usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

A^{4}NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation

Rakshith Shetty, Bernt Schiele, and Mario Fritz, *Max Planck Institute for Informatics*

https://www.usenix.org/conference/usenixsecurity18/presentation/shetty

This paper is included in the Proceedings of the
27th USENIX Security Symposium.

August 15–17, 2018 • Baltimore, MD, USA

978-1-939133-04-5

A^{4}NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation

Rakshith Shetty   Bernt Schiele   Mario Fritz
Max Planck Institute for Informatics
Saarland Informatics Campus
Saarbrücken, Germany
Email: `firstname.lastname@mpi-inf.mpg.de`

Abstract

Text-based analysis methods enable an adversary to reveal privacy relevant author attributes such as gender, age and can identify the text's author. Such methods can compromise the privacy of an anonymous author even when the author tries to remove privacy sensitive content. In this paper, we propose an automatic method, called the Adversarial Author Attribute Anonymity Neural Translation (A^{4}NT), to combat such text-based adversaries. Unlike prior works on obfuscation, we propose a system that is fully automatic and learns to perform obfuscation entirely from the data. This allows us to easily apply the A^{4}NT system to obfuscate different author attributes. We propose a sequence-to-sequence language model, inspired by machine translation, and an adversarial training framework to design a system which learns to transform the input text to obfuscate the author attributes without paired data. We also propose and evaluate techniques to impose constraints on our A^{4}NT model to preserve the semantics of the input text. A^{4}NT learns to make minimal changes to the input to successfully fool author attribute classifiers, while preserving the meaning of the input text. Our experiments on two datasets and three settings show that the proposed method is effective in fooling the attribute classifiers and thus improves the anonymity of authors.

1 Introduction

Natural language processing (NLP) methods including stylometric tools enable identification of authors of anonymous texts by analyzing stylistic properties of the text [1–3]. NLP-based tools have also been applied to profiling users by determining their private attributes like age and gender [4]. These methods have been shown to be effective in various settings like blogs, reddit comments, twitter text [5] and in large scale settings with up to 100,000 possible authors [6]. In a recent famous case, authorship attribution tools were used to help confirm J.K Rowling as the real author of *A Cuckoo's Calling* which was written by Ms. Rowling under pseudonymity [7]. This case highlights the privacy risks posed by these tools.

Apart from the threat of identification of an anonymous author, the NLP-based tools also make authors susceptible to profiling. Text analysis has been shown to be effective in predicting age group [8], gender [9] and to an extent even political preferences [10]. By determining such private attributes an adversary can build user profiles which have been used for manipulation through targeted advertising, both for commercial and political goals [11].

Since the NLP based profiling methods utilize the stylistic properties of the text to break the authors anonymity, they are immune to defense measures like pseudonymity, masking the IP addresses or obfuscating the posting patterns. The only way to combat them is to modify the content of the text to hide stylistic attributes. Prior work has shown that while people are capable of altering their writing styles to hide their identity [12], success rate depends on the authors skill and doing so consistently is hard for even skilled authors [13]. Currently available solutions to obfuscate authorship and defend against NLP-methods has been largely restricted to semi-automatic solutions which suggest possible changes to the user [14] or hand-crafted transformations to text [15] which need re-engineering on different datasets. This however limits the applicability of these defensive measures beyond the specific dataset it was designed on. To the best of our knowledge, text rephrasing using generic machine translation tools [16] is the only prior work offering a fully automatic solution to author obfuscation which can be applied across datasets. But as found in prior work [17] and further demonstrated with our experiments, generic machine translation based obfuscation fails to sufficiently hide the identity and protect against attribute classifiers.

Additionally the focus in prior research has been towards protecting author identity. However, obfuscating identity does not guarantee protection of private attributes like age and gender. Determining attributes is generally easier than predicting the exact identity for NLP-based adversaries, mainly due to former being small closed-set prediction task compared to later which is larger and potentially open-set prediction task. This makes obfuscating attributes a difficult but an important problem.

Our work. We propose an unified automatic system (A^{4}NT) to obfuscate authors text and defend against NLP adversaries. A^{4}NT follows the imitation model of defense discussed in [12] and protects against various attribute classifiers by learning to imitate the writing style of a target class. For example, A^{4}NT learns to hide the gender of a female author by re-synthesizing the text in the style of the male class. This imitation of writing style is learned by adversarially training [18] our style-transfer network against the attribute classifier. Our A^{4}NT network learns the target style by learning to fool the authorship classifiers into misclassifying the text it generates as target class. This style transfer is accomplished while aiming to retain the semantic content of the input text.

Unlike many prior works on authorship obfuscation [14, 15], we propose an end-to-end learnable author anonymization solution, allowing us to apply our method not only to authorship obfuscation but to the anonymization of different author attributes including identity, gender and age with a *unified approach*. We illustrate this by successfully applying our model on three different attribute anonymization settings on two different datasets. Through empirical evaluation, we show that the proposed approach is able to fool the author attribute classifiers in all three settings effectively and better than the baselines. While there are still challenges to overcome before applying the system to multiple attributes and situations with very little data, we believe that A^{4}NT offers a new data driven approach to authorship obfuscation which can easily adapt to improving NLP-based adversaries.

Technical challenges: We design our A^{4}NT network architecture based on the sequence-to-sequence neural machine translation model [19]. A key challenge in learning to perform style transfer, compared to other sequence-to-sequence mapping tasks like machine translation, is the lack of paired training data. Here, paired data refers to datasets with both the input text and its corresponding ground-truth output text. In obfuscation setting, this means having a large dataset with semantically same sentences written in different styles corresponding to the attributes we want to hide. Such paired data is infeasible to obtain and this has been a key hurdle in developing automatic obfuscation methods. Some prior attempts to perform text style transfer required paired training data [20] and hence were limited in their applicability beyond toy-data settings. We overcome this by training our A^{4}NT network within a generative adversarial networks (GAN) [18] framework. GAN framework enables us to train the A^{4}NT network to generate samples that match the target distribution without need for paired data.

We characterize the performance of our A^{4}NT network along two axes: privacy effectiveness and semantic similarity. Using automatic metrics and human evaluation to measure semantic similarity of the generated text to the input, we show that A^{4}NT offers a better trade-off between privacy effectiveness and semantic similarity. We also analyze the effectiveness of A^{4}NT for protecting anonymity for varying degrees of input text "difficulty".

Contributions: In summary, the main contributions of our paper are. **(1):** We propose a novel approach to authorship obfuscation that uses a style-transfer network (A^{4}NT) to automatically transform the input text to a target style and fool the attribute classifiers. The network is trained without paired data by adversarial training. **(2):** The proposed obfuscation solution is end-to-end trainable, and hence can be applied to protect different author attributes and on different datasets with no changes to the overall framework. **(3):** Quantifying the performance of our system on privacy effectiveness and semantic similarity to input, we show that it offers a better trade-off between the two metrics compared to baselines.

2 Related Work

In this section, we review prior work relating to four different aspects of our work – author attribute detection (our adversaries), authorship obfuscation (prior work), machine translation (basis of our A^{4}NT network) and generative adversarial networks (training framework we use).

Authorship and attribute detection Machine learning approaches, where a set of text features are input to a classifier which learns to predict the author, have been popular in recent author attribution works [2]. These methods have been shown to work well on large datasets [6], duplicate author detection [21] and even on non-textual data like code [22]. Sytlometric models can also be applied to determine private author attributes like age or gender [4].

Classical author attribution methods rely on a predefined set of features extracted from the input text [23]. Recently deep-learning methods have been applied to learn to extract the features directly from data [3, 24]. [24] uses a multi-headed recurrent neural network (RNN) to train a generative language model on each author's text and use the model's perplexity on the test document to predict the author. Alternatively, [3] uses convolutional neural network (CNN) to train an author classifiers. To show generality of our A^{4}NT network, we test it against both RNN and CNN based author attribute classifiers.

