Turning Attacks into Protection: Social Media Privacy Protection Using Adversarial Attacks

Xiaoting Li[*] Lingwei Chen[*] Dinghao Wu[*†]

Abstract

Machine learning, especially deep learning, has emerged as one of the most powerful tools for attribute inference attacks over social media, which poses serious threats to users' privacy and security. In this paper, we explore a novel perspective of protecting data privacy in social media, where we take advantage of the vulnerability of machine learning, and introduce adversarial attacks to forge latent feature representations and mislead attribute inference attacks. Considering that text data in social media shares the most significant privacy of users, we investigate how text-space adversarial attacks can be elaborated to obfuscate users' attributes, and accordingly present a text-space *adversarial attack as defense*, or *AaaD* for short. Specifically, we advance AaaD by constructing semantically and visually similar word candidates to perturb, and leveraging word importance scores as selection probabilities to upgrade a population-based optimization to expedite adversarial text generation. We evaluate the performance of AaaD on two social media data sets, while the experimental results validate its effectiveness against inference attacks. Our work yields great value and unveils a new insight on the applicability of adversarial attacks for attribute obfuscation and privacy protection.

1 Background and Motivation

Social media has been enjoying explosive growth for a decade, while its worldwide accessibility has drastically reshaped the world that allows billions of people all around the globe to conveniently perform numerous activities such as creating online profiles, sharing personal posts, and interacting with other people. Such a heterogeneous environment generates a rich source of user-oriented data, which attracts not only researchers for studying and understanding social communities and individuals, but also attackers for infiltrating users' sensitive information to deliberately fulfill the economic, social, or political intents (e.g., unwanted advertising, user tracing) [29, 2]. This puts users' privacy at risk.

In response to these privacy concerns, social media generally takes action to protect those explicit sensitive user data like credentials by all means. However, with the rapid development in machine learning, and especially the revolutionary learning structures and capabilities raised by deep learning, it is highly probable for the attackers to launch automated attribute inferences from implicit data, which cause unintentional user attribute information leakage and threaten social media privacy [7, 30, 12]. For instance, a user's tweets can be fed to a well-trained machine learning model to infer the user's various private attributes, such as gender, age, and location [10]. Despite their remarkable inference ability, machine learning models are suffering from the inherent learning vulnerability to adversarial attacks [8, 5]. It has shown that by adding small perturbations to the input data, these pre-trained models can be easily fooled into misclassification. To this end, if we take advantage of such a vulnerability, social media privacy protection problem can be reduced to a feasible adversarial attack formulation problem against attribute inference attacks.

Some recent works [10, 19, 26, 11] showed that adversarial attacks have been starting to be leveraged as defenses against inference attacks, which present great potentials to help data obfuscation and privacy protection. However, the prior attempts of this kind focus on the specific application scenarios where their target is limited to continuous data. The investigation into more challenging text data of discrete property has been scarce. In fact, text data is an important component of social media, which shares the most significant privacy of users. On the other hand, natural language processing (NLP)-based models have been widely and effectively used to parse information of text data from different perspectives [17, 9, 15]. Therefore, in this paper, we would like to focus on text data to investigate how text-space adversarial attacks can be formulated to obfuscate users' attributes and enforce NLP-based inference attacks as less effective as possible for privacy disclosure.

More specifically, we present a text-space *adversarial attack as defense*, or *AaaD* for short, against NLP-

[*] College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802, USA.

[†] Corresponding author. dinghao@psu.edu.

based attribute inferences over social media data. AaaD proceeds by iteratively perturbing the source text originated from social media, such that its specific attribute label is changed, while the underlying constraints conformable to text-space attacks are satisfied. This naturally leads to the following two goals for AaaD: (1) constructing a sequence of constrained perturbations to automatically craft plausible adversarial texts, and (2) making the inference attack model fail to predict correct attribute values from the perturbed input texts. As an example, Figure 1 shows two perturbations performed by AaaD on a tweet. The first perturbation changes "*like*" to a semantically similar word "*love*", while the second one replaces "*joker*" with a visually similar word "*jokor*", both of which follow our defined constraints and successfully obfuscate the target attribute. Though there are challenges for attribute annotation on social media data, we believe that our work has implications on the applicability of adversarial attacks for undermining NLP-based inference threats and improving privacy protection in practice.

In summary, this paper has the following major contributions:

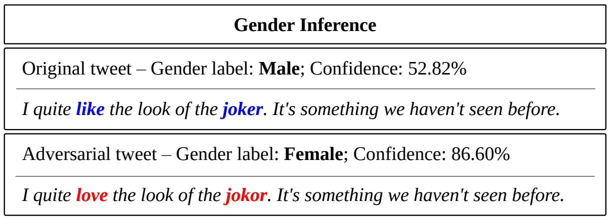
- We explore a novel perspective of protecting data privacy on social media, where we take advantage of the vulnerability of machine learning, and introduce adversarial attacks to forge latent feature representations and mislead attribute inference attacks.
- We design a new text-space adversarial attack AaaD for user attribute protection. In AaaD, the constraints conformable to text-space attacks are first defined; guided by that, we iteratively perturb the input text using the constructed word candidates chained by an upgraded population-based optimization to generate the adversarial texts, which are valid to humans but misclassified by the inference model.
- We conduct comprehensive experiments on two social media data sets to evaluate the performance of AaaD on attribute obfuscation and privacy effectiveness.

