Deep Reinforcement Learning-based Text Anonymization against Private-Attribute Inference

Ahmadreza Mosallanezhad **Ghazaleh Beigi** **Huan Liu**
Computer Science and Engineering
Arizona State University
{amosalla, gbeigi, huanliu}@asu.edu

Abstract

User-generated textual data is rich in content and has been used in many user behavioral modeling tasks. However, it could also leak user private-attribute information that they may not want to disclose such as age and location. User's privacy concerns mandate data publishers to protect privacy. One effective way is to anonymize the textual data. In this paper, we study the problem of textual data anonymization and propose a novel Reinforcement Learning-based Text Anonymizor, RLTA, which addresses the problem of private-attribute leakage while preserving the utility of textual data. Our approach first extracts a latent representation of the original text w.r.t. a given task, then leverages deep reinforcement learning to automatically learn an optimal strategy for manipulating text representations w.r.t. the received privacy and utility feedback. Experiments show the effectiveness of this approach in terms of preserving both privacy and utility.

1 Introduction

Social media users generate a tremendous amount of data such as profile information, network connections and online reviews and posts. Online vendors use this data to understand users preferences and further predict their future needs. However, user-generated data is rich in content and malicious attackers can infer users' sensitive information. AOL search data leak in 2006 is an example of privacy breaches which results in users re-identification according to the published AOL search logs and queries (Pass et al., 2006). Therefore, these privacy concerns mandate that data be anonymized before publishing. Recent research has shown that textual data alone may contain sufficient information about users' private-attributes that they do not want to disclose such as age, gender, location, political views and sexual orientation (Mukherjee and Liu, 2010; Volkova et al., 2015). Little attention has been paid to protect users textual information (Li et al., 2018; Zhang et al., 2018; Anandan et al., 2012; Saygin et al., 2006).

Anonymizing textual information comes at the cost of losing utility of data for future applications. Some existing work shows the degraded quality of textual information (Anandan et al., 2012; Zhang et al., 2018; Saygin et al., 2006). Another related problem setting is when the latent representation of the user generated texts is shared for different tasks. It is very common to use recurrent neural networks to create a representation of user generated text to use for different machine learning tasks. Hitaj el al. show text representations can leak users' private information such as location (Hitaj et al., 2017). This work aims to anonymize users' textual information against private-attribute inference attacks.

Adversarial learning is the state-of-the-art approach for creating a privacy preserving text embedding (Li et al., 2018; Coavoux et al., 2018). In these methods, a model is trained to create a text embedding, but we cannot control the privacy-utility balance. Recent success of reinforcement learning (RL) (Paulus et al., 2017; Sun and Zhang, 2018) shows a feasible alternative: by leveraging reinforcement learning, we can include feedback of attackers and utility in a reward function that allows for the control of the privacy-utility balance. Furthermore, an RL agent can perturb parts of an embedded text for preserving both utility and privacy, instead of retraining an embedding as in adversarial learning. Therefore, we propose a novel Reinforcement Learning-based Text Anonymizer, namely, RLTA, composed of two main components: 1) an attention based task-aware text representation learner to extract latent embedding representation of the original text's content w.r.t. a

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing, pages 2360–2369,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

given task, and 2) a deep reinforcement learning based privacy and utility preserver to convert the problem of text anonymization to a one-player game in which the agent's goal is to learn the optimal strategy for text embedding manipulation to satisfy both privacy and utility. The Deep Q-Learning algorithm is then used to train the agent capable of changing the text embedding w.r.t. the received feedback from the privacy and utility subcomponents.

We investigate the following challenges: 1) How could we extract the textual embedding w.r.t. a given task? 2) How could we perturb the extracted text embedding to ensure that user private-attribute information is obscured? and 3) How could we preserve the utility of text embedding during anonymization? Our main contributions are: (1) we study the problem of text anonymization by learning a reinforced task-aware text anonymizer, (2) we corporate a data-utility task-aware checker to ensure that the utility of textual embeddings is preserved w.r.t. a given task, and (3) we conduct experiments on real-world data to demonstrate the effectiveness of RLTA in an important natural language processing task.

2 Related Work

Reinforcement Learning (RL) has applications in natural language processing and recommendation systems. For example, a recent paper (Paulus et al., 2017) combines RL with a supervised method to get a readable and informative article summary. Another work uses RL to solve the problem of adversarial generative models for text generation (Shi et al., 2018). Sun et al. also uses RL in a recommendation system which recommends items according to the users' feedbacks and preferences (Sun and Zhang, 2018)

Textual data is rich in content and recent research has shown that users' private-attributes can be easily inferred from the text (Beretta et al., 2015; Mukherjee and Liu, 2010; Volkova et al., 2015), however, few papers consider user privacy w.r.t. such data. Anandan et al. (2012) introduce t-Plausibility which uses an information theoretic based approach to sanitize documents heuristically. This method does not preserves the utility of data during anonymization process. Another work focuses on leveraging differential privacy (Dwork et al., 2017) to make the extracted Term Frequency Inverse Document (TF-IDF) textual vectors private (Zhang et al., 2018). It has been shown that TF-IDF cannot accurately capture semantic meaning of the text which can hurt its usefulness for different tasks (Lan et al., 2005).