Authorship obfuscation Authorship obfuscation methods are adversarial in nature to stylometric methods of author attribution; they try to change the style of the input text so that the author identity is not discernible. The majority of prior works on author attribution are semi-automatic [14, 25], where the system suggests authors to make changes to the document by analyzing the stylometric features. The few available automatic obfuscation methods have relied on general rephrasing methods like generic machine translation [16] or on predefined text transformations [26]. Round-trip machine translation, where input text is translated to multiple languages one after the other until it is translated back to the source language, is proposed as an automatic method of obfuscation in [16]. Recent work [26] obfuscates text by moving the stylometric features towards the average values on the dataset by applying pre-defined transformations on input text.

We propose the first method to achieve fully automatic obfuscation using text style transfer. This style transfer is not pre-defined but learnt directly from data optimized for fooling attribute classifiers. This allows us to apply our model across datasets without extra engineering effort.

Machine translation The task of style-transfer of text data shares similarities with the machine translation problem. Both involve mapping an input text sequence onto an output text sequence. Style transfer can be thought of as machine translation on the same language.

Large end-to-end trainable neural networks have become a popular choice in machine translation [27, 28]. These methods are generally based on sequence-to-sequence recurrent models [19] consisting of two networks, an encoder which encodes the input sentence into a fixed size vector and a decoder which maps this encoding to a sentence in the target language.

We base our A^{4}NT network architecture on the word-level sequence-to-sequence language model [19]. Neural machine translation systems are trained with large amounts of paired training data. However, in our setting, obtaining paired data of the same text in different writing styles is not viable. We overcome the lack of paired data by casting the task as matching style distributions instead of matching individual sentences. Specifically, our A^{4}NT network takes an input text from a source distribution and generates text whose style matches the target attribute distribution. This is learnt without paired data using distribution matching methods. This reformulation allows us to demonstrate the first successful application of the machine translation models to the obfuscation task.

Generative adversarial networks Generative Adversarial Networks (GAN) [18] are a framework for learning a generative model to produce samples from a target distribution. It consists of two models, a generator and a discriminator. The discriminator network learns to distinguish between the generated samples and real data samples. Simultaneously, the generator learns to fool this discriminator network thereby getting closer to the target distribution. In this two-player game, a fully optimized generator perfectly mimics the target distribution [18].

We train our A^{4}NT network within the GAN framework, directly optimizing A^{4}NT to fool the attribute classifiers by matching style distribution of the target class. A recent approach to text style-transfer proposed in [29] also utilizes GANs to perform style transfer using unpaired data. However, the solution proposed in [29] changes the meaning of the input text significantly during style transfer and is applied on the sentiment transfer task. In contrast, authorship obfuscation task requires the generated text to preserve the semantics of the input. We address this problem by proposing two methods to improve the semantic consistency between the input and the A^{4}NT output.

Attacks against machine-learning models: Recent works have shown that machine learning models are susceptible to attacks by adversaries which can manipulate the input of these models [30–32]. By adding only a small amount of perturbation to the input image, barely noticeable to the human eye, the adversary can fool state-of-the art image classifiers to wrongly classify the input [30, 31]. Adding adversarial perturbation to images has also been proposed as a means of protecting the users' privacy [33]. While large portion of research on adversarial perturbations has focused on the image domain, few recent works have shown that one can also fool NLP classifiers by deleting, adding or replacing few salient words [34, 35] and by adding whole sentences unrelated to the topic of the document [36]. However, while the focus of these works is to fool the NLP classifiers with producing realistic text, there is no consideration to whether the meaning of the input text is preserved. Additionally the transformations performed are restricted to the predefined classes like add, remove or replace, with independently tuned heuristics for each of these transformations. In contrast, we propose a machine translation model which automatically learns to transform the input text appropriately to fool the attribute classifiers, while aiming to preserve the meaning of the input text.

3 Threat Model

In our target scenario, our user is faced with an adversary who can access the text written by the user and the adversary wishes to determine the user's private attributes for identification or for profiling. We assume that the author has taken care to remove obvious identifiable features from the text like name, zip code, IP address etc. The adversary has to rely on stylistic properties of the text for the analysis. To aid with this analysis, adversary can train NLP models on large amount of publicly available data, for example blog dataset [37], twitter dataset [38]. In this scenario, the proposed A^{4}NT system enables automatic obfuscation of user's writing style to hide any desired private attribute like age group, gender or identity.

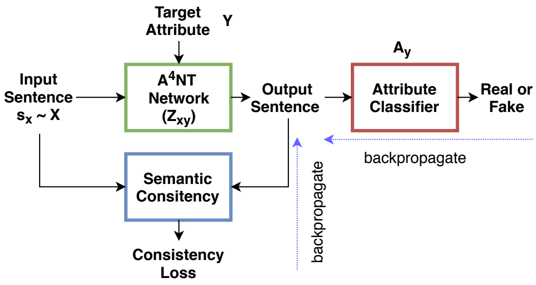

Figure 1: GAN framework to train our A^{4}NT network. Input sentence is transformed by A^{4}NT to match the style of the target attribute. This output is evaluated using the attribute classifier and semantic consistency loss. A^{4}NT is trained by backpropagating through these losses.

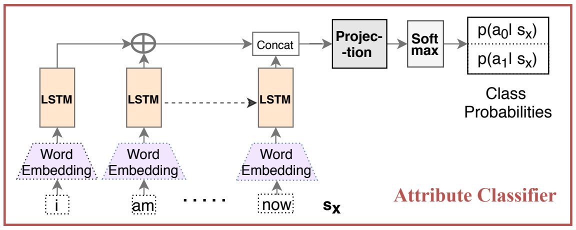

Figure 2: Block diagram of the attribute classifier network. The LSTM encoder embeds the input sentence into a vector. Sentence encoding is passed to linear projection followed by softmax layer to obtain class probabilities

4 Author Attribute Anonymization

We propose an author adversarial attribute anonymizing neural translation (A^{4}NT) network to defend against NLP-based adversaries. The proposed solution includes the A^{4}NT Network , the adversarial training scheme, and semantic and language losses to learn to protect private attributes. The A^{4}NT network transforms the input text from a source attribute class to mimic the style of a different attribute class, and thus fools the attribute classifiers.

Technically, A^{4}NT network is essentially solving a sequence to sequence mapping problem — from text sequence in the source domain to text in the target domain — similar to machine translation. Exploiting this similarity, we design our A^{4}NT network based on the sequence-to-sequence neural language models [19], widely used in neural machine translation [27]. These models have proven effective when trained with large amounts of paired data and are also deployed commercially [28]. If there were paired data in source and target attributes, we could train our A^{4}NT network exactly like a machine translation model, with standard supervised learning. However, such paired data is infeasible to obtain as it would require the same text written in multiple styles.

To address the lack of paired data, we cast the anonymization task as learning a generative model, $Z_{xy}(s_x)$, which transforms an input text sample s_x drawn from source attribute distribution $s_x \sim X$, to look like samples from the target distribution $s_y \sim Y$. This formulation enables us to train the A^{4}NT network $Z_{xy}(s_x)$ with the GAN framework to produce samples close to the target distribution Y, using only unpaired samples from X and Y. Figure 1 shows this overall framework.

The GAN framework consists of two models, a generator producing synthetic samples to mimic the target data distribution, and a discriminator which tries to distinguish real data from the synthesized "fake" samples from the generator. The two models are trained adversarially, i.e. the generator tries to fool the discriminator and the discriminator tries to correctly identify the generated samples. We use an attribute classifier as the discriminator and the A^{4}NT network as the generator. The A^{4}NT network, in trying to fool the attribute classification network, learns to transform the input text to mimic the style of the target attribute and protect the attribute anonymity.

For our A^{4}NT network to be a practically useful defensive measure, the text output by this network should be able to fool the attribute classifier while also preserving the meaning of the input sentence. If we could measure the semantic difference between the generated text and the input text it could be used to penalize deviations from the input sentence semantics. Computing this semantic distance perfectly would need true understanding of the meaning of input sentence, which is beyond the capabilities of current natural language processing techniques. To address this aspect of style transfer, we experiment with various proxies to measure and penalize changes to input semantics, which will be discussed in Section 4.4. Following subsections will describe each module in detail.