2 Methods and Technical Solutions

In this section, we first define the problem (i.e., inference model, adversarial attack as defense, and constraints), and then dive into technical details of AaaD.

2.1 Problem Statement

In social media environment, users tend to post text data for sharing; such text data may indicate their sensitive information, and thus easily expose the users to the attackers who can access the texts and infer the private attributes of interest to fulfill the harmful intents [26]. In this work, we assume that the attackers would take advantage of the implicit information from text data to train NLP-based models so as to achieve their inference goals.



Gender Inference

Original tweet – Gender label: **Male**; Confidence: 52.82%

*I quite **like** the look of the **joker**. It's something we haven't seen before.*

Adversarial tweet – Gender label: **Female**; Confidence: 86.60%

*I quite **love** the look of the **jokor**. It's something we haven't seen before.*

Figure 1: Attribute obfuscation by AaaD.

Inference threat model. We put our work under the practical black-box setting, where the devised adversarial attack is not aware of the threat model architecture, parameters, or training data, but capable of querying the threat model with text inputs and retrieving the output predictions for the attributes and their confidence scores [1]. Without loss of generality, we denote social media text data $\mathcal{D}$ to be of the form $\mathcal{D} = \{d_i, y_i^t\}_{i=1}^n$ of n texts, where each text $d \in \mathcal{D}$ is associated with a ground-truth label $y^t \in \mathcal{Y}^t$ for an attribute $t \in \mathcal{T}$; $\mathcal{Y}^t$ is the label set of the attribute t and $\mathcal{T}$ is the attribute set. For instance, $\mathcal{T}$ has different possible values, which is specified as $\mathcal{T} = \{\text{gender}, \text{age}, \text{location}, \text{political view}, \cdots\}$, while for gender attribute, $\mathcal{Y}^t = \{0\text{:male}, 1\text{:female}\}$. To facilitate an NLP-based inference model l using text implicit information, each text d has to be mapped into a k-dimensional feature vector $\mathbf{x} = \phi(d)$ where ϕ is a feature learning function such that the predicted label of text $\mathbf{x}$ can be derived from $\arg\max_{i \in \mathcal{Y}^t} l_i(\mathbf{x})$, while $l_i(\mathbf{x})$ is the confidence score of the attribute t's i-th label.

Adversarial attack as defense. Given an inference attack target (i.e., one attribute to infer), we formulate text-space adversarial attacks as defenses that attempt to automatically perturb the texts to obfuscate that attribute and prevent threat models from correctly identifying their private attribute values. As aforementioned, we consider the black-box setting such that our formulation is applicable to evade a wide range of attribute inference models. Formally, for an original text $\mathbf{x}$, the purpose of a text-space adversarial attack is to modify $\mathbf{x}$ with assigned label y^t to a text $\widehat{\mathbf{x}}$ that is classified to any other label $\widehat{y^t} \in \mathcal{Y}^t$, $\widehat{y^t} \neq y^t$ through adding a perturbation δ, the objective function of which can be defined as follows:

$$f(\mathbf{x} + \delta) = l_{y^t}(\mathbf{x} + \delta) - \max_{i \neq y^t}\{l_i(\mathbf{x} + \delta)\} \tag{2.1}$$

Clearly, $\mathbf{x}$ is classified as a member of $\widehat{y^t}$ if and only if $f(\mathbf{x} + \delta) < 0$ [22]. The majority of methods [3, 8, 20, 16] intuitively perform a gradient-based adversarial attack in the general feature space by solving the following optimization problem:

$$\begin{aligned} \delta^* &= \arg\min_{\delta\in\mathbb{R}^k} f(\mathbf{x}+\delta) \\ &\text{s.t.} \;\; \|\delta\|_p < \epsilon \;\text{ and }\; f(\mathbf{x}+\delta) < 0 \end{aligned} \tag{2.2}$$

However, gradients computed from the feature space are hard to define in text space due to its discrete property; also in black-box settings, it is impossible to compute gradients since the model parameters are not observable. Therefore, gradient-driven adversarial attack methods cannot be directly applied to text space. In addition, to formulate a feasible text-space adversarial attack, we have to comply with some essential constraints on the modification of the texts. To this end, in the following subsection, we identify the main types of constraints on the text-space transformations so that the perturbation δ satisfying such constraints will lead to a valid adversarial example to defend against attribute inference attacks for our problem.