Two recent similar works Li et al. (2018); Coavoux et al. (2018) convert textual data anonymization into a minimax problem. These works use the idea of adversarial learning to create a text embedding which satisfies utility and protects users against private-attribute leakage. Our scenario is similar to the work of (Li et al., 2018) as they have considered several attribute-inference attackers as adversaries to create a privacy-preserving text embedding. (Beigi et al., 2019b,a) propose a method for privacy preserving text representation. In this method, they try to add noise to an existing text representation in a way that it does not change the meaning of the text and it preserves the user's private attributes.

Our work is different from the existing works. First, we consider the task that the textual information will be used for and protect users against leakage of private-attributes. Second, we incorporate deep RL to anonymize the extracted text embedding by receiving privacy and utility feedbacks and automatically learning the optimal strategy for proper manipulation of text embeddings.

3 Problem Statement

Let $\mathcal{X} = \{x_1, x_2, ..., x_N\}$ denotes a set of N documents and each document x_i is composed of a sequence of words. We denote $\mathbf{v}_i \in R^{d\times 1}$ as the embedded representation of the original document x_i. Let $\mathcal{P} = \{p_1, p_2, ..., p_m\}$ denotes a set of m private-attributes that users do not want to disclose such as age, gender, location, etc. The goal of reinforced task-aware text anonymizer is to learn an embedding representation of each document and then anonymize it such that 1) users privacy is preserved by preventing any potential attacker to infer users' private-attribute information from the textual embedding data, and 2) utility of the text embedding is maintained for a given task $\mathcal{T}$ which incorporates such data, e.g., classification. In this paper, we study the following problem:

Problem 3.1. Given a set of documents $\mathcal{X}$, set of private-attributes $\mathcal{P}$, and given task $\mathcal{T}$, learn an anonymizer f that can learn a private embedded representation $\mathbf{v}_i$ from the original document x_i so that, 1) the adversary cannot infer the targeted user's private-attributes $\mathcal{P}$ from the private

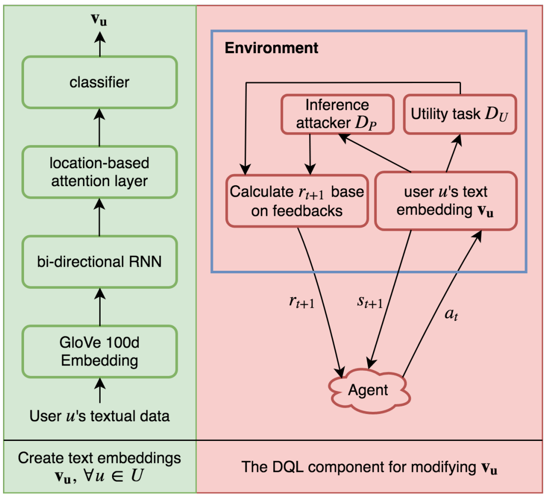

$\mathbf{v_u}$
classifier
location-based attention layer
bi-directional RNN
GloVe 100d Embedding
User u's textual data
Create text embeddings $\mathbf{v_u}$, $\forall u \in U$
Environment
Inference attacker D_P
Utility task D_U
Calculate r_{t+1} base on feedbacks
user u's text embedding $\mathbf{v_u}$
r_{t+1}
s_{t+1}
a_t
Agent
The DQL component for modifying $\mathbf{v_u}$

Figure 1: The architecture of RLTA method

text representation $\mathbf{v}_i$, and 2) the generated private representation $\mathbf{v}_i$ is good for the given task $\mathcal{T}$. The problem can be formally defined as:

$$\mathbf{v}_i = f(x_i, \mathcal{P}, \mathcal{T}) \tag{1}$$

Due to the success of Reinforcement Learning (Shi et al., 2018; Paulus et al., 2017), we use RL to address the aforementioned problem. RL (Sutton and Barto, 2018) formulates the problem within the framework of Markov Decision Process (MDP), and learns an action-selection policy based on past observations of transition data. An MDP is defined by state space $S = \{s\}$, action space $A = \{a\}$, transition probability function $P : S \times A \times S \to [0, 1]$ and reward function $r : S \times A \times S \to R$.