4.1 Author Attribute Classifiers

We build our attribute classifiers using neural networks that predict the attribute label by directly operating on the text data. This is similar to recent approaches in authorship recognition [3, 24] where, instead of hand-crafted features used in classical stylometry, neural networks are used to directly predict author identity from raw text data. However, unlike in these prior works, our focus is attribute classification and obfuscation. We train our classifiers with recurrent networks operating at word-level, as opposed to character-level models used in [3, 24] for two reasons. We found that the word-level models give good performance on all three attribute-classification tasks we experiment with (see Section 6.1). Additionally, they are much faster than character-level models, making it feasible to use them in GAN training described in Section 4.2.

Specifically, our attribute classifier A_x to detect attribute value x is shown in Figure 2. It consists of a Long-Short Term Memory (LSTM) [39] encoder network to compute an embedding of the input sentence into a fixed size vector. It learns to encode the parts of the sentence most relevant to the classification task into the embedding vector, which for attribute prediction is mainly the stylistic properties of the text. This embedding is input to a linear layer and a softmax layer to output the class probabilities.

Given an input sentence $s_x = \{w_0, w_1, \cdots, w_{n-1}\}$, the words are one-hot encoded and then embedded into fixed size vectors using the word-embedding layer shown in Figure 2 to obtain vectors $\{v_0, v_1, \cdots, v_{n-1}\}$. The word embedding layer is simply a matrix of $V \times d_{wv}$ containing the word vectors of d_{wv} dimensions for each word in the vocabulary of size V. This matrix is multiplied with the one-hot encoding of the word to obtain the representation of the corresponding word. The learned word vectors encode the similarities between words and can help deal with large vocabulary sizes. The word vectors are randomly initialized and then learned from the data during the training of the model. This approach works better than using pre-trained word vectors like word2vec [40] or Glove [41] since the learned word-vectors can encode similarities most relevant to the attribute classification task at hand.

This sequence of word vectors is recursively passed through an LSTM to obtain a sequence of outputs $\{h_0, h_1, \cdots, h_{n-1}\}$. We refer the reader to [39] for the exact computations performed to get the LSTM output.

Sentence embeddings are obtained by concatenating the final LSTM output and the mean of the LSTM outputs from other time-steps.

$$E(s_x) = \left[h_{n-1}; \frac{1}{n-1} \sum h_{n-1}\right] \qquad (1)$$

At the last time-step the LSTM network has seen all the words in the sentence and can encode a summary of the sentence in its output. However, using LSTM outputs from all time-steps, instead of just the final one, speeds up training due to improved flow of gradients through the network. Finally, $E(s_x)$ is passed through linear and softmax layers to obtain class probabilities, for each class c_i. The network is then trained using cross-entropy loss.

$$p_{\text{auth}}(c_i|s_x) = \text{softmax}(W \cdot E(s_x)) \qquad (2)$$

$$\text{Loss}(A_x) = \sum_i t_i(s_x) \log\left(p_{\text{auth}}(c_i|s_x)\right) \qquad (3)$$

where $t(s_x)$ is the one-hot encoding of the true class of s_x.

The same network architecture is applied for all our attribute prediction tasks including identity, age and gender.

4.2 The A^{4}NT Network

A key design goal for the A^{4}NT network is that it is trainable purely from data to obfuscate the author attributes. This is a significant departure from prior works on author obfuscation [14, 26] that rely on hand-crafted rules for text modification to achieve obfuscation. The methods relying on hand-crafted rules are limited in applicability to specific datasets they were designed for.

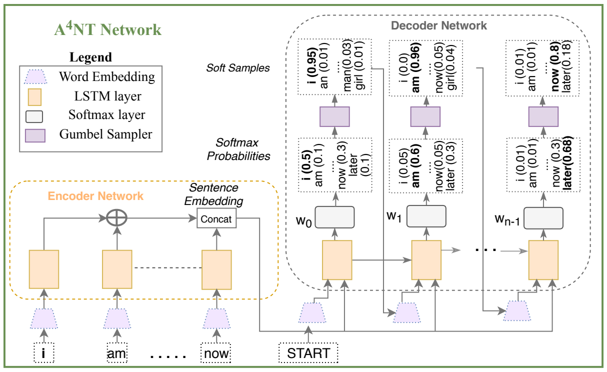

Figure 3: Block diagram of the A^{4}NT network. First LSTM encoder embeds the input sentence into a vector. The decoder maps this sentence encoding to the output sequence. Gumbel sampler produces "soft" samples from the softmax distribution to allow backpropagation.

To achieve this goal, we base our A^{4}NT network Z_{xy}, shown in Figure 3, on a recurrent sequence-to-sequence neural translation model [19] (*Seq2Seq*) popular in many sequence mapping tasks. As seen from the wide-range of applications mapping text-to-text [27], speech-to-text [42], text-to-part of speech [43], the *Seq2Seq* models can effectively learn to map input sequences to arbitrary output sequences, with appropriate training. They operate on raw text data and alleviate the need for hand-crafted features or rules to transform the style of input text, predominantly used in prior works on author obfuscation [14, 26]. Instead, appropriate text transformations can be learnt directly from data. This flexibility allows us to easily apply the same A^{4}NT network and training scheme to different datasets and settings.

The A^{4}NT network Z_{xy} consists of two components, an encoder and a decoder modules, similar to standard sequence-to-sequence models. The encoder embeds the variable length input sentence into a fixed size vector space. The decoder maps the vectors in this embedding space to output text sequences in the target style. The encoder is an LSTM network, sharing the architecture of the sentence encoder in Section 4.1. The same architecture applies here as the task here is also to embed the input sentence s_x into a fixed size vector $E_G(s_x)$. However, $E_G(s_x)$ should learn to represent the semantics of the input sentence allowing the decoder network to generate a sentence with similar meaning but in a different style.

The sentence embedding from the encoder is input to the decoder LSTM which generates the output sentence one word at a time. At each step t, the decoder LSTM takes $E_G(s_x)$ and the previous output word w^o_{t-1}

to produce a probability distribution over the vocabulary. Sampling from this distribution outputs the next word.

$$h_t^{\text{dec}}(s_x) = \text{LSTM}\left[E_G(s_x), W_{\text{emb}}(\tilde{w}_{t-1})\right] \tag{4}$$

$$p(\tilde{w}_t|s_x) = \text{softmax}_V(W_{\text{dec}} \cdot h_t^{\text{dec}}(s_x)) \tag{5}$$

$$\tilde{w}_t = \text{sample}(p(\tilde{w}_t|s_x)) \tag{6}$$

where W_{emb} is the word embedding, W_{dec} matrix maps the LSTM output to vocabulary size and V is the vocabulary.

In most applications of *Seq2Seq* models, the networks are trained using parallel training data, consisting of input and ground-truth output sentence pairs. A sentence is input to the encoder and propagated through the network and the network is trained to maximize the likelihood of generating the paired ground-truth output sentence. However, in our setting, we do not have access to such parallel training data of text in different styles and the A^{4}NT network Z_{xy} is trained in an unsupervised setting.

We address the lack of parallel training data by using the GAN framework to train the A^{4}NT network. In this framework, the A^{4}NT network Z_{xy} learns by generating text samples and improving itself iteratively to produce text that the attribute classifier, A_y, classifies as target attribute. A benefit of GANs is that the A^{4}NT network is directly optimized to fool the attribute classifiers. It can hence learn to make transformations to the parts of the text which are most revealing of the attribute at hand, and so hide the attribute with minimal changes.

However, to apply the GAN framework, we need to differentiate through the samples generated by Z_{xy}. The word samples from $p(\tilde{w}_t|s_x)$ are discrete tokens and are not differentiable. Following [44], we apply the Gumbel-Softmax approximation [45] to obtain differentiable soft samples and enable end-to-end GAN training. See Appendix A for details.

Splitting decoder: To transfer styles between attribute pairs, x and y, in both directions, we found it ineffective to use the same network Z_{xy}. A single network Z_{xy} is unable to sufficiently switch its output word distributions solely on a binary condition of target attribute. Nonetheless, using a separate network for each ordered pair of attributes is prohibitively expensive. A good compromise we found is to share the encoder to embed the input sentence but use different decoders for style transfer between each ordered pair of attributes. Sharing the encoder allows the two networks to share a significant number of parameters and enables the attribute specific decoders to deal with the words found only in the vocabulary of the other attribute group using shared sentence and word embeddings.