Text-space attack constraints. Misleading the attributes of a text can be achieved with different levels of adversarial perturbations [23]. For instance, we can simply copy the words from others with different attribute labels for impersonation, or heavily obfuscate the source text for evasion. These adversarial attacks, however, suffer from semantic loss, generate implausible text, and have a noticeable effect on a human viewer. As such, we define a set of constraints to guide our text-space adversarial attack and clarify its strengths.

- *End-to-end learnability.* The major requirement in text-space adversarial example generation task is to enforce adversarial attacks to be performed from text space to text space rather than feature space. More specifically, the adversarial attacks should follow the transformation flow $\mathcal{D} \to \mathcal{D}$, where $d \mapsto \widehat{d}$ takes an original text d and generates an adversarial version $\widehat{d}$.
- *Semantic preservability.* The semantic preservability requires that the original text d and the adversarial text $\widehat{d}$ express the consistent semantic meaning to humans. Generally, distance metric over the feature space is used to limit the perturbation range [21], but there is no guarantee that small distance in the feature space preserves semantics for texts [6]. To address this issue, semantics can be evaluated at both text and word levels: for text level, the edit distance between d and $\widehat{d}$ should limit to small word operations made to d; for word level, a fine-tuned word embedding space could ensure semantic similarity between d and $\widehat{d}$.
- *Text plausibility.* The text plausibility requires that the generated adversarial example $\widehat{d}$ is valid to humans. In this respect, artifacts, which easily reveal that an adversarial text is invalid, will not be included. However, due to its fast-sharing and informal-writing property, social media may tolerate words with small misspellings or typos, which are still readable and plausible to humans.
- *Attack automaticity.* The attack automaticity requires that the perturbations are performed automatically without human intervention. This implies that possible changes to d should not include any handcrafted transformations or need re-engineering on different datasets. In this way, the adversarial attack can be reusable without extra updates.

2.2 Overview of Proposed Method AaaD The described threat model and attack constraints pose unique challenges to the design of our attack AaaD: (1) we consider black-box setting to formulate AaaD that is not aware of the inference model it tries to mislead but can query it for predicting confidence score, and (2) we follow the constraints to construct a sequence of plausible perturbations to automatically craft the adversarial text with preserved semantics. To this end, we propose to perturb the text tokens directly with guidance of the misclassification of the target attribute, which naturally satisfies the end-to-end learnability requirement. As the latent representations using word embedding can better encode the implicit information from texts than characters and the search space of possible changes over words is much smaller than characters, in our attack, we focus on perturbing the texts at word-level. For the word-level perturbations, we use edit distance metric in terms of the number of word changes to control the size of modifications. As such, we can update the feature-space adversarial attacks in Eq. (2.2) to a new text-space optimization problem as follows:

$$\begin{aligned} \delta^* &= \arg\min_{\delta\in\mathcal{W}} f(\phi(d+\delta)) \\ &\text{s.t.} \;\; \widehat{d} = d+\delta, \;\; s(\widehat{d}, d) < \epsilon \;\text{ and }\; f(\phi(\widehat{d})) < 0 \end{aligned} \tag{2.3}$$

where + implies the high-level word change, $s(\widehat{d}, d)$ denotes the number of different words between $\widehat{d}$ and d, and $\mathcal{W}$ is the set of plausible and semantic-preserving word candidates for perturbation. Based on Eq. (2.3), AaaD proceeds with a sequence of word perturbations, where each perturbation takes the current text d, replaces a chosen word with the optimized candidate, and generates a new version $\widehat{d}$ such that d and $\widehat{d}$ are semantically equivalent, until the attribute label is changed or the maximum allowed perturbation ϵ is reached. Note that, since all these operations and optimizations do not require manual intervention, and all the candidate constructions and word perturbations are performed on the

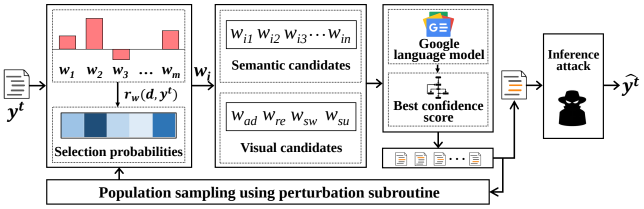


Figure 2: Overview of our proposed AaaD.

fly, we can accordingly ensure the automaticity for our proposed attack.

2.3 Technical Solutions Alzantot et al. [1] proposed to generate semantically similar adversarial texts by iteratively evolving candidate perturbations towards better solutions through a population-based optimization algorithm, where, however, random population sampling at each iteration makes the perturbation procedure inefficient, and none of the visually similar word candidates have been considered. On the contrary, Gao et al. merely applied character transformations to generate adversarial texts, which not only ignores word perturbations of semantic similarity but also is computationally expensive [6]. Different from these existing methods, we consider the social media property and elaborate our algorithm based on both semantically and visually similar word candidates for perturbations to force inference models to misbehave faster. More specifically, the realization of our text-space adversarial attack involves three building blocks: (1) select ready-to-perturb word, (2) construct semantically and visually similar candidates for each selected word, and (3) determine the best candidate for replacement. An overview of our attack strategy is illustrated in Figure 2. We discuss the technical details of these building blocks in the following separate subsections.