4 Proposed Method

We discuss the reinforced task-aware text anonymizer framework. The input of this private system is the user generated text, and the output is a privacy-preserving text representation. As in Figure. 1, this framework consists of two major components: 1) an attention based task-aware text representation learner, and 2) a deep RL based privacy and utility preserver. The text representation learner aims to extract the embedded representation of a document w.r.t. a given task by minimizing the task's loss function. Then, the deep RL preserver manipulates the embedded text representation by learning the optimal strategy so that both privacy and utility of the embedded representation are preserved. It includes two sub-components: 1) private-attribute inference attacker D_P, and 2) data-utility task-aware checker D_U. The former seeks to infer user private-attribute information based on their embedded text representation. The latter incorporates the given manipulated embedded text representation for a given task $\mathcal{T}$ and investigates the usefulness of the latent representation for $\mathcal{T}$.

The RL component then utilizes the feedback of the two sub-components to guide the data manipulation process by ensuring that the new text embedding does not leak user private-attributes by confusing the adversary in D_P and the changes made to the representation does not destroy the semantic meaning for $\mathcal{T}$.

4.1 Extracting Textual Embedding

Let $x = \{w_1, ..., w_m\}$ be a document with m words. Attention mechanism has shown to be effective in capturing embedding of textual information w.r.t. a given task (Pennington et al.; Vaswani et al., 2017). It allows the model to attend to different parts of the given original document at each step and then learns what to attend based on the input document and what it has produced as embedding representation so far, as shown in Figure. 1.

We use a bi-directional recurrent neural network (RNN) to encode the given document into an initial embedding representation. RNN has been shown to be effective for summarizing and learning semantic of unstructured noisy short texts (Cho et al., 2014; Shang et al., 2015). We use GloVe 100d (Pennington et al.) to exchange each word w_i with its corresponding word vector, note that different dimensionality can be used. This process produces a matrix of text $x' \in \mathcal{R}^{m*100}$.

We employ the gated recurrent unit (GRU) as the cell type to build the RNN, which is designed in a manner to have a more persisted memory (Cho et al., 2014). The bi-directional GRU will read the text forward and backwards, then outputs two hidden states $\mathbf{h_t^{fw}}, \mathbf{h_t^{bw}}$ and an output $\mathbf{o_t}$. We then concatenate two hidden states as the initial encoded embedding of the given original document:

$$\mathbf{H_t} = Concat(\mathbf{h_t^{fw}}, \mathbf{h_t^{bw}}) \tag{2}$$

After calculating the initial context vector $\mathbf{H_t}$, we seek to pinpoint specific information within the $\mathbf{H_t}$, which helps the classifier to predict the labels with higher confidence (Luong et al., 2015) We use the location-based attention layer based on the work of Luong et al. (2015). The attention layer calculates a vector $\mathbf{a_t}$ including a weight for each element in the $\mathbf{H_t}$, showing the importance of that element. The context vector $\mathbf{v_t}$ is calculated:

$$\mathbf{v_t} = \sum_{i=1}^{m} a_{t,i}\mathbf{H_i} \tag{3}$$

The vector $\mathbf{v_t}$ is then fed to a neural network classifier for the given utility task. Classification is one of the common tasks for textual data. Based on the output of the classifier and loss function, we update the three networks so that the output of the attention layer is an useful context that can be used for a utility task (Ranzato et al., 2015).

4.2 Reinforced Task-Aware Text Anonymizer

Here, we discuss the details of the second component which seeks to preserve privacy and utility.

4.2.1 Protecting Private-Attributes

Textual information is rich in content and publishing textual embedding representation without proper anonymization leads to privacy breach and revealing the private-attributes of an individual such age, gender and location. It is thus essential to protect the textual information before publishing it. The goal of our model is to manipulated learned embedded representation such that any potential adversary cannot infer users' private-attribute information. However, a challenge is that the text anonymizer does not know the adversary's attack model. To address this challenge, we add a private-attribute inference attacker D_P sub-component to our text anonymizer. This sub-component learns a classifier that can accurately identify the private information of users from their embedded text representations $\mathbf{v_u}$. We incorporate this sub-component to understand how the textual embedded representation should be anonymized to obfuscate the private information.

Inspired by the success of RL (Kaelbling et al., 1996; Mnih et al., 2013; Van Hasselt et al., 2016), we model this problem using RL to automatically learn how to anonymize the text representations w.r.t. the private-attribute inference attacker. In our RL model, one agent is trained to change a randomly selected text embedding representation. Then, the agent keeps interacting with the environment and changes the text embedding accordingly based on its current state and received rewards so that the private-attribute inference attacker cannot correctly identify user's private-attribute information given his embedding. In this part, we define the main four parts of RL environment in our problem, i.e., environment, state, action and reward.