4.3 Style Loss with GAN

We train the two A^{4}NT networks Z_{xy} and Z_{yx} in the GAN framework to produce samples which are indistinguishable from samples from distributions of attributes y and x respectively, without having paired sentences from x and y. Figure 4 shows this training framework.

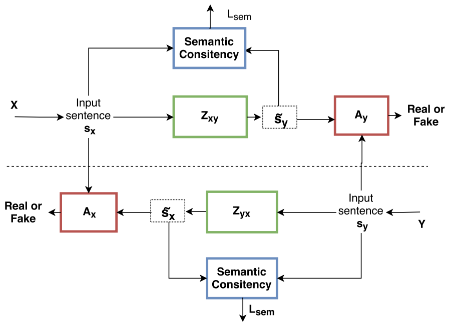

Figure 4: Illustrating use of GAN framework and cyclic semantic loss to train a pair of A^{4}NT networks.

Given a sentence s_x written by author with attribute x, the A^{4}NT network outputs a sentence $\tilde{s_y} = Z_{xy}(s_x)$. This is passed to the attribute classifier for attribute y, A_y, to obtain probability $p_{\text{auth}}(y|\tilde{s_y})$. Z_{xy} tries to fool the classifier A_y into assigning high probability to its output, whereas A_y tries to assign low probability to sentences produced by Z_{xy} while assigning high probability to real sentences s_y written by y. The same process is followed to train the A^{4}NT network from y to x, with x and y swapped. The loss functions used to train the A^{4}NT network and the attribute classifiers in this setting is given by:

$$L(A_y) = -\log\left(p_{\text{auth}}(y|s_y)\right) - \log\left(1 - p_{\text{auth}}(y|\tilde{s_y})\right) \tag{7}$$

$$L_{\text{style}}(Z_{xy}) = -\log\left(p_{\text{auth}}(y|\tilde{s_y})\right) \tag{8}$$

The two networks Z_{xy} and A_y are adversarially competing with each other when minimizing the above loss functions. At optimum it is guaranteed that the distribution of samples produced by Z_{xy} is identical to the distribution of y [18]. However, we want the A^{4}NT network to only imitate the style of y, while keeping the content from x. Thus, we explore methods to enforce the semantic consistency between the the input sentence and the A^{4}NT output.

4.4 Preserving Semantics

We want the output sentence, $\tilde{s_y}$, produced by $Z_{xy}(s_x)$ to not only fool the attribute classifier, but also to preserve the meaning of the input sentence s_x. We propose a semantic loss $L_{\text{sem}}(\tilde{s_y}, s_x)$ to quantify the meaning changed during the anonymization by A^{4}NT . Simple approaches like matching words in $\tilde{s_y}$ and s_x can severely limit the effectiveness of anonymization, as it penalizes even synonyms or alternate phrasing. In the following subsection we will discuss two approaches to define L_{sem}, and later in Section 6 we compare these approaches quantitatively.

4.4.1 Cycle Constraints

One could evaluate how semantically close is $\tilde{s_y}$ to s_x by evaluating how easy it is to reconstruct s_x from

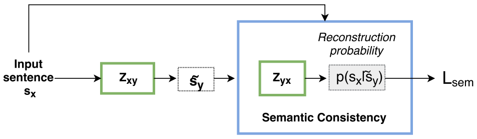

Figure 5: Semantic consistency in A[4]NT networks is enforced by maximizing cyclic reconstruction probability.

$\tilde{s_y}$. If $\tilde{s_y}$ means exactly the same as s_x, there should be no information loss and we should be able to perfectly reconstruct s_x from $\tilde{s_y}$. We could use the A^{4}NT network in the reverse direction to obtain a reconstruction, $\ddot{s_x} = Z_{yx}(\tilde{s_y})$ and compare it to input sentence s_x. Such an approach, referred to as cycle constraint, has been used in image style transfer [46], where l_1 distance is used to compare the reconstructed image and the original image to impose semantic relatedness penalty. However, in our case l_1 distance is not meaningful to compare $\ddot{s_x}$ and s_x, as they are sequences of possibly different lengths. Even a single word insertion or deletion in $\ddot{s_x}$ can cause the entire sequence to mismatch and be penalized by the l_1 distance.

A simpler and more stable alternative we use is to forgo the reconstruction and just computing the likelihood of reconstruction of s_x when applying reverse style-transfer on $\tilde{s_y}$. This likelihood is simple to obtain from the reverse A^{4}NT network Z_{yx} using the word distribution probabilities at the output. This cyclic loss computation is illustrated in Figure 5. Duly, we compute reconstruction probability $P_r(s_x|\tilde{s_y})$ and define the semantic loss as:

$$P_r(s_x|\tilde{s_y}) = \prod_{t=0}^{n-1} p_{z_{yx}}(w_t|\tilde{s_y}) \tag{9}$$

$$L_{\text{sem}}(\tilde{s_y}, s_x) = -\log P_r(s_x|\tilde{s_y}) \tag{10}$$

The lower the semantic loss L_{sem}, the higher the reconstruction probability and thus more meaning of the input sentence s_x is preserved in the style-transfer output $\tilde{s_y}$.

4.4.2 Semantic Embedding Loss

An alternative approach to measuring the semantic loss is to embed the two sentences, $\tilde{s_y}$ and s_x, into a semantic space and compare the two embedding vectors using l_1 distance. The idea is that a semantic embedding method puts similar meaning sentences close to each other in this vector space. This approach is used in many natural language processing tasks, for example in semantic entailment [47]

Since we do not have annotations of semantic relatedness on our datasets, it is not possible to train a semantic embedding model but instead we have to rely on pre-trained models known to have good transfer learning performance. Several such semantic sentence embeddings are available in the literature [47, 48]. We use the universal sentence embedding model from [47], pre-trained on the Stanford natural language inference dataset [49].

We embed the two sentences using this semantic embedding model F and use the l_1 distance to compare the two embeddings and define the semantic loss as:

$$L_{\text{sem}}(\tilde{s_y}, s_x) = \sum_{dim} \left|F(s_x) - F(\tilde{s_y})\right| \tag{11}$$

4.5 Smoothness with Language Loss

The A^{4}NT network can minimize the style and the semantic losses, while still producing text which is broken and grammatically incorrect. To minimize the style loss the A^{4}NT network needs to add words typical of the target attribute style. While minimizing the semantic loss, it needs to retain the semantically relevant words from the input text. However neither of these two losses explicitly enforces correct grammar and word order of $\tilde{s}$.

On the other hand, unconditional neural language models are good at producing grammatically correct text. The likelihood of the sentence produced by our A^{4}NT model $\tilde{s}$ under an unconditional language model, M_y, trained on the text by target attribute authors y, is a good indicator of the grammatical correctness of $\tilde{s}$. The higher the likelihood, the more likely the generated text $\tilde{s}$ has syntactic properties seen in the real data. Therefore, we add an additional language smoothness loss on $\tilde{s}$ in order to enforce Z to produce syntactically correct text.

$$L_{\text{lang}}(\tilde{s}) = -\log M_y(\tilde{s}) \tag{12}$$

Overall loss function: The A^{4}NT network is trained with a weighted combination of the three losses: style loss, semantic consistency and language smoothing loss.

$$L_{\text{tot}}(Z_{xy}) = w_{\text{sty}} L_{\text{style}} + w_{\text{sem}} L_{\text{sem}} + w_l L_{\text{lang}} \tag{13}$$

We chose the above three weights so that the magnitude of the weighted loss terms are approximately equal at the beginning of training. Model training was not sensitive to exact values of the weights chosen that way.

Implementation details: We implement our model using the PyTorch framework [50]. The networks are trained by optimizing the loss functions described with stochastic gradient descent using the RMSprop algorithm [51]. The A^{4}NT network is pre-trained as an autoencoder, i.e to reconstruct the input sentence, before being trained with the loss function described in (13). During the GAN training, the A^{4}NT network and the attribute classifiers are trained for one minibatch each alternatively. We will open source our code, models and data at the time of publication.