2.3.1 Selecting Ready-to-perturb Word To reduce the vast random search space of possible words encountered by the original population-based adversarial attack proposed in [1], we would like to first score the importance of words to guide the population sampling to touch the important words and thus expedite the adversarial text generation. Since the black-box setting does not allow us to compute the partial derivative of the confidence score regarding the predicted attribute label at each input word to approximate the word importance, one feasible way is to directly measure the effect of the word by removing it from the text [14, 6], and then compute the confidence score difference before and after removing a word to specify its importance. Specifically, we assume the input text $d = (w_1, w_2, \cdots, w_m)$, and the scoring function that determines the importance of i-th word in d can be denoted as:

$$\begin{aligned} r_{w_i}(d, y^t) = l_{y^t}(w_1, w_2, \cdots, w_m) \\ - l_{y^t}(w_1, \cdots, w_{i-1}, w_{i+1}, \cdots, w_m) \end{aligned} \tag{2.4}$$

Eq. (2.4) implies that the greater the contribution of a word to attribute prediction, the more likely we are to modify it to mislead inference models. Considering the fact that there exist some stop words (e.g., to, the, a, it, etc.) or irrelevant words in a text that make little sense to tamper with, we further use softmax function to normalize the importance scores to serve as word selection probabilities for population sampling. In this regard, we give priority to modifying the more important words in the sentences.

2.3.2 Constructing Word Candidates In order to satisfy the constraints that the generated adversarial text should retain semantic equivalence with the original text and visually hurt little to human understanding on social media text contents, we consider two types of word candidates for perturbation: semantically similar candidates and visually similar candidates.

Semantically similar candidates. We obtain a set of words by searching the nearest neighbors of the ready-to-perturb word according to the distance in word embedding space. Here we define a threshold η to filter out candidates with distance greater than η such that the semantic preservability requirement could be less violated. GloVe is a context-aware word embedding space [21], but it tends to coalesce the notions of semantic similarity and conceptual association and thus fails to distinguish synonyms from antonyms [18]. Examples of such anomalies can be seen in Table 1, where words such as "high" and "low", and "similar" and "different" are deemed similar in GloVe embedding space; replacing such words with each other would completely change the semantics of the text. By contrast, counter-fitting embedding provided by Mrkšić et al. [18] leverages synonym and antonym relations to fine-tune GloVe vectors (shown in Table 1), which is a better choice for our problem. Therefore, we use counter-fitting embedding to search for the nearest neighbors for the given word.

Visually similar candidates. Apart from legitimate candidates derived from vocabulary, we also expand the candidate pool with slightly transformed words. The reasons behind this are that (1) social media, as a fast-sharing and informal-writing environment, is highly misspelling-tolerant, where satiric or deliberate misspellings are not uncommon; (2) words with small character changes are imperceptibly to human eyes and have no significant impact on semantics [24]; and (3) would also very likely enforce the selected word to be out

Table 1: Nearest neighbors for target words using different embeddings: antonym and synonym example pairs are highlighted as red and blue respectively

Embedding	high	red	similar
GloVe	low higher highest	blue yellow purple	same different particular
Counter-fitting	highest supreme higher	rojo flushed cardinal	equivalent same like

of dictionary with "unknown" embedding such that the output classification may change [6, 14]. To guarantee the text plausibility, we restrict that only small changes can be performed on the original word to create visually similar candidates, and those transformed words will not be selected for a second perturbation. We present different word transformation methods as follows: (1) add a space or a random character into the word; (2) remove a random character from the word; (3) swap any two adjacent characters; (4) substitute a character in the word with a random character. Note that, both the first and last positions in the original word will not be modified for better perturbation invisibility.

2.3.3 Determining Best Candidate for Replacement Based on the constructed word candidates, we can observe that the semantically similar candidates may not be always used in the same contexts (e.g., "red" and "flushed" in Table 1). To address this issue, we proceed with filtering out those candidates that do not fit within the context by using Google language model [4] to further ensure the semantic correctness. The rest are then integrated with all the misspellings to form the final candidates. Afterwards, we choose the best candidate among them that will maximize the confidence score of the target attribute label $\widehat{y^t}$ ($\widehat{y^t} \neq y^t$) prediction when it replaces the ready-to-perturb word in d. Then we perturb the text with the optimal candidate and generate a new text as a population member.

2.3.4 Population-based Optimization Equipped with three building blocks, we can formulate a *perturbation subroutine* that accepts an input text (either perturbed or original), perturbs one selected word, and generates a perturbed-version text towards the misclassification of the target attribute. In this way, we are ready to generate a set of these perturbations for the given text. We aim to minimize the number of word perturbations, which makes the adversarial text less likely to be perceived. Therefore, instead of using greedy-based procedure [6, 14], we follow the work presented in [1] and leverage population-based optimization to chain the word perturbations together such that our adversarial attack as defense target in the text space is reached.