- **Environment:** Environment in our problem includes the private-attribute inference attackers D_P and the text embedding $\mathbf{v_u}$. Note that D_P is trained beforehand.
- **State:** State describes the current situation. Here, state is the current text embedding vector $\mathbf{v_{u,t}}$ which reflects the results of the agents' actions on $\mathbf{v_u}$ up to time t.
- **Actions:** Action is define as selecting one element such as $v_{u,k}$ in text embedding vector $\mathbf{v_u} = \{v_{u,1}, ..., v_{u,m}\}$ and changing it to a value near -1, 0 or 1. This results in $3.m$ actions where m is the size of the embedding vector.

  Changing value to near *1*: In this action, the agent changes the value of $v_{u,k}$ to a value between 0.9 to 1.0. As $v_{u,k}$ will be multiplied by a classifier's weight, the output will be the weight as is. In another word, the value $v_{u,k}$ will become important to the classifier.

  Changing value to near *0*: In this action the $v_{u,k}$ will be changed to a value between -0.01 to 0.01. This action makes $v_{u,k}$ seem neutral and unimportant to a classifier as it will result in a 0 when multiplied by a weight.

  Changing value to near *-1*: In this action, the agent changes $v_{u,k}$ to a value between -1.0 to -0.9. This action will make $v_{u,k}$ important to a classifier, but, in a negative way.
- **Reward:** Reward in our problem is defined based on how successfully the agent obfuscated the private-attribute information against the attacker so far. In particular, we defined the reward function at state s_{t+1} according to the confidence of private-attribute inference attacker C_{p_k} for private-attribute p_k given the resultant text embedding at state s_{t+1}, i.e., $\mathbf{v_{t+1}}$. Considering the classifier's input data as $\mathbf{v_u}$ and its correct label as i, we define the confidence for a multi-class classifier as the difference between the probability of actual value of the private-attribute and the minimum probability of other values of the private-attribute:

  $$C_{p_k} = Pr(l = i|\mathbf{v_u}) - \min_j Pr(l = j|\mathbf{v_u}) \tag{4}$$

  Where l indicates label. For each private-attribute attacker p_k, the confidence score C_{p_k} is within the range $[-1, 1]$. Positive value demonstrates that the attacker has predicted private-attribute accurately, and negative value indicates that the attacker was not able to infer user's private-attribute. According to this definition, the reward will be positive if action a_t has caused information hiding, and will be negative if the action a_t was not able to hide sensitive information. Having confidence of private-attribute inference attackers, reward function at state s_{t+1} is defined as:

$$r_{t+1}(s_{t+1}) = -\sum_{p_k \in D_P} C_{p_k}(s_{t+1}) \tag{5}$$

The reward r_t is calculated according to the state s_{t+1} which associated with the transition of agent from state s_t after applying action a_t. Note that the goal of agent is to maximize the amount of received rewards so that the mean of rewards r over time $t \in [0, T]$ (T is the terminal time) will be positive and above 0.

4.2.2 Preserving Utility of Text Embedding

Thus far, we have discussed how to 1) learn textual embeddings from the given original document w.r.t. the given task, and 2) prevent leakage of private-attribute information by developing a reinforcement learning environment which incorporates a private-attribute inference attacker and manipulates the initial given text embedding accordingly to fool the attacker. However, data obfuscation comes at the cost of data utility loss. Utility is defined as the quality of the given data for a given task. Neglecting the utility of the text embedding while manipulating it, may destroy the semantic meaning of the text data for the given task. Classification is one of the common tasks. In order to preserve the utility of data, we need to ensure that preserving privacy of data does not destroy the semantic meaning of the text embedding representation w.r.t. the given task. We approach this challenge by changing the agent's reward function w.r.t. the data utility. We add a utility sub-component, i.e., classifier D_U, to the reinforcement learning environment which its goal is to assess the quality of resultant embedding representation. We use the confidence of the classifier for the given task to measure the utility of embedding representation using the text embedding vector $\mathbf{v_u}$ the its correct label i.

$$C = Pr(l = i|\mathbf{v_u}) - \min_j Pr(l = j|\mathbf{v_u}) \tag{6}$$

The agent can then use the feedback from the utility classifier to make decision when taking actions. We thus modify the reward function in order to incorporate the confidence of utility sub-component. Reward function at state s_{t+1} can be defined as:

$$\begin{aligned} r_{t+1}(s_{t+1}) =& \alpha C_{D_U}(s_{t+1}) - \\ & -(1-\alpha) \sum_{p_k \in D_P} C_{p_k}(s_{t+1}) - \mathcal{B} \end{aligned} \tag{7}$$

where C_{D_U} and C_{p_k} represent the confidence of utility sub-component and private-attribute inference attacker, respectively. Moreover, $\mathcal{B}$ demonstrates a baseline reward which forces the agent to reach a minimum reward value. The coefficient α also control the amount of contribution from both private-attribute inference and utility sub-components in the Eq. 7.