5 Experimental Setup

We test our A^{4}NT network on obfuscation of three different attributes of authors on two different datasets. The three attributes we experiment with include author's age (under 20 vs over 20), gender (male vs female authors), and author identities (setting with two authors).

5.1 Datasets

We use two real world datasets for our experiments: Blog Authorship corpus [37] and Political Speech dataset. The datasets are from very different sources with distinct language styles, the first being from mini blogs written by several anonymous authors, and the second from political speeches of two US presidents Barack Obama and Donald Trump. This allows us to show that our approach works well across very different language corpora.

Blog dataset: The blog dataset is a large collection of micro blogs from blogger.com collected by [37]. The dataset consists of 19,320 "documents" along with annotation of author's age, gender, occupation and star-sign. Each document is a collection of all posts by a single author. We utilize this dataset in two different settings; split by gender (referred to as blog-gender setting) and split by age annotation (blog-age setting). In the blog-age setting, we group the age annotations into two groups, teenagers (age between 13-18) and adults (age between 23-45) to obtain data with binary age labels. Age-groups 19-22 are missing in the original dataset. Since the dataset consists of free form text written while blogging with no proper sentence boundaries markers, we use the Stanford CoreNLP tool to segment the documents into sentences. All numbers are replaced with the NUM token. For training and evaluation, the whole dataset is split into training set of 13,636 documents, validation set of 2,799 documents and test set of 2,885 documents.

Political speech dataset: To test the limits of how far style imitation based anonymization can help protect author identity, we also test our model on two well known political figures with very different verbal styles. We collected the transcriptions of political speeches of Barack Obama and Donald Trump made available by the The American Presidency Project [52]. While the two authors talk about similar topics they have highly distinctive styles and vocabularies, making it a challenging dataset for our A^{4}NT network. The dataset consists of 372 speeches, with about 65,000 sentences in total as shown in Table I. We treat each speech as a separate document when evaluating the classification results on the document-level. This dataset contains a significant amount of references to named entities like people, organizations, etc. To avoid that both attribute classifiers and the style transfer model rely on these references to specific people, we use the Stanford Named Entity Recognizer tool [53] to identify and replace these entities with entity labels. The dataset is split into training set of 250 speeches, validation set of 49 speeches and test set of 73 speeches.

The comparison of the two datasets can be found in Table I. The blog dataset is much larger and therefore we run most of our evaluation on it. Using these two datasets, we evaluate our model in three different attribute obfuscation settings, namely age (blog-age), gender (blog-gender) and identity obfuscation (speech dataset). Detailed analysis of our model presented in Section 6.2 is done on the validation split of the blog dataset, in the blog-age setting, containing 2,799 documents and 518,268 sentences.

Dataset	Attributes	# Documents	# Sentences	# Vocabulary
Speech	Identity	372	65k	5.6k
Blog	Age, Gender	19320	3.38 Mil	22k

Table I: Comparing statistics of the two datasets.

5.2 Evaluation Methods

We evaluate our models with according to two different aspects: **(a)** *Privacy effectiveness*, to measure how well the A^{4}NT network can fool the attribute classifiers and **(b)** *Semantic similarity*, quantifying how much of the input sentence semantics is preserved after style transfer.

To quantify privacy effectiveness we compare the F1-score of the attribute classifiers on the original data and the generated text. To mimic a real-world setting where we will not have access to the parameters of the attribute-classifier used by the adversary to train our A^{4}NT network, we test it against attribute classifiers which have been separately trained on real data.

The F1-score for the attribute classifiers is defined as:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{14}$$

With successful style transfer, the attribute classifier will misclassify generated text and the F1-scores will drop. We consider the obfuscation to be successful if the F1-score drops below that of a random guessing classifier. Note that since the datasets are not perfectly balanced with respect to the binary attributes, the F1-score for random classifier is not 0.5. The expected document-level F1-scores of random guessing are 0.54 for the blog-age setting, 0.5 for the blog-gender setting, and 0.61 for the speech data.

For achieving obfuscation, having the attribute classifier around random chance-level is more desirable than pushing the classifier to 0% accuracy. Zero accuracy means the classifier has perfectly misclassified every sample. In a binary classification setting like age-group or gender, if the adversary detects there has been obfuscation he can fully recover the original classes by just flipping the decisions of the classifier. However if the obfuscation is to the random chance-level, even upon detection, the best the adversary can do is to get the random chance accuracy. This also holds true for multi-class situation, as can be seen from the information theoretic point of view. To achieve perfect obfuscation, we want the attribute classifier output to contain minimum information about the true class of the input text. When the classifier accuracy of the k-class attribute classifier is at the random chance-level, it is guessing the class labels with uniform probability $p(y|c) \sim \text{Uniform}(1,2,\cdots,k)$.

In this case the mutual information between the classifier predicted label y and true label c is zero, since the $p(y|c) = p(y)$. However, the prediction of classifier $p(y|c)$ at 0% accuracy is not independent of the input class-label since it cannot take the correct class value c, i.e $p(y|c) \sim \text{Uniform}(1,2,\cdots,c-1,c+1,\cdots,k)$. This leads to non-zero mutual information between y and c. Hence, we use the random chance-level as our success criteria for obfuscation instead of targeting 0% classifier accuracy.

To quantify semantic similarity, we use the meteor metric [54]. It is used in machine translation and image captioning to evaluate the similarity between the candidate text and a reference text. Meteor compares the candidate text to one or more references by matching n-grams, while allowing for soft matches using synonym and paraphrase tables. Meteor score lies between zero and one with zero indicating no similarity and one indicating identical sentences. For a point of reference, the state-of-the-art methods for paraphrase generation task achieve meteor scores between 0.35-0.4 [55] and for multimodal machine translation task achieve meteor score in the range 0.5-0.55 [56]. We use the meteor score between the generated and input text as the measure of semantic similarity.

However, the automatic evaluation for semantic similarity is not perfectly correlated with human judgments, especially with few reference sentences. To address this, we additionally conduct two user studies on a subset of the test data of 745 sentences, first to compare the semantic similarity between different obfuscation methods relatively, and second to measure the semantic similarity between the model output and input text on an absolute scale. We ask human annotators on Amazon Mechanical Turk (AMT) to judge the semantic similarity of the generated text from our models. No other information was collected from the annotators, thereby keeping them anonymous. The annotators were compensated for their work through the AMT system. We manually screened the text shown to the annotators to make sure it contained no obvious offensive content.

5.3 Baselines

We use the two baseline methods below to compare our model with. Both chosen baselines are automatic obfuscation methods not relying on hand-crafted rules.

Autoencoder We train our A^{4}NT network Z as an autoencoder, where it takes as input s_x and tries to reproduce it from the encoding. The autoencoder is trained similar to a standard neural language model with cross entropy loss. We train two such auto-encoders Z_{xx} and Z_{yy} for the two attributes. Now simple style transfer can be achieved from x to y by feeding the sentence s_x to the autoencoder of the other attribute class Z_{yy}. Since Z_{yy} is trained to output text in the y domain, the sentence $Z_{yy}(s_x)$ tends to look similar to sentences in y. This model sets the baseline for style transfer that can be achieved without cross domain training using GANs, with the same network architecture and the same number of parameters.

Google machine translation: A simple and accessible approach to change writing style of a piece of text without hand designed rules is to use generic machine translation software. The input text is translated from a source language to multiple intermediate languages and finally translating back to the source language. The hope is that through this round-trip the style of the text has changed, with the meaning preserved. This approach was used in the PAN authorship obfuscation challenge recently [16].

We use the Google machine translation service[1] to perform the round-trip translation of our input sentences. We have tried a varying number of intermediate languages, results of which will be discussed in Section 6. Since Google limits the api calls and imposes character limits on manual translation, we use this baseline only on the subset of 745 sentences from the test set for human evaluation.