The population-based optimization performs by sampling the population at each iteration, searching for those better population members that achieve better performances, and taking them as "parents" to produce the population for next iteration [1]. The population at each iteration is called generation. Given an input text d, the sampled population $\mathcal{P}^0$ is initialized as N perturbed texts created by performing perturbation subroutine N times on different selected words in d. At generation g ($g \geq 0$), the confidence score of each population member for predicting the target attribute label $\widehat{y^t}$ ($\widehat{y^t} \neq y^t$) is computed. If the predicted attribute label of a population member is equal to $\widehat{y^t}$, the optimization is complete and such population member is returned as a successful adversarial text $\widehat{d}$; otherwise, pairs of population members from $\mathcal{P}^g$ are sampled according to their confidence scores, where each pair of them synthesizes a "child" text in a crossover way, such that N "child" texts are generated. Accordingly, the new population $\mathcal{P}^{g+1}$ is sampled by performing perturbation subroutine on these N "child" texts. After that, the optimization procedure moves to next generation $g + 1$. Different from the prior work, we improve the success rate of population samplings by choosing those ready-to-perturb words of high importance scores, while visually similar candidates introduced further expedite the adversarial example generation. Through the proposed AaaD, we can turn text-space adversarial attacks into defense against the attribute inference attacks, and thus protect the social media data privacy.

3 Empirical Evaluation

3.1 Experimental Setup

Datasets. We test our method on two social media datasets: user gender tweets and blog authorship corpus [25], which are good representatives for social media data as tweets and blogs are posted by various users, and can be easily accessed by attackers to uncover their private attributes. Specifically, user gender tweets are collected from Kaggle[1]. We filter out those with gender confidence score less than 0.5, and thus obtain 13,926 tweets with two genders (female and male). For blog data, it consists of 19,320 documents, each of which contains the posts by a single user. In our experiments, we extract 25,176 blogs with two attribute inference

[1] https://www.kaggle.com/crowdflower/twitter-user-gender-classification

settings: (1) gender (female and male), and (2) age (teenagers (age between 13-18) and adults (age between 23-45)). Note that, age-groups 19-22 are missing in the original data. For tweets and blogs, we randomly split them into training and testing datasets. The data statistics are summarized in Table 2.

Threat model for attribute inference attacks. To show our black-box method AaaD is effective, we perform our experiments on a well trained word-level LSTM since it is one of the most popular and feasible neural networks to address NLP problems, which can be easily built by the attackers to perform attribute inferences. The LSTM network contains 250 hidden units, where the dimension of each hidden unit is 128, and the dropout rate is 0.3. We use the pre-trained GloVe [21] model to map each word into a 300-dimensional embedding space. To conduct inference attacks, the training texts are fed to the model for training first and then the trained model is used to predict private attributes over the testing texts. The inference accuracy for Twitter-gender, blog-gender, and blog-age is 62.25%, 69.20%, and 72.92%, relatively close to the state-of-the-art results on each dataset.

Text-space adversarial attack baselines. We compare AaaD with two other state-of-the-art text-space adversarial attack methods. Both methods use the same black-box setting. They can be specified as follows:

- Genetic attack [1]: this attack uses population-based optimization algorithm to generate semantically similar adversarial examples, where population sampling is performed in a random way at each generation.
- WordBug [6]: this attack scores the word importance, and perturbs the words in the descending order regarding word importance score using word misspellings; to be comparable, we measure the effect of each word by removing it from the text.

Parameter setting. We use euclidean distance as distance metric to construct semantic-similar candidates from embedding space, where the distance threshold is set to $\eta = 0.5$ to filter out those less similar ones, and the size of candidate pool for each word is set as 8 (we choose the best one for replacement). The iteration size $I = 20$. Also, we remove half of semantically similar candidates (i.e., 4 in this setting) using Google language model. We limit the maximum allowed word perturbations to 25% of the text length. We further evaluate its impact on attack performance in Section 3.4.

3.2 Evaluation of AaaD In this section, we validate the effectiveness of AaaD over well-trained inference models to defend against attribute inference attacks. We perturb the correctly classified text examples from the testing datasets of three attribute settings to evaluate our algorithm AaaD under different population sizes. In particular, we test the results of our generated adversarial texts with population size $N \in \{5, 10, 15, 20\}$ respectively against different inference attacks. The experimental results are shown in Table 3. As we can see from the results, AaaD drastically decreases the accuracy of the state-of-the-art inference models and achieves the goal of obfuscating the attributes and protecting social media text data privacy. When $N = 15$, our method reduces the accuracy of the Twitter-gender inference attack from 62.25% to 15.64%; for the larger and longer blog data, we degrade inference accuracy of gender and age from 69.20% to 8.21% and from 72.92% to 18.59%. Generally, when we enlarge the population size, the success rate of generating adversarial samples increases while the required perturbation number tends to go up as well. However, due to the perturbation limit for each text, the actual attack performance might not always improve for larger population size. For example, the inference accuracy for three settings either slightly decreases or stays flat when N changes from 15 to 20. As such, we use $N = 15$ throughout the following evaluations.