4.3 Optimization Algorithm

Given the formulation of states and actions, we aim to learn the optimal strategy via manipulating text representations w.r.t. the private-attribute attackers and utility sub-component feedbacks. We manipulate the text embeddings by repeatedly choosing an action a_t given current state s_t, and then applying actions on current state to transit to the new one s_{t+1}. The agent then receives reward r_{t+1} as a consequence of interacting with the environment. The goal of agent is to manipulate text embedding $v_{u,k}$ in a way that maximizes its reward according to Eq. 7. Moreover, the agent updates its action selection policy $\pi(s)$ so that it can achieve the maximum reward over time.

In RLTA we use Deep Q-Learning which is a variant of Q-Learning. In this algorithm the goal is to find the following function:

$$Q^*(s_t, a_t) = \mathrm{E}_{s_{t+1}}[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a')] \tag{8}$$

where $Q(s, a)$ corresponds to the Q-function for extracting actions and it is defined as the expected return based on state s and action a. Moreover, $Q^*(s, a)$ denotes the optimal action-value Q-function which has the maximum expected return using the optimal policy $\pi(s)$. Rewards are also discounted by a factor of γ per time step. The agent keeps interacting with the environment till it reaches the terminal time T.

Since it is not feasible to estimate $Q^*(s, a)$ in Eq.8, we use a function approximator to estimate the state-action value function $Q^*(s, a) \approx$

Algorithm 1 The Learning Process of RLTA

```
Require: $\mathbf{v}$, $D_P$, $D_U$, $\alpha$, $\gamma$, $\mathcal{B}$, $T$.
Initialize replay memory $M$ with size $N$
while training is not terminal do
    $s_t \leftarrow \mathbf{v}$
    for $t \in \{0, 1, ..., T\}$ do
        Choose action $a_t$ using $\epsilon$-greedy
        Perform $a_t$ on $s_t$ and get $(s_{t+1}, r_{t+1})$
        $M \leftarrow M + (s_t, a_t, r_{t+1}, s_{t+1})$
        $s_t \leftarrow s_{t+1}$
        Sample mini-batch $b$ from memroy $M$
        for $(s, a, s', r) \in b$ do
            Update DQN weights using Eq. 11
        end for
    end for
end while
```

$Q(s, a; \theta)$. Given neural networks as excellent function approximators (Cybenko, 1989), we lverage a deep neural network function approximator with parameters θ, or a Deep Q-Network (DQN) (Mnih et al., 2013) by minimizing the following:

$$L(\theta) = \mathrm{E}_{s_t, a_t, r_{t+1}, s_{t+1}}[(y - Q(s, a; \theta))^2] \quad (9)$$

in which y is the target for the current iteration:

$$y = \mathrm{E}_{s_{t+1}}[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^p)] \quad (10)$$

θ^p is the parameters from the previous iteration.

We update the DQN according to the derivation of Eq. 9 with respect to the parameter θ:

$$\begin{aligned} \nabla_\theta L(\theta) = & \mathrm{E}_{s_t, a_t, r_{t+1}, s_{t+1}}[(r \\ & + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^p) \\ & - Q(s_t, a_t; \theta)) \nabla_\theta Q(s_t, a_t; \theta)] \end{aligned} \quad (11)$$

Algorithm 1 shows the optimization process.

5 Experiments

Experiments are designed to answer the following questions: **Q1**(*Privacy*): How well RLTA can obscure users' private-attribute information? **Q2**(*Utility*): How well RLTA can preserve utility of the textual data w.r.t. the given task? **Q3**(*Privacy-Utility Relation*): How does improving user privacy affects loss of utility?

To answer the first question (**Q1**), we investigate the robustness of resultant text embedding against private-attribute inference attacks. We consider two private-attribute information, i.e., location and gender. To answer the second question (**Q2**), we report experimental results w.r.t. a well-known task, sentiment analysis. Sentiment analysis has many applications in user-behavioral modeling and Web (Zafarani et al., 2014). In particular, we predict sentiment of the given textual embedding. To answer the final question (**Q3**), we examine the privacy improvement against utility loss.

