplexity and Redaction Schedule has worse perplexity compared to Pub+Redacted.
- Vanilla finetuning like Public is insufficient for domain privacy.
- Domain privacy becomes feasible when models finetuned on redacted datasets (at least partially).

References

Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. Selective differential privacy for langauage modeling. In *NAACL*, pages 2848–2859, 2022a.

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 *USENIX Security Symposium*, pages 2633–2650, 2021. ISBN 978-1-939133-24-3.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

Weiyan Shi, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. Just fine-tune twice: Selective differential privacy for large language models. In *EMNLP*, pages 6327–6340, 2022b.