Table 2: Comparing statistics of the two datasets

Dataset	Attribute	#Posts	#Training	#Testing
Twitter	Gender	13,926	9,763	2,450
Blog	Gender, Age	25,176	17,623	7,553

Table 3: Evaluation of AaaD via inference accuracy (%)

Inference	Population Size				
	-	$N = 5$	$N = 10$	$N = 15$	$N = 20$
Twitter-gender	62.25	18.50	17.94	15.64	15.61
Blog-gender	69.20	10.86	9.58	8.21	8.21
Blog-age	72.92	21.88	20.01	18.59	21.44

3.3 Comparisons with Other Text-space Adversarial Attacks We also compare AaaD with Genetic attack [1] and WordBug [6]. Specifically, we randomly sample 50% of correctly classified examples from the testing tweets and blogs to measure the performance of attacks. The comparative results are illustrated in Table 4, where Genetic attack achieves better attack success rate than WordBug while WordBug perturbs less words; and AaaD outperforms both baselines with slightly higher perturbation number than WordBug. From the results, we can observe that (1) pro-

Table 4: Comparisons of different text-space attacks

Inference	Metric (%)	Genetic	WordBug	AaaD
Twitter-gender	Success Rate	58.06	38.96	**78.98**
	Median Ptb	13.48	**8.33**	11.11
	Mean Ptb	15.83	**10.34**	12.14
Blog-gender	Success Rate	83.97	56.04	**88.29**
	Median Ptb	7.05	**3.38**	5.03
	Mean Ptb	9.94	**5.93**	8.89
Blog-age	Success Rate	68.33	52.02	**75.91**
	Median Ptb	12.31	10.32	**9.38**
	Mean Ptb	13.37	**9.81**	11.89

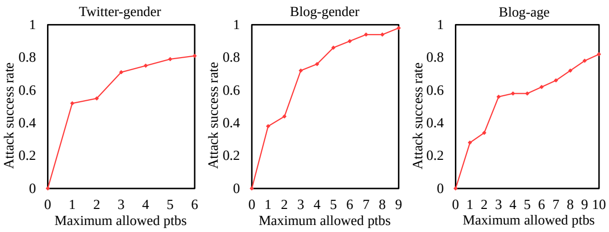


Figure 3: Evaluation on maximum allowed perturbation (ϵ) via cumulative distribution of attack success rate.

jecting an important word into "unknown" may enforce inference models to misbehave, while ignoring semantically similar candidates would also miss good evasion chances, and (2) leveraging word importance to facilitate population-based optimization expedites adversarial example generation. When we look into the generated adversarial texts, we find that WordBug fails in most of those adversarial texts with more modifications required, and hence obtains a small perturbation number on average in results. By contrast, AaaD either converts those failed texts from Genetic and WordBug to adversarial examples, or decreases the number of required perturbations, which significantly advances the text-space adversarial attack with respect to effectiveness and efficiency. Thus, we conclude that AaaD is feasible in a real social media environment on attribute obfuscation and privacy effectiveness.

3.4 Parameter Evaluation In this set of experiments, we conduct the parameter analysis of how different choices of ϵ (i.e., maximum allowed word perturbations) will affect the performance of AaaD, since ϵ significantly reflects the similarity between the generated adversarial texts and the original texts, and thus has direct impact on the semantic preservability and plausibility of the adversarial texts. We use the cumulative distribution function (CDF) of attack success rate regarding the number of ϵ to illustrate the evaluation results. From the results shown in Figure 3, we can observe that as ϵ increases, the attack success rate increases as well because of the larger modification space, but the mean sentence semantics quality would decease. Actually, using AaaD, most of the generated adversarial texts manage to evade the inference models after perturbing very few words in the texts. More specifically, for Twitter-gender inference, about 64% of the successful adversarial texts perturb only one word, while this ratio increases to 88% when $\epsilon \leq 3$; for blog-gender inference, the ratios are 39% and 73% when $\epsilon \leq 1$ and $\epsilon \leq 3$ respectively; for blog-age inference, these two ratios are 30% and 59%. All these results imply that (1) AaaD enables most of adversarial texts to be similar to the original texts; (2) the number of perturbations relatively relies on the length of the texts: the average lengths for twitter-gender, blog-gender and blog-age are 14, 36 and 54 respectively, while the average perturbations are 1.7, 3.2, and 6.5 for the corresponding inference tasks, where we can also see that age attribute seems more difficult to be obfuscated.