5.1 Data

We use a real-world dataset from Trustpilot (Hovy et al., 2015). This dataset includes user reviews along with users private-attribute information such as location and gender. We remove non-English reviews based on LANGID.py[1] (Lui and Baldwin, 2012) and only keep reviews classified as English. Then, we consider English reviews associated with location of US and UK and create a subset of data with $10k$ users. Each review is associated with a rating score. We consider the review's sentiment as positive if its rating score is $\{4, 5\}$ and consider it as negative if rating is $\{1, 2, 3\}$

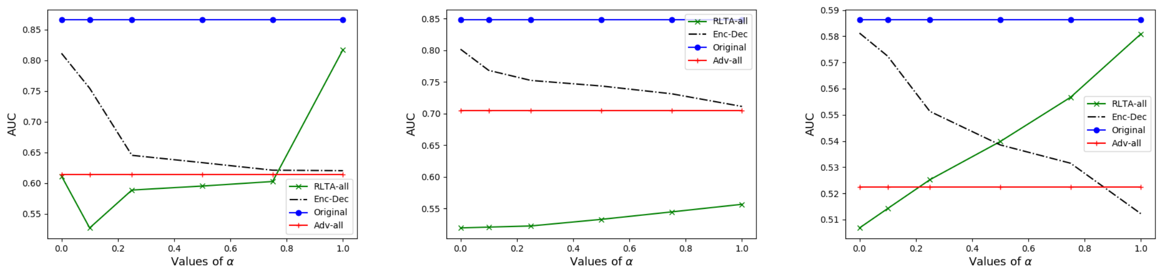
5.2 Implementation Details

For extracting the initial textual embedding, we use a bi-directional RNNs which their hidden sizes are set to 25. This makes the size of the final hidden vector $\mathbf{H_t}$ as 50. We also use a logistic regression with a linear network as the classifier in the attention mechanism. We use a 3-layer network for the Deep Q-network, i.e., input, hidden and output layers. Dimensions of the input and hidden layers are set to 50 and 700, respectively. Dimension of the last layer, i.e., output, is also set as 150. This layer outputs the state-action values which we execute the action with the best value.

For each of the private-attribute attackers and utility sub-components, we use feed-forward network with a single hidden layer with dimension of 100 which gets the textual embedding as input and uses a $Softmax$ function as output.

We first train both private-attribute inference attacker D_P and utility sub-component D_U on the training set. These sub-components do not change after that. Then, we train an agent on each selected data for 5000 episodes. The reward discount for agents is $\gamma = 0.99$ and batch size $b = 32$. We also set the terminal time $T = 25$. We run RLTA for 5 times and select the best agent based on the cumulative reward. We also vary α as $\alpha = \{0, 0.25, 0.5, 0.75, 1\}$. The higher values of

[1] https://github.com/saffsd/langid.py



(a) Gender inference attack **(b) Location inference attack** **(c) Sentiment prediction**

Figure 2: AUC scores for private-attribute and sentiment prediction tasks for different values of α. Lower AUC for private-attribute inference attacks shows higher privacy, while higher AUC for the sentiment prediction task indicates higher utility.

α indicate more utility contribution in RLTA.

5.3 Experimental Design

We use 10-fold cross validation of RLTA for evaluating both private-attribute inference attacker and an utility task with the following baselines:

- **ORIGINAL:** This baseline is a variant of proposed RLTA which does not change the original user text embeddings $\mathbf{v_u}$ and publishes it as is.
- **ADV-ALL:** This adversarial method has two main components, i.e., generator and discriminator, and creates a text representation that has high quality for a given task, but has poor quality for inferring private-attributes (Li et al., 2018).
- **ENC-DEC:** Using an auto-encoder is one of the effective methods to create a text embedding (Nallapati et al., 2016). We modify this simple method to create a privacy-preserving text embedding. This method gets the original text x and outputs a re-constructed text $\bar{x}$. The following loss function is used to train the model. After training, we use the encoder's output as the text representation $\mathbf{v_u}$ (Cho et al., 2014).

$$loss = -\sum_{x \in X} \log Pr(\bar{x}|x) + \\ + \alpha((\sum_{p_k} C_{p_k}) - C_{D_U}) \tag{12}$$

In which α is the privacy budget.

To examine the privacy of final text embedding, we apply the trained private-attribute attacker sub-component D_P to the output of each method to evaluate the users' privacy. We consider two private attributes, i.e., location and gender. We then compute the attacker's AUC. Lower attacker's AUC indicates that textual embeddings have higher privacy after anonymization against the private-attribute inference attacker. We also report experimental results w.r.t. the utility. In particular, we predict sentiment (positive and negative) of the given textual embedding by applying trained utility sub-component D_U to the resultant text embedding from test set for each method. We then compute AUC score for sentiment prediction task. Higher values of AUC demonstrate that the utility of textual embedding has been preserved.

5.4 Experimental Results

We answer the three question **Q1**, **Q2** and **Q3** to evaluate our proposed method RLTA.We use a natural language processing task, sentiment prediction, using a three layer neural network.

Privacy (Q1). Figure. 2 (a-b) demonstrates the results of private-attribute inference attack w.r.t. gender and location attributes. The lower the value of AUC is, the more privacy user has in terms of obscuring private attributes. We also report the performance of RLTA for different values of α.

