Abstract

Redaction has been the most common approach to protecting text data, but synthetic data presents a potentially more reliable alternative for disclosure control. By producing new sample values which closely follow the original sample distribution but do not contain real values, privacy protection can be improved while utility from the data for specific purposes is maintained. We extend the synthetic data approach to natural language by developing a neural generative model for such data. We find that the synthetic models outperform simple redaction on both comparative risk and utility.

1 Introduction

In recent years there has been a substantial rise in demand for reproducible research [9], coupled with growing interest in and availability of observational research using non-traditional data sources, such as social media, geo-tagged data, or other data "exhausts" [21]. Often standing in opposition to the desire of sharing data, particularly social or behavioral information, is the privacy of individuals or organization whose information is contained in it. Statistical Disclosure Control (SDC) methods have developed for researchers to alter data sets in order to safely release public versions. As data demands have grown, these methods have evolved and progressed to facilitate releases of analytically useful public data sets. In some cases, protection of individuals’ information is required by law, such as data compliant with the U.S.’s Health Insurance Portability and Accountability Act of 1996, while in other cases it is a question of ethics and corporate responsibility. This latter situation includes high profile cases where data custodians have failed to sufficiently protect the data, and researchers revealed personal individual information leading to significant reputation damage for the data custodians. Some examples include the Netflix prize data [13], the AOL data [1], or the Facebook data [28].

Most SDC work up to this point has focused either on tabular data or microdata, primarily using methods of suppression, aggregation, perturbation, or synthetic data, see [8] or [27] for an overview of these techniques. However, with new technology and the dawn of the era of large-scale data, unstructured data such as natural language has become more frequently used and desirable for researchers. Application of classical SDC methods to unstructured text is difficult for two primary reasons: first, it lacks the standard structure and values seen in classical social science categorical or quantitative data and second, it is often so large that classical methods are computationally unfeasible.
As a result, specific methods are needed to ensure disclosure control when publishing or sharing text data. So far the question of privacy for unstructured data has been largely ignored. For commonly used data sources such as Twitter data there is typically no practice of disclosure control when sharing data even though some researchers have set out ethical guidelines specifying that privacy should be maintained while sharing Twitter data [22].

In this paper we argue for disclosure control when releasing social media text data and present a generative method for satisfying this need. Although there are currently few legal requirements (apart from proprietary concerns) for protecting privacy when sharing social media data, research ethics should still require the protection of individual privacy. Generally social media users agree to having their information published on a given platform, but users might feel differently if their data is connected to other data or if additional information is generated through research. For example, it is not hard to imagine a study using sentiment analysis to assign typical attributes such as sad or happy tweets but also more sensitive ones such as violent or racist. If such results were attributed to tweets and it were possible to link these tweets back to specific users, it would be a serious intrusion of privacy. Twitter, as well as some other social media platforms such as Facebook, also typically archive posts after a certain period of time, so Twitter users can reasonably expect that their archived tweets will not be shared in such a way that they can be re-identified.

Accordingly, we define risk in this paper as the proportion of identifiable users in a (Twitter) data set. This emulates the situation where a researcher releases archived tweets with anonymized user labels, and a malicious party trains a model using non-archived tweets to try to identify specific users in the shared data. To minimize this risk, we propose generating a synthetic corpus to release in lieu of original tweets. We draw on the theoretical concept of synthetic data in the SDC literature, which originated from the concept of multiple imputation, see [23], [17], and [18]. The idea is that privacy is maintained because no original samples are released, and utility is maintained theoretically by drawing similar samples from the same generative model that produced the original data. To synthesize such high dimensional data (i.e., text), we design a neural generative language model.

To compare the methods, we estimate the risk and utility with the (i) original tweet corpus, which does not alter the free text at all, (ii) redaction of all twitter handles and hashtags in the text, and (iii) full data synthesis using variations of a neural generative model. Intuitively, it may be possible that most of the identification risk in tweets comes from named entities such as hashtags, handles, or specific terms frequently used by users. In this case, redaction of these terms would be sufficient to reduce any risk of identification. We show, however, that the original tweets carry substantial risk, and simple redaction methods do not completely protect the data, but synthesis methods almost entirely minimize the proportion of identifiable users. For utility, we find that one neural model outperforms redaction by maintaining higher distributional similarity for users’ tweet lengths and language used.