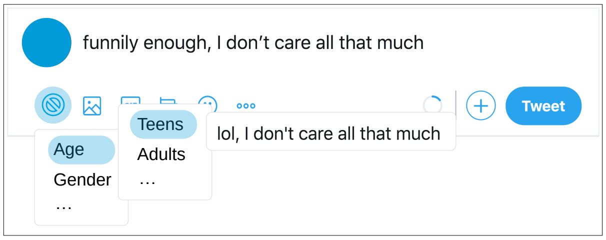
3.5 Qualitative Analysis

Case study. In this section, we present some of our generated adversarial texts in Table 5. We can have a straightforward insight that AaaD perturbs important words in an either semantic or visual-similar replacement manner towards the opposite attribute target. For instance, "recognize" and "last" are replaced with "recoqnize" and "final" in a tweet that changes the gender from female to male; AaaD also performs four perturbations in a blog that modify "awarded", "torn", "offer", and "would" to "allotted", "hesitant", "offar", and "ought" respectively to enforce a misclassification on gender with high confidence score; for blog-age setting, the perturbations are "just" $\mapsto$ "exclusively", "school" $\mapsto$ "schkool", "boring" $\mapsto$ "tiresome", "hard" $\mapsto$ "challenging", and "stupid" $\mapsto$ "foolish". All these perturbations meet our designed constraints and successfully mislead the inference attack models.

User study. We further ask 15 volunteers to simulate the real-world social media scenarios to review the adversarial texts generated by AaaD. Our user study is composed of two parts: (1) testing the plausibility, where volunteers are given 100 (half original and half adversarial) texts randomly selected from three inference settings to evaluate if they are readable to them; (2) investigating the semantic preservability, where volunteers are given 50 sample pairs consisting of the original texts and their adversarial versions without knowing orders to evaluate their semantic similarity with 4 scales from 1 (very different) to 4 (very similar). The results are consistent to our experimental observations. The plausibility rates are (84.67±4.11)% and (82.53±3.22)%

Table 5: Examples of generated adversarial texts for different inference settings

Classifier: Twitter-gender. **Original label:** female (confidence=86.72%).**New label:** male (confidence=53.36%)
I didn't ~~recognize~~ recoqnize him till the ~~last~~ final moment when she looks at him from the window.
Classifier: Blog-gender. **Original label:** female (confidence=93.39%). **New label:** male (confidence=72.28%)
So I get home today and there is a letter from the Dubya telling me that I have been ~~awarded~~ allotted 5000 in student aid for this year. Omigod! That's more than I thought I would get. I am so ~~torn~~ hesitant I really want to take advantage of this but I want to stay here now. I have to see what is willing to ~~offer~~ offar. I have to contact financial aid and do a comparison. It might be better considering I ~~would~~ ought save an out of state fee.
Classifier: Blog-age. **Original label:** teens (confidence=96.46%). **New label:** adults (confidence=61.11%)
My head is like a 8 year old's finger painting. It's ~~just~~ exclusively a buncha colors and hues all smeared up and stuff, so anyways ~~school~~ schkool is off to a really ~~boring~~ tiresome start. Oh well, at least I won't have to work very ~~hard~~ challenging uhhhhh..Hm either. There's a lot going on or hardly anything at all. Ok this post is really ~~stupid~~ foolish and it doesn't make any sense so I'm gonna call it quits before I look even more stupid than I usually do.


funnily enough, I don't care all that much

lol, I don't care all that much

Figure 4: Example of attribute obfuscation service.

on the original and adversarial texts, which imply that the perturbations introduced by AaaD barely hurt people's perception. The average semantic similarity score is 2.99 ± 0.30, which indicates that the perceived difference is also small. This user study further validates the rationality of AaaD.

4 Discussion

Applications. In practice, AaaD can apply to an attribute obfuscation service provided on social media client side. For example, as illustrated in Figure 4, AaaD is developed as an API that is integrated into Twitter posting and editing systems to allow users to choose the adversarial text according to their target attribute (i.e., age) and text content. Once users give privileges to this replacement, the posting data will be obfuscated and updated on behalf of the users. Similarly, AaaD can also serve to exhaustively obfuscate the social media data before making it publicly available.

Limitations. Our approach also poses some limitations which we discuss as follows. (1) The lack of the ground truth on real social media systems disables AaaD from generating the adversarial texts in a real-time fashion. To this end, we may need to first recognize the targets to facilitate better attribute protection. (2) The inference attackers could leverage more robust learning paradigms (e.g., adversarial training, misspelling correction) to reveal attributes and thus degrade AaaD. We acknowledge this limitation and leave the investigation on this arms race as our future work, yet it does not impact the general validity of our new insight on the adversarial attacks for attribute obfuscation and protection, as robust models could always be evaded by more sophisticated adversarial techniques.

5 Related Work

Anonymization paradigms [26, 2] have been conventionally developed to anonymize and protect user identifiable information on social media, while machine learning based inferences [19, 7] can easily utilize non-anonymous data to re-identify users. Some promising defense methods have been thus presented to alleviate such inference attacks, such as differential privacy [28], deep data obfuscation [13], and game-theoretic optimization [27, 10], but they are still suffering from limitations of either cost-expensive, large utility loss, or introducing additional privacy concerns. Recently, the vulnerabilities of machine learning are starting to be leveraged as defenses against inference attacks [10, 19, 26, 11], which have delivered great potentials. However, most of these works aim to combat inference attacks over continuous data, while only very few of them perform on text data. Differently, our work focuses on more challenging text data, and advances the existing text-space adversarial attacks [1, 6, 14] by new candidate construction and optimization procedure.