We observe that ORIGINAL is not robust against private-attribute inference attack for both gender and location attributes. This confirms leakage of users private information from their textual data. Moreover, RLTA has significantly lower AUC score for both gender and location attributes in comparison to other methods. This demonstrates the effectiveness of RL for obfuscating private attributes. In RLTA, the AUC score for private-attribute inference attack increases for both attributes with the increase of α which shows the degradation in user privacy. The reason is because of the fact that agent pays less attention to privacy by increasing the value of α.

In the ENC-DEC method, as the value of α increases, the encoder tries to generate a text representation that is prune to inference attacks but it does not lose its utility w.r.t. the given task D_U. The results show that as α increases, the AUC of inference attackers will decrease.

Utility (Q2). To answer the second question, we investigate the utility of embeddings w.r.t. sentiment prediction. Results for different values of α are demonstrated in Figure. 2(c). The higher the value of the AUC is, the higher utility is preserved.

The Original approach has the highest AUC score which shows the utility of the text embeddings before any anonymization. We observe that the results for RLTA is comparable to the Original approach which shows that RLTA preserves the utility of text embedding. Moreover, RLTA outperforms Adv-all which confirms the effectiveness of reinforced task-aware text anonymization approach in preserving utility of the textual embeddings. We also observe that the AUC of RLTA w.r.t. sentiment prediction task increases with the increase of value of α. This is because with the increase of α, the agent pays more attention to the feedbacks of utility sub-component.

We also observe a small utility loss after applying RLTA when $\alpha = 1$. This is because the agent keeps changing the text embedding until it reaches the terminal time. These changes result in loss of utility even when the $\alpha = 1$.

Finally, in the Enc-Dec method, as both utility and attackers have the same importance, trying to preserving privacy would result in huge utility loss as we increase the value of α.

Privacy-Utility Relation (Q3). Results show that the Original achieves the highest AUC score for both utility task and private-attribute inference attack. This shows that Original has the highest utility which comes at the cost of significant user privacy loss. However, comparing results of privacy and utility for $\alpha = 0.5$, we observe RLTA has achieved the lowest AUC score for attribute inference attacks in comparison to other baselines, thus has the highest privacy. It also reaches the higher utility level in comparison to the Adv-all. RLTA also has comparable utility results to the Original approach. We also observe that increasing the α reduces the performance of RLTA in terms of privacy but increases its performance for utility. However, with $\alpha = 1$, RLTA preserves both user privacy and utility in comparison to Original, Enc-Dec, and Adv-all.

Method	Location	Gender	Utility
Original	84.77	86.54	58.57
Enc-Dec	71.55	58.35	53.78
Adv-all	70.37	57.15	52.15
RLTA	53.34	56.41	54.83
RLTA-gen	56.64	**55.02**	**56.67**
RLTA-loc	**52.04**	56.64	54.13

Table 1: Impact of different private-attribute inference attackers on RLTA when $\alpha = 0.5$. With $\alpha = 0.5$, privacy and utility will contribute equally.

5.4.1 Impact of Different Components

Here, we investigate the impact of different private-attribute inference attackers. We define two variants of our proposed model, RLTA-Gen and RLTA-Loc. In each of these variants, we train the agent in RLTA w.r.t. the one of private-attribute attackers, e.g., RLTA-Gen is trained to solely hide *gender* attribute. For this experiment we set $\alpha = 0.5$ as in this case privacy and utility sub-components contribute equally during training phase (Eq. 7). Results are shown in Table 1.

RLTA-Loc and RLTA-Gen have the best performance amongst all methods in obfuscating location and gender private-attributes, respectively. Results show that using RLTA-Loc could also help improve privacy on gender and likewise for (RLTA-Gen) in comparison to other approaches.

RLTA-Gen performs better in terms of utility, in comparison to RLTA which incorporates both gender and location attackers. Moreover, results show that both RLTA-Gen and RLTA-Loc have better utility than other baselines.

To sum-up, these results indicate that although using one private-attribute attacker in the training process can help in preserving more utility, it can compromise obscuring other private-attributes.

Parameter Analysis: Our proposed method RLTA has an important parameter α to change the level of privacy and utility. We illustrate the effect of this parameter by changing it as $\alpha \in \{0.0, 0.1, 0.25, 0.5, 0.75, 1.0\}$. According to the Figure 2, when the α parameter increases, the privacy loss will decrease, but, the utility loss will increase. This shows the utility and the privacy have an association with each other. Hence, the more privacy loss decreases, the utility loss increases. Choosing the right value for α depends on the application and usage of this method. According to the results, choosing $\alpha = 0.5$ would result in a balanced privacy-utility. In some applications where the privacy of users are important and critical, we can set the α parameter above 0.5.