6 Experimental Results

We test our model on the three settings discussed in Section 5 with the goal to understand if the proposed A^{4}NT network can fool the attribute classifiers to protect the anonymity of the author attributes. Through quantitative evaluation done in Section 6.1, we show that this is indeed the case: our A^{4}NT network learns to fool the attribute classifiers across all three settings. We compare the two semantic loss functions presented in Section 4.4 and show that the proposed reconstruction likelihood loss does better than pre-trained semantic encoding.

However, this privacy gain comes with a trade-off. The semantics of the input text is sometimes altered. In Section 6.2, using qualitative examples, we analyze the failure modes of our system and identify limits up to which style-transfer can help preserve anonymity.

We use three variants of our model in the following study. The first model uses the semantic encoding loss described in Section 4.4.2 and is referred to as *FBsem*. The second uses the reconstruction likelihood loss discussed in Section 4.4.1 instead, and is denoted by *CycML*. Finally, *CycML+Lang* uses both cyclic maximum likelihood and the language smoothing loss described in Section 4.5.

6.1 Quantitative Evaluation

Before analyzing the performance of our A^{4}NT network, we evaluate the attribute classifiers on the three settings we use. For this, we train the attribute classifier model in Section 4.1 on all three settings. Table II shows the F1-scores of the attribute classifiers on the training and the validation splits of the blog and the speech datasets. Document-level scores are

[1] https://translate.google.com/

Setting	Training Set		Validation Set	
	Sentence	Document	Sentence	Document
Speechdata	0.84	1.00	0.68	1.00
Blog-age	0.76	0.92	0.74	0.88
Blog-gender	0.64	0.93	0.52	0.75

Table II: F1-scores of the attribute classifiers. All of them do well and better than the document-level random chance (0.62 for speech), (0.53 for age), and (0.50 for gender).

obtained from accumulating the class log-probability scores on each sentence in a document before picking the maximum scoring class as the output label. We also tried hard voting to accumulate sentence level decisions, and observed that the hard voting results follow the same trend across datasets and splits.

On the smaller political speech dataset, the attribute classifier is able to easily discriminate between the two authors, Barack Obama and Donald Trump, achieving perfect F1-score of 1.0 on both the training and the validation splits. The model also performs well on the age-group classification, achieving F1-score of 0.88 on the validation set at the document-level. Gender classification turns out to be the hardest to generalize, with a significant drop in F1-score on the validation set compared to the training set (down to 0.75 from 0.93). However, we note that our gender classifier achieves similar document-level classification accuracies to the traditional stylometric methods (76.1% accuracy reported in [4], compared to 74.1% accuracy achieved by our model). In all three tasks, the performance on sentence-level is worse than on document-level classification. This is especially true in the gender classification task, where the sentence-level classification does only marginally better than the random chance (0.52 F1-score). This is expected since the stylistic differences between authors of different genders is not visible in strong in all the sentences. However there is usually a few sentences within a document where these stylistic differences are strong. Thus when we aggregate the classifier scores over a document, the classifier does much better and is well above the random chance (0.75 F1-score vs 0.50 chance-level). Document-level classification also generalizes better with less difference between training and validation set F1-scores in Table II. Henceforth, we will use document-level F1-score as our primary metric when evaluating the effectiveness of A^4NT networks.

6.1.1 Quantifying Privacy Gains from A^4NT

We evaluate the performance of our A^4NT network using the attribute classifiers presented in the previous section. To obfuscate the authorship of a document, the A^4NT network is run on every sentence in a document to transfer style from the true attribute to the target one. For example, to mask age of a teenager written document we use the A^4NT network $Z_{Teen-Adult}$. Style-transferred document is input to the attribute classifier of the source attribute and F1-score of this classifier is computed. This measures the privacy effectiveness of the style transfer. Meteor score is computed between the source sentence and the A^4NT output, to measure the semantic similarity.

Table III shows these results in the three settings. On the small speech dataset all methods, including the autoencoder baseline described in Section 5.3, successfully fool the attribute classifier. They all obtain F1-scores below the chance-level, with our A^4NT networks doing better. However the meteor scores of all models is significantly lower than in the blog dataset, indicating significant amount of semantic loss in the process of anonymization.

On the larger blog dataset, the autoencoder baseline fails to fool the attribute classifier, with only a small drop in F1-score of 0.03 (from 0.88 to 0.85) in case of age and 0.14 in case of gender (from 0.75 to 0.61) Our A^4NT models however do much better, with all of them being able to drop the F1-score below the random chance.

The *FBsem* model using semantic encoder loss achieves the largest privacy gain, by decreasing the F1-scores from 0.88 to 0.08 in case of age and from 0.75 to 0.39 in case of gender. This model however suffers from poor meteor scores, indicating the sentences produced after the style transfer are no longer similar to the input.

The model using reconstruction likelihood to enforce semantic consistency, *CycML*, fares much better in meteor metric in both age and gender style transfer. It is still able to fool the classifier, albeit with smaller drops in F1-scores (still below random chance). Finally, with addition of the language smoothing loss (*CycML+Lang*), we see a further improvement in the meteor score in the blog-age setting, while the performance remains similar to *CycML* on blog-gender setting and the speech dataset. However, the language smoothing model *CycML+Lang* fares better in the user study discussed in Section 6.1.2 and also produces better qualitative samples as will be seen in Section 6.2.

Generalization to other classifiers: An important question to answer if A^4NT is to be applied to protect the privacy of author attributes, is how well it performs against unseen NLP based adversaries ? To test this we trained ten different attribute classifiers networks on the blog-age setting. These networks vary in architectures (LSTM, CNN and LSTM+CNN) and hyper-parameters (number of layers and number of units), but all of them achieve good performance in predicting the age attribute. The networks were chosen to reflect real-world architecture choices used for text classification. Results from evaluating the text generated by the A^4NT networks using these "holdout" classifiers are shown in Table IV. The column "mean" shows the mean performance of the ten classifiers and "max" shows the score of best performing classifier

Model	Blog-age data			Blog-gender data			Speech dataset		
	Sent F1	Doc F1	Meteor	Sent F1	Doc F1	Meteor	Sent F1	Doc F1	Meteor
Random classifier	0.54	0.54	-	0.53	0.5	-	0.60	0.61	-
Original text	0.74	0.88	1.0	0.52	0.75	1.0	0.68	1.00	1.0
Autoencoder	0.69	0.85	0.77	0.49	0.61	0.78	*0.26*	*0.12*	0.32
A^4NT FBsem	*0.43*	*0.08*	0.40	*0.45*	*0.39*	0.53	*0.12*	*0.00*	0.28
A^4NT CycML	*0.49*	*0.20*	0.57	*0.41*	*0.44*	**0.79**	*0.11*	*0.00*	**0.29**
A^4NT CycML+Lang	*0.54*	*0.32*	**0.69**	*0.44*	*0.39*	**0.79**	*0.12*	*0.00*	**0.29**

Table III: Performance of the style transfer anonymization in fooling the classifiers, across the three settings. F1 (lower is better) and Meteor (higher is better). F1-scores below chance levels are shown in italics.

Holdout classifiers have good performance on the original text, achieving mean 0.85 document-level F1-score. Table IV shows that all three A^4NT networks generalize well and are able to drop the document F1-score of the holdout classifiers to the random chance level (0.54 for the blog-age setting). They perform slightly worse than on the seen LSTM classifier, but are able to significantly drop the performance of all the holdout classifiers (mean F1 score drops from 0.85 to 0.53 or below). This is a strong empirical evidence that the transformations applied by the A^4NT networks are not specific to the classifier they are trained with, but can also generalize to other adversaries.

We conclude that the proposed A^4NT networks are able to fool the attribute classifiers on all three tested tasks and also show generalization ability to fool classifier architectures not seen during training.

Note that here we have considered dropping the classifier score below the random chance-level on average as the criteria for fooling the classifiers and protecting privacy as motivated in Section 5.2. However, in some scenarios it is desirable for the obfuscation model to fool the classifiers all the time and pushing adversary to achieving 0% classification accuracy. This guarantee would provide the user the freedom to use the obfuscation only when desired. Our A^4NT model is not able to reach this level of 100% obfuscation. It remains an important open problem for the future work to achieve 100% obfuscation while preserving semantics, so that the user remains in full control.