6 Conclusion

In this paper, we cast social media privacy protection problem as an adversarial attack formulation problem to defend against attribute inference attacks. We investigate text data for our problem and present a text-space adversarial attack AaaD, where a sequence of constrained yet plausible perturbations are formulated to craft the adversarial texts and chained by an upgraded population-based optimization algorithm. We conduct experimental studies on real-world social media datasets to evaluate the performance of AaaD, which validate its effectiveness against inference attacks. Despite the challenges and limitations, we believe that our work has implications on the applicability of adversarial attacks for attribute obfuscation and privacy protection in practice.

Acknowledgments

The work was supported in part by a seed grant from the Penn State Center for Security Research and Education (CSRE) and the PNC Technologies Career Development Professorship.

References

[1] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," *arXiv:1804.07998*, 2018.

[2] G. Beigi, K. Shu, Y. Zhang, and H. Liu, "Securing social media user data: An adversarial approach," in *Proceedings of HT*, 2018, pp. 165–173.

[3] N. Carlini and D. Wagner, "Towards evaluating robustness of neural networks," in *SP*, 2017, pp. 39–57.

[4] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," *arXiv preprint arXiv:1312.3005*, 2013.

[5] L. Chen, Y. Ye, and T. Bourlai, "Adversarial machine learning in malware detection: Arms race between evasion attack and defense," in *EISIC*, 2017, pp. 99–106.

[6] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *SPW*, 2018, pp. 50–56.

[7] N. Z. Gong and B. Liu, "Attribute inference attacks in online social networks," *TOPS*, 21 (1), pp. 3, 2018.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[9] K. Ikeda, G. Hattori, C. Ono, H. Asoh, and T. Higashino, "Twitter user profiling on text and community mining for market analysis," *KBS*, pp. 35–47, 2013.

[10] J. Jia and N. Z. Gong, "Attriguard: A practical defense against attribute inference attacks via adversarial machine learning," in *USENIX Security*, 2018, pp. 513–529.

[11] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *CCS*, 2019, pp. 259–274.

[12] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, "Inferring user attributes in online social networks using markov random fields," in *WWW*, 2017, pp. 1561–1569.

[13] Y. Keswani, H. Trivedi, P. Mehta, and P. Majumder, "Author masking through translation." in *CLEF (Working Notes)*, 2016, pp. 890–894.

[14] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "TextBugger: Generating adversarial text against real-world applications," *arXiv preprint arXiv:1812.05271*, 2018.

[15] A. Makazhanov, D. Rafiei, and M. Waqar, "Predicting political preference of twitter users," *Social Network Analysis and Mining*, 4 (1), pp. 193, 2014.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *CVPR*, 2016, pp. 2574–2582.

[17] A. Morgan-Lopez, A. Kim, R. Chew, and P. Ruddle, "Predicting age groups of twitter users based on language and metadata features," *PloS one*, 2017.

[18] N. Mrkšić, D. O. Séaghdha, B. Thomson, M. Gašić, L. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, and S. Young, "Counter-fitting word vectors to linguistic constraints," *arXiv:1603.00892*, 2016.

[19] S. J. Oh, M. Fritz, and B. Schiele, "Adversarial image perturbation for privacy protection a game theory perspective," in *ICCV*, 2017, pp. 1491–1500.

[20] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, and A. Swami, "Practical black-box attacks against machine learning," in *AsiaCCS*, 2017, pp. 506–519.

[21] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.

[22] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ml attacks in the problem space," *arXiv:1911.02142*, 2019.

[23] E. Quiring, A. Maier, and K. Rieck, "Misleading authorship attribution of source code using adversarial learning," in *USENIX Security*, 2019, pp. 479–496.

[24] G. Rawlinson, "The significance of letter position in word recognition," *IEEE AESM*, 22, pp. 26–27, 2007.

[25] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker, "Effects of age and gender on blogging." in *AAAI Symposium: CAAW*, 2006, pp. 199–205.

[26] R. Shetty, B. Schiele, and M. Fritz, "A4nt: author attribute anonymity by adversarial training of neural machine translation," in *USENIX Security*, 2018, pp. 1633–1650.

[27] R. Shokri, "Privacy games: Optimal user-centric data obfuscation," *PETS*, 2015 (2), pp. 299–315, 2015.

[28] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *USENIX Security*, 2017, pp. 729–745.

[29] S. Yu, Y. Vorobeychik, and S. Alfeld, "Adversarial classification on social networks," in *AAMAS*, 2018, pp. 211–219.

[30] Y. Zhang, M. Humbert, T. Rahman, C.-T. Li, J. Pang, and M. Backes, "Tagvisor: A privacy advisor for sharing hashtags," in *WWW*, 2018, pp. 287–296.