This paper, like any synthesis approach, assumes that the generative model perfectly captures the distribution of interest w.r.t. the research to be performed on the released dataset in order to maximize utility. This may be the case for work on topic modeling, sentiment, and social network analysis, but in general synthetic data cannot fulfill every research need. A common scenario in the practice is to use synthetic data as an exploratory tool, where research models are developed on the released dataset, but evaluated on original non-deidentified data under the care of the custodian. Close similarity between the original and released data is desirable but they are not assumed to be exchangeable.

2 Research Design

2.1 Background on Synthetic Data & Redaction for Disclosure Control

We pursue two approaches to free text privacy: redaction and text synthesis. Redaction represents a common approach to protecting sensitive information in text, and it parallels concepts such as removal of direct identifiers or suppression of quasi-identifiers in tabular or microdata. Redaction has been implemented largely to protect privacy in other releases of unstructured text, largely for medical or government documents, see [6] for example. Recent redaction research has focused on automated methods for redaction of text such as [3], [5], or [24]. Redaction relies upon the concept that text data is only identifiable by key identifiers in the text which can be linked to other data sources. These identifiers can be as simple as twitter handles or hashtags in the case of Twitter data, or they can be more sophisticated such as names of locations, companies, or individuals. In this paper we use only redaction of handles and hashtags, since it can be argued that these comprise the large majority of
identifying entities in Twitter data. We tested some smart redaction methods but preliminary results suggested they did not improve protection, and we did not use then for the final results.

We compare redaction with synthetic data, for which we present a novel model for generating synthetic text. Synthetic data developed originally from the multiple imputation literature, [23], based on the simple concept of drawing new “samples” from a Bayesian posterior predictive distribution (BPPD), estimating the generative distribution, to release in place of the original data. If modeled well, synthetic data should capture the original distributional aspects of the data giving high utility, and none of the values are the original ones ensuring low risk. This method has a strong methodological base in the privacy literature. For a brief overview, early works included [17] and [18], [19] extended the methods for approximating generative models to non-parametric techniques, and [20] introduced the concept of simple synthesis showing that draws from the BPPD were not always necessary. See [4] for an overview of synthetic data methods.

Traditional synthetic data, such as draws from the BPPD, is not realistic in the scenario we present in this paper. These generative models are either approximated jointly, if the form is known, or using fully conditional sequential models. Both in terms of the model estimates and the computational burden, neither of these present a viable option for unstructured text data. For example, to use a standard synthetic data method for our data set would have required estimating over 40,000 Multinomial regression equations, each fully conditioned on the all previous ones. As current synthetic data do not methods exist to handle this type of data, we implement neural architectures to address this problem. Still, our work fits in the general synthetic data framework, since we approximate an underlying generative process along with a type of joint distribution between text and users.

2.2 A Neural Synthesis Model

Given a finite set of users (or authors) and a finite set of documents, which could be short or long texts, we aim to model its unknown data generating process. In particular, to better capture temporal and structural information inherent in the data, given a finite, dictionary $D$, we hypothesize that the distribution of interest is $p(c_t|y,C_{\text{history}})$, where $c_t$ is a 1-of-$|D|$ encoding of a symbol (either a character or token) at time $t$ indexed in $D$. $C_{\text{history}} = (c_{t-1}, c_{t-2}, \ldots, c_{t-m})$, which is the symbol’s history of length $m$ and $y$ is the index of the user this symbol is to be associated to an encoded symbol. By learning a good generative model of this conditional distribution, we would then be able to generate valid symbol sequence samples that are representative of specific, target users.

We draw inspiration from [14] and [11], which are effectively cases of hybrid architectures that model multiple perspectives of a data-set through separate blocks of observed variables. In order to learn from and generate sequences of symbols, one must account for the inherent temporal information (or rather, ordering of symbols). To this end, we design a recurrent neural architecture that specifically learns to model $p(c_t|y,C_{\text{history}})$. The architecture processes symbol encodings sequentially while at each time step also taking an encoding of the current user index $e_y$ (also 1-of-k binary encoding) and is trained to predict the symbol at $t$ given a previous token at $t - 1$ and the model’s current internal hidden state $h_{t-1}$. A model is built given the original corpus we desire to generate synthetic data reflective of, and, as mentioned earlier, symbols could either be characters or words. The architecture, unfolded over 4 time steps, is shown in Figure[1]