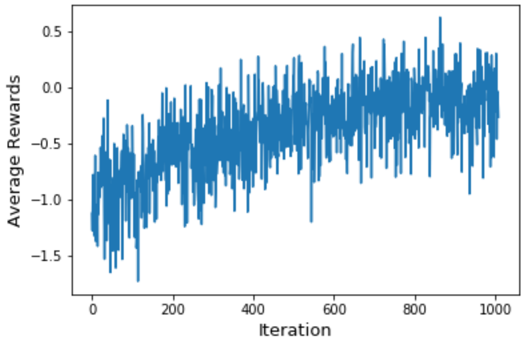


Figure 3: The average of agent's rewards

On the other hand, if the users privacy is not top priority, this parameter can be set to a lower value than 0.5 which although it does not protect users' private attribute as good as when $\alpha >= 0.5$, but it does protect users' private attribute at a reasonable level.

5.4.2 Rewards Convergence

To evaluate the convergence of rewards, we consider agent's reward during training phase for each episode, shown in Figure. 3. The result indicates that agent's average reward is low at the beginning and then it increases afterward. This is because agent performs many random actions at the beginning to explore the action state space. We also observe that after several episodes, the reward converges to the baseline reward $\mathcal{B}$. This confirms that the agent has learned a proper action selection policy $\pi(s)$ to preserve both utility and privacy by satisfying the objectives of Eq. 7.

6 Conclusion

In this paper, we propose a deep reinforcement learning based text anonymization, RLTA, which creates a text embedding such that does not leak user's private-attribute information while preserving its utility w.r.t. a given task. RLTA has two main components: (1) an attention based task-aware text representation learner, and (2) a deep RL based privacy and utility preserver. Our results illustrate the effectiveness of RLTA in preserving privacy and utility. One future direction is to generate privacy preserving text rather than embeddings. We also adopt deep Q-learning to train the agent. A future direction is to apply different RL algorithms and investigate how it impacts results. It would be also interesting to adopt RLTA for other types of data.

7 Acknowledgements

The authors would like to thank Jundong Li for his help throughout the paper. This material is based upon the work supported, in part, by NSF 1614576, ARO W911NF-15-1-0328 and ONR N00014-17-1-2605.

References

Balamurugan Anandan, Chris Clifton, Wei Jiang, Mummoorthy Murugesan, Pedro Pastrana-Camacho, and Luo Si. 2012. t-plausibility: Generalizing words to desensitize text. *Trans. Data Privacy*, 5(3):505–534.

Ghazaleh Beigi, Kai Shu, Ruocheng Guo, Suhang Wang, and Huan Liu. 2019a. I am not what i write: Privacy preserving text representation learning. *arXiv preprint arXiv:1907.03189*.

Ghazaleh Beigi, Kai Shu, Ruocheng Guo, Suhang Wang, and Huan Liu. 2019b. Privacy preserving text representation learning. *Proceedings of the 30th on Hypertext and Social Media (HT19). ACM*.

Valentina Beretta, Daniele Maccagnola, Timothy Cribbin, and Enza Messina. 2015. An interactive method for inferring demographic attributes in twitter. In *ACM Conference on Hypertext & Social Media*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Maximin Coavoux, Shashi Narayan, and Shay B Cohen. 2018. Privacy-preserving neural representations of text. *arXiv preprint arXiv:1808.09408*.

George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2017. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51.

Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. 2017. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618. ACM.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th International Conference on World Wide Web*.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*.

Man Lan, Sam-Yuan Sung, Hwee-Boon Low, and Chew-Lim Tan. 2005. A comparative study on term weighting schemes for text categorization. In *IEEE International Joint Conference on Neural Networks*.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. In *The 56th Annual Meeting of the Association for Computational Linguistics*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Arjun Mukherjee and Bing Liu. 2010. Improving gender classification of blog authors. In *Empirical Methods in natural Language Processing (EMNLP)*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *1st international conference on Scalable information systems (InfoScale)*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of conference on Empirical Methods in natural Language Processing (EMNLP)*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Yücel Saygin, Dilek Hakkini-Tur, and Gökhan Tur. 2006. Sanitization and anonymization of document repositories. In *Web and information security*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2018. Toward diverse text generation with inverse reinforcement learning. In *27th International Joint Conference on Artificial Intelligence*.

Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *ACM SIGIR Conference on Research & Development in Information Retrieval*.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *30th AAAI Conference*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Svitlana Volkova, Yoram Bachrach, Michael Armstrong, and Vijay Sharma. 2015. Inferring latent user properties from texts published in social media. In *29th AAAI Conference on Artificial Intelligence*.

Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. 2014. *Social media mining: an introduction*. Cambridge University Press.

Jinxue Zhang, Jingchao Sun, Rui Zhang, Yanchao Zhang, and Xia Hu. 2018. Privacy-preserving social media data outsourcing. In *IEEE INFOCOM Conference on Computer Communications*.