Comparison to prior work: We also compare the performance of our model to a prior work on automatic anonymization [26]. This work proposes to anonymize the writing style by computing average statistics on certain text features and applying pre-defined transformations to change the input text statistics towards the average. We refer to this model as *C-Avg* and test this model on our age obfuscation task using the official code from [26]. The results are shown in Table V. For fair comparison we compare the performance against a holdout classifier, not seen by our model as well. We can see that the *C-Avg* model [26] does not perform well on the age obfuscation task, managing to drop the F1-score only to 0.77 from 0.84, which is well above the random chance-level. Our A^4NT model however drops the F1-score below the chance-level to 0.44. Our model does better in semantic similarity as well, achieving meteor score of 0.69 compared to 0.55 obtained by *C-Avg*. The poor performance of *C-Avg* model [26] on the age obfuscation task is due to the fact that *C-Avg* relies on hand designed transformations (eg. substituting synonyms from a dictionary) which does not generalize well to the diverse writing styles found in the blog dataset. This highlights the advantage of the proposed approach to learn to perform obfuscation directly from the data.

Different operating points : Our A^4NT model offers the ability to obtain multiple different style-transfer outputs by simply sampling from the models distribution. This is useful as different text samples might have different levels of semantic similarity and privacy effectiveness. Having multiple samples allows users to choose the level of semantic similarity vs privacy trade-off they prefer.

We illustrate this in Figure 6. Here five samples are obtained from each A^4NT model for each sentence in the test set. By choosing the sentence with minimum, maximum or random meteor scores w.r.t the input text, we can obtain a trade-off between semantic similarity and privacy. We see that while the *FBsem* model offers limited variability, *CycML+LangLoss* offers a wide range of choices of operating points. All operating points of *CycML+LangLoss* achieve better meteor score than 0.5, which indicates this model preserves the semantic similarity well.

6.1.2 Human Judgments for Semantic Consistency

In machine translation and image captioning literature, it is well known that automatic semantic similarity evaluation metrics like meteor are only reliable to a certain extent. Evaluation from human judges is still the gold-standard with which models can be reliably compared.

Accordingly, we conduct user studies to judge the se-

Model	Seen Classifier F1-score	Holdout Classifiers	
		Mean F1	Max F1
Original text	0.88	0.85	0.87
Autoencoder	0.85	0.83	0.84
A^{4}NT FBsem	*0.08*	*0.19*	*0.31*
A^{4}NT CycML	*0.20*	*0.41*	0.58
A^{4}NT CycML+Lang	*0.32*	*0.53*	0.62

Table IV: Evaluating the A^{4}NT anonymization against previously unseen (holdout) classifiers, on blogdata (age). Document-level F1 score is used.

Model	Holdout Classifier Doc F1-score	Meteor
Original text	0.84	1.0
C-Avg [26]	0.77	0.55
Ours	***0.44***	**0.69**

Table V: Comparison of our A^{4}NT model to prior work on automatic anonymization. We compare both privacy effectiveness against a classifier and semantic consistency (meteor metric).

mantic similarity preserved by our A^{4}NT networks. The evaluations were conducted on a subset of 745 random sentences from the test split of the blog-age dataset. First, output from different A^{4}NT models is obtained for the 745 test sentences. If any model generates identical sentences to the input, this model is ranked first automatically without human evaluation. Note that, in some cases, multiple models can achieve rank-1, when they all produce identical outputs. The cases without any identical sentences to the input are evaluated using human annotators on Amazon Mechanical Turk (AMT). An annotator is shown one input sentence and multiple style-transfer outputs and is asked to pick the output sentence which is closest in meaning to the input sentence. Three unique annotators are shown each test sample and majority voting is used to determine the model which ranks first. Cases with no majority from human evaluators are excluded.

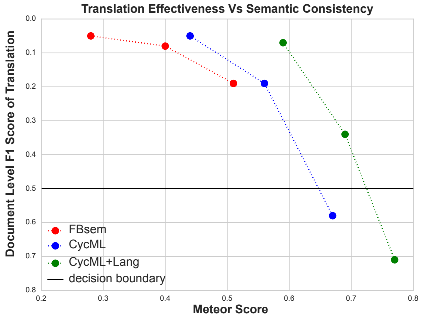

Figure 6: Operating points of A^{4}NT models on test set.

The main goal of the study is to identify which of the three A^{4}NT networks performs best in terms of semantic similarity according to human judges. We also compare the best of our three systems to the baseline model based on Google machine translation, discussed in Section 5.3.

For the machine translation baseline, we obtain style-transferred texts from four different language round-trips. We started with English→German→French→English, and obtained three more versions with incrementally adding Spanish, Finnish and finally Armenian languages into the chain before the translation back to English.

To pick the operating points for the user study, we compare the performance of these four machine translation baselines and our three models on the human-evaluation test set in Figure 7. Note that here we show sentence-level F1 score on the y-axis as the human-evaluation test set is too small for document-level evaluation. We see that none of the Google machine translation baselines are able to fool the attribute classifiers. The model with 5-hop translation achieves best (lowest) F1-score of 0.81 which is only slightly less than the input data F1-score of 0.9. This model also achieves significantly worse meteor score than any of our A^{4}NT models.

We conduct the user study comparing our style-transfer models on two operating points of 0.5 F1-score and 0.66 F1-scores, to obtain human judgments at two different levels of privacy effectiveness as shown in Table VI. We see that the model *CycML+Lang* outperforms the other two models at both operating points. *CycML+Lang* wins 50.74% of the time (ignoring ties) at operating point 0.5 and 57.87% of the time at operating point 0.66. These results combined with quantitative evaluation discussed in Section 6.1 confirm that the cyclic ML loss combined with the language model loss gives the best trade-off between semantic similarity and privacy effectiveness.

Finally, we conduct the user study between the *CycML+Lang* model operating at 0.79 and the Google machine translation baseline with 3 hops. The operating point is chosen so that the two models are closest to each other in privacy effectiveness and meteor score. Results in Table VII show that our model wins over the GoogleMT baseline by approximately 16% (59.46% vs 43.76% rank1) on semantic similarity as per human judges, while still having better privacy effectiveness. This is largely because our A^{4}NT model learns not to change the input text if it is already ambiguous for the attribute classifier, and only makes changes when necessary. In contrast, changes made by GoogleMT round trip are not optimized towards maximizing privacy gain, and can change the input text even when no change is needed.

Apart from the relative evaluation between our model

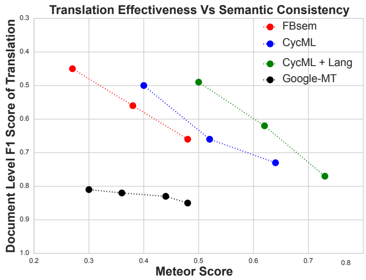

Figure 7: Privacy and semantic consistency of A^{4}NT and the Google MT baseline on the human evaluation test set

Operating Point	FBsem	CycML	CycML + Lang
0.66	32.02	39.75	**57.87**
0.5	15.03	31.68	**50.74**

Table VI: User study to judge semantic similarity. Three variants of our model are compared. Numbers show the % times the model ranked first. Can add to more than 100% as multiple models can have rank-1.

Comparison	A^{4}NT CycML + Lang	GoogleMT
Operating point	0.79	0.85
Relative (% Rank 1)	**59.46**	43.76
Absolute (0-5)	**4.51±0.84**	4.16±0.89

Table VII: User study of our best model and the Google MT baseline.

and the GoogleMT baseline, we additionally conduct separate a user study for both the models to assess the semantic similarity to the input sentence in an absolute scale. This study is conducted on the same human-evaluation test set containing 745 sentences and using the AMT platform as before. We show each human judge the input sentence and output form either of the models and ask them to rate the similarity to the input in a Likert scale from zero to five. We adopt the instruction used in SemEval task [57] to describe the different rating values to the user. Here zero rating corresponds to the worst case where the input and output sentences are not semantically related and five corresponds to the best case where they are equivalent in meaning. Full definition of scales and further details about the user study is presented in the appendix B. Each input-output pair is evaluated by three human judges and we report the mean score and standard deviation in Table VII. We see the same trend as in the relative evaluation and our model achieves better overall score of 4.51/5.0 compared to 4.16 obtained by the GoogleMT baseline. The score of the A^{4}NT model lies between the ratings of 4.0 (sentences are equivalent with unimportant details differing) and 5.0 (sentences are equivalent). This shows that the A^{4}NT model preserves the meaning of the input sentence on average, by making semantically equivalent changes to fool the authorship classifier.