Our specialized neural architecture also addresses a recent problem found in neural conversation agents [25]. This issue is one of “coherence”, where the model has no sense of identity or self (even at the crudest level), and thus may be asked the question, “Are you married?” responding with “No!”, and then followed up with the question “What is your wife’s name?”, responding, “Cynthia”. While this issue is more prominent in sequence-to-sequence modeling tasks (such as question-answering), we also argue that in synthetic data-generation, where samples often come with meta-data, having a model that preserves local information such as which user generated which sample is crucial. While there are a variety of approaches one may take to preserving such local information, such as building a unique model per user, we decided to encode user identifiers in a sparse one-hot vector, since this is a problem of density modeling and not necessarily of prediction. In this way, we may construct a single model that models multiple users at the same time, while also benefiting from the aggregation of knowledge across all document samples. We note that our work is not the only one that has attempted to address the coherence/consistency problem in neural models [25],[12]. However, our approach, which is also simpler in representing user-local information, notably differs in the ultimate goal of
Figure 1: The proposed, user-based neural generative architecture, unfolded over four time steps. Note that parameters are shared across each time-step.

what is being modeled and the problem context (for example, our goal is simply to build a useful generative model, not a sequence-to-sequence mapper).

Formally, with model parameters \( \Theta = (W_{user}, W_{hid}, W_{rec}, W_{out}) \) (biases have been omitted for clarity), we calculate hidden and output states via the following set of equations:

\[
\begin{align*}
\mathbf{h}_t &= \phi_{hid}(W_{hid} \mathbf{x} + W_{user} \mathbf{e}_y + W_{rec} \mathbf{h}_{t-1}) \\
\mathbf{o}_t &= \phi_{out}(W_{out} \mathbf{h}_t)
\end{align*}
\]

where \( \mathbf{e}_y \) is the one-hot encoding of a user index \( y \) and \( (\phi_{hid}, \phi_{out}) \) are the activation functions for input-to-hidden and hidden-to-output layers respectively. In this paper, \( \phi_{out} \) was set to be the softmax activation function, \( \phi(v) = e^{-v} / \sum_{v_i \in \{v_1, \ldots, v_{|y|}\}} e^{-v_i} \), so that outputs lie on the simplex. Note that this single-hidden layer model is “deep in time” and can be easily made to be deep in structure (i.e., stacking more hidden layers). However, each hidden layer should sport a set of skip-connections to the user input layer, so that way all layer activations of the model are biased towards user-local distributed representations. Parameters of the model are fit to the data via stochastic gradient descent using truncated back-propagation through time (where \( m \) is used to control the length of the window, or number of steps back in time). The objective is to minimize the negative log likelihood of the predictive posterior of the sequence loss:

\[
\mathcal{L}(p(\mathbf{c}_t|y, C_{history})) = - \sum_{t=1}^{m} \log p(\mathbf{c}_t|y, \mathbf{c}_{t-1})
\]

The parameters used to model individual users are also modified as part of the back-propagation-based procedure. Note that these learnt user “embeddings” could be useful for auxiliary tasks, such as clustering user representations together for similarity-search-based applications.

To generate samples from the neural model, we simply make use of the model’s efficient inference procedure, similar to that in [7]. Specifically, by clamping the input units corresponding to a desired target user index and feeding in a “null” vector (or vector of all zeroes) as initial input, we may sample from its output probabilities and ultimately generate synthetic symbol sequences for individuals by feeding in a sample of model’s predicted output back in as input for the next step. A sequence is continuously generated until either a simple end-token is generated (in our case this was a token in the set [, !?], or simple punctuation) or an upper bound on the character limit is reached (this is particularly useful for Twitter data, which naturally caps text at 140 characters).

3 Experimental Results

We test our approach using a sample of Twitter data that was collected in the context of a study on anti-immigration sentiment. We construct a hypothetical release data set, consisting of 5364 tweets (413 unique users) and an attacker data set consisting of 4387 tweets (414 users).
Table 1: Example softsignRNN synthetic tweets for two users

| user | tweet                                                                 |
|------|------------------------------------------------------------------------|
| 0    | #islam https://t.co/nea7jbaW6q #pjnet #uk #quran #muhammad #wahhabism  |
|      | -LRB- #defeat #jesus -LRB- iraq if pm dangerous declare 2016 threaten   |
|      | #religion without #spirituality #politicalislam #365usa #love #religion |
|      | #united #terbaru #isis #wahhabi #isis #uk #islam may european          |
| 1    | police claimed 2 women person chaghmaghh culture : islamization of it    |
|      | new metro key u.s. on you stupidity celebrating light medical ...        |
|      | slogans warning for a be to online that could stop means mayor again israel |
|      | roulette amazing and ... # god hillary pop allows boy for https://t.co/ylfdtlzvht |

3.1 Training the Synthesizer

The release sample of tweets were used to learn the parameters of the generative model. We experimented with several variants of the basic architecture, primarily changing $\phi_{hid}$, where function candidates included the logistic sigmoid $\phi(v) = 1/(1 + e^{-v})$, the linear rectifier $\phi(v) = max(0, v)$, and the softsign $\phi(v) = v/(1 + |v|)$. Other meta-parameter settings explored included the hidden layer size, [50, 100, 200], the learning rate, [0.1, 0.01, 0.001], and the variance of the centered Gaussian distributions used to initialize the input-to-hidden, hidden-to-output, and recurrent weight matrices, [0.1, 0.25, 0.5] (except in the case of rectifier and softsign units–recurrent weights were initialized to a scaled identity matrix as in [10]). Over 50 epochs, stochastic gradient descent (SGD) was applied with mini-batches of size 50 (when applicable, no zero-padding was used, so sequences of length less than $m$ were used as separate sample updates). Truncated back-propagation through time was used to estimate parameter gradients, (window size $m = 10$). We also experimented with an alternative optimizer, RMSProp (with decay set to 0.95). All gradients were clipped to the magnitude range of $[-1, 1]$. All the words in the Twitter sample vocabulary (5978 words total) were used to construct the feature dictionary.

Table 1 shows two samples of the Tweets generated by our neural generative model. The two types of samples presented demonstrate that the proposed model is able to capture certain “styles” or properties of specific users, such as those that write with more hashtags in their tweets versus those write with less. One will notice that the neural generative model seems to generate relatively poor syntax (meaning the data would hurt linguistic-centered utility tasks), which is generally one of the most appraised properties of recurrent language models. We note that this result should not be too surprising, however, since the original Twitter sample rarely contained complete sentences, let alone properly formed syntax (some users had even only written a few characters in sequence). As a result, the neural generative architecture must not only learn how to model the noisy data of the set of users, but must also learn the structure of language from relatively poor exemplars. A strong remedy to this would be to perhaps “pre-train” the language model first on well-formed sentences and phrases before fine-tuning it on Twitter data.

3.2 Risk and Utility Models

In this study we assess the identification risk using machine learning techniques, and we propose a particular setting under which we can quantify risk. We assume the potential attacker has compiled a data set with tweets of users she wants to identify (henceforth $\mathcal{A}$). For each user in $\mathcal{A}$ the attacker trains a binary classifier that separates tweets by this user from tweets by all other users\(^2\).

This formulation of risk deviates slightly from traditional concepts where only direct identifiers and named entities present risk of re-identification in a text. Our framework broadens this, and we believe it sets the bar higher in some ways for the protection required. One potential weakness is that our

\[^1\]Skip-connections were added from the input to the output layer directly but this did not help improve performance. A character-level version yielded poorer results so far–this is the left for future work. We also note that improvement may lie in using gated models, such as the Gated Recurrent Unit architecture.

\[^2\]We evaluated several standard classifier to construct a somewhat realistic risk setting. The final chosen classifier is a linear SVM regularized with the L2 norm.
Identification risk is estimated in the following way. For each release data set \( \mathcal{R} \) (i.e. the original, redacted and synthesized data) and for each user in \( \mathcal{A} \) we use the user specific classifiers trained on \( \mathcal{A} \) to classify each tweet in \( \mathcal{R} \). We then calculate the proportion of tweets that were classified positively for each user. The best guess for who the current user is is then the user that has the highest proportion of positively classified tweets in \( \mathcal{R} \). After doing this for each user in \( \mathcal{A} \) we use the proportion of users that could be correctly identified with this procedure as a measure of risk for the whole dataset. The output of this algorithm is visualized in Figure 2 for a sample of four attacked users and the proportion of identifiable users is given in Table 2.