6.2 Qualitative Analysis

In this section we analyze some qualitative examples of anonymized text produced by our A^{4}NT model and try to identify the strengths and the weaknesses of this approach. Then we analyze the performance of the A^{4}NT network on different levels of input difficulty. We use the attribute classifiers' score as a proxy measure of the input text difficulty. If the text is confidently correctly classified (with classification score of 1.0) by the attribute classifier, then the A^{4}NT network has to make significant changes to fool the classifier. If it is already misclassified, the style-transfer network should ideally not make any changes.

6.2.1 Examples of Style Transfer for anonymization

Table VIII shows the results of our A^{4}NT model *CycML+Lang* applied to some example sentences in the blog-age setting. Style transfer in both directions, teenager to adult and adult to teenager, is shown along with the corresponding source attribute classifier scores. The examples illustrate some of the common changes made by the model and are grouped into three categories for analysis (# column in Table VIII).

# 1. Using synonyms: The A^{4}NT network often uses synonyms to change the style to target attribute. This is seen in style transfers in both directions, teen to adult and adult to teen in category # 1 samples in Table VIII. We can see the model replacing "yeh" with "ooh", "would" with "will", "..." with "," and so on when going from teen to adult, and replacing "funnily enough" with "haha besides", "work out" with "go out" and so on when changing from adult to teen. We can also see that the changes are not static, but depend on the context. For example "yeh" is replaced with "alas" in one instance and with "ooh" in another. These changes do not alter the meaning of the sentence too much, but fool the attribute classifiers thereby providing privacy to the author attribute.

# 2. Replacing slang words: When changing from teen to adult, A^{4}NT often replaces the slang words or incorrectly spelled words with standard English words, as seen in category #2 in Table VIII. For example, replacing "wad" (what) with "definitely", "wadeva" with "perhaps" and "nuthing" with "ofcourse". The opposite effect is seen when going from adult to teenager, with addition of "diz" (this) and replacing of "think" with "relized" (realized). These changes are learned entirely from the data, and would be very hard to encode explicitly in a rule-based system due to the variety in slangs and spelling mistakes.

# 3. Semantic changes: One failure mode of A^{4}NT is when the input sentence has semantic content which is significantly more biased to the author's class. These examples are shown in category #3 in Table VIII. For example, when an adult author mentions his "wife", the

#	Input: Teen	A(x)	Output: Adult	A(x)
1	and yeh... it's raining lots now	0.97	and ooh... it's raining lots now	0.23
1	yeahh... i never let anyone really know how i'm feeling.	0.94	anyhow, i never let anyone really know how i'm feeling .	0.24
1	yeh, it's just goin ok here too!	0.95	alas, it's just goin ok here too!	0.30
1	would i go so far to say that i love her?	0.52	will i go so far to say that i love her?	0.36
2	wad a nice day.. spend almost the whole afternoon doing work!	0.99	definitely a nice day.. spend almost the whole afternoon doing work!	0.19
2	wadeva told u secrets wad did u do ?	0.98	perhaps told u secrets why did u do ?	0.49
2	i don't know y i even went into dis relationship	0.92	i don't know why i even went into another relationship .	0.33
2	i have nuthing else to say about this horrid day.	0.79	i have ofcourse else to say about this accountable day.	0.08
3	after school i got my hair cut so it looks nice again.	1.0	after all i have my hair cut so it looks nice again.	0.42
3	i had an interesting day at skool.	0.97	i had an interesting day at wedding.	0.05
#	**Input: Adult**	**A(x)**	**Output: Teen**	**A(x)**
1	funnily enough , i do n't care all that much.	0.58	haha besides , i do n't care all that much.	0.05
1	i may go to san francisco state, or i may go back.	0.54	i shall go to san francisco state, or i may go back.	0.09
1	i wonder if they 'll work out... hard to say.	0.52	i wonder if they 'll go out... hard to say.	0.39
2	one is to mix my exercise order a bit more.	0.97	one is to mix my diz exercise order a bit more.	0.08
2	ok, think i really will go to bed now.	0.79	ok, relized i really will go to bed now.	0.08
3	my first day going out to see clients after vacation.	0.98	my first day going out to see some1 after vacation.	0.04
3	i'd tell my wife how much i love her every time i saw her.	0.96	i'd tell my crush how much i love her every time i saw her.	0.06
3	i do believe all you need is love.	0.58	i dont think all you need is love .	0.11

Table VIII: Qualitative examples of anonymization through style transfer in the blog-age setting. Style transfer in both direction is shown along with the attribute classifier score of the source attribute.

Input: Obama	Output: Trump
we can do this because we are MISC.	we will do that because we are MISC.
we can do better than that.	we will do that better than anybody.
it's not about reverend PERSON.	it's not about crooked PERSON.
but i'm going to need your help.	but i'm going to fight for your country.
so that's my vision.	so that's my opinion.
their situation is getting worse.	their media is getting worse.
i'm kind of the term PERSON because i do care.	i'm tired of the system of PERSON PERSON because they don't care.
that's what we need to change.	that's what she wanted to change.
that's how our democracy works.	that's how our horrible horrible trade deals.

Table IX: Qualitative examples of style transfer on the speech dataset from Obama to Trump's style

A^{4}NT network replaces it with "crush", altering the meaning of the input sentence. Some common entity pairs where this behavior is seen are with (*school*↔*work*), (*class*↔*office*), (*dad*↔*husband*), (*mum*↔*wife*), and so on. Arguably, in such cases, there is no obvious solution to mask the identity of the author without altering these obviously biased content words.

On the smaller speech dataset however, the changes made by the A^{4}NT model alter the semantics of the sentences in some cases. Few example style transfers from Obama to Trump's style are shown in Table IX. We see that A^{4}NT inserts hyperbole ("better than anybody", "horrible horrible", "crooked"), references to "media" and "system", all salient features of Trump's style. We see that the style-transfer here is quite successful, sufficient to completely fool the identity classifier as was seen in Table III. However, and somewhat expectedly, the semantics of the input sentence is generally lost. A possible cause is that the attribute classifier is too strong on this data, owing to the small dataset size and the highly distinctive styles of the two authors, and to fool them the A^{4}NT network learns to make drastic changes to the input text.

6.2.2 Performance Across Input Difficulty

Figure 8 compares the attribute classifier score on the input sentence and the A^{4}NT output. Ideally we want all the A^{4}NT outputs to score below the decision boundary, while also not increasing the classifier score compared to input text. This "ideal score" is shown as grey solid line. We see that for the most part all three A^{4}NT models are below or close to this ideal line. As the input text gets more difficult (increasing attribute classifier score), the *CycML* and *CycML+Lang* slightly cross above the ideal line, but still provide significant improvement over the input text (drop in classifier score of about ~ 0.45).

Now, we analyze how much of input semantics is preserved with increasing difficulty. Figure 9 plots the meteor

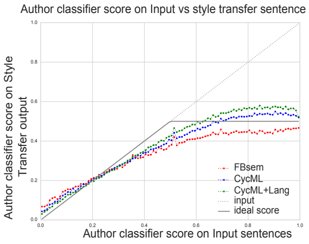

Figure 8: Output Privacy vs Privacy on Input.

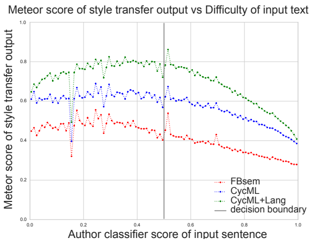

Figure 9: Meteor score plotted against input difficulty.