To assess utility, we compare the original tweets with those that have undergone redaction or are produced by the synthesis models. We use two statistics to estimate utility. First, we compute the distribution of tweet lengths across users for each dataset. If the tweets after redaction or synthesis were drastically longer or shorter, then any analytic models will have significantly different results.
Table 2: Utility results for comparison of original and altered tweet corpus and risk results for user identification probabilities

|                                    | median user tweet length | user cosine similarity MSE from 1 | user identification probabilities |
|------------------------------------|--------------------------|----------------------------------|----------------------------------|
| original tweets                    | 14.85                    | 0.00                             | 0.31                             |
| redacted                           | 16.24                    | 0.49                             | 0.20                             |
| ReluRNN                            | 21.57                    | 0.71                             | 0.01                             |
| RMSsigmRNN                         | 22.68                    | 0.64                             | 0.01                             |
| RMSsoftsignRNN                     | 20.80                    | 0.71                             | 0.01                             |
| sigmRNN                            | 15.84                    | 0.80                             | 0.02                             |
| softsignRNN                        | 17.65                    | 0.30                             | 0.02                             |
| softSignRNNnaive                  | 28.30                    | 0.57                             | 0.06                             |

Second, we calculate the distribution of cosine similarity scores between frequency vectors of words in the original and the altered tweets for users. This measures the relative similarity in the original and altered corpora of words used by a given user. Together these give a good general picture of how closely the altered tweets maintain the distributional qualities of the data. If a researcher were to use the tweets for topic modeling (see [16]) or sentiment analysis (see [15]), common uses for text social media data, maintaining similarity of tweet lengths and words used for specific users would be essential for maintaining similarity of analytic results.

Utility results are given in Figures and Table 2. We compare the average user tweet lengths and cosine similarities between the original tweets and our protection methods of redaction and synthesis. As we can see in the visualization and the table, the softsignRNN model and the redaction achieve the greatest utility, based on the similarity in tweet length distribution and cosine similarity. For the left graph, high utility implies a similarity in location and shape to the original tweet length distribution. On the right graph, high utility implies closeness to 1. We quantify these in Table 2 with the median of the tweet length distributions, and the mean-squared error (MSE) of the cosine similarities with respect to 1 (perfect alignment). The cosine similarity, for most analytic situations, is of more importance than tweet length, thus overall the softsignRNN model shows the best utility. The rest of the synthetic methods performed worse than the redaction only.

It is interesting that using the softsign activation function, inspired by the work of [2], offered the best performance, adding another data-point confirming the value of this activation. Improvement might yet still be further improved by employing quadratic filters as well. We note that it is surprising the RMSProp optimizer underperformed simple SGD, but this could be due to lack of proper tuning of the decay hyper-parameter.

4 Conclusions

This works presents new methods for protection of unstructured text data using a neural-based approach. We find that these models outperform simple redaction both in terms of minimizing the risk of identification of users in the text and maintaining similar term distributions as the original text. Further work should improve upon these synthesis models, which will only increase their appeal as protection mechanisms. In particular we find the softsignRNN model to be the most promising.

To test our methods we use Twitter data, which present some issues for training the language models. These models should improve with a larger corpus of test data or different text data with more complete grammatical structure. Regardless, our results perform well, and this work should be easily extendable to other social media data. Protection of social media data to enable further sharing of research data while maintaining privacy of the individuals in the data is a key step towards ensuring participant trust, reproducible research, and a strong foundation for future research.

References

[1] Barbaro, M., Zeller, T., and Hansell, S. A face is exposed for aol searcher no. 4417749. New York Times 9, 2008 (2006), 8For.
[26] VINYALS, O., AND LE, Q. A neural conversational model. *arXiv preprint arXiv:1506.05869* (2015).

[27] WILLENBORG, L., AND DE WAAL, T. *Statistical disclosure control in practice*. No. 111. Springer Science & Business Media, 1996.

[28] ZIMMER, M. More on the “anonymity” of the facebook dataset—it’s harvard college'. *MichaelZimmer.org Blog* (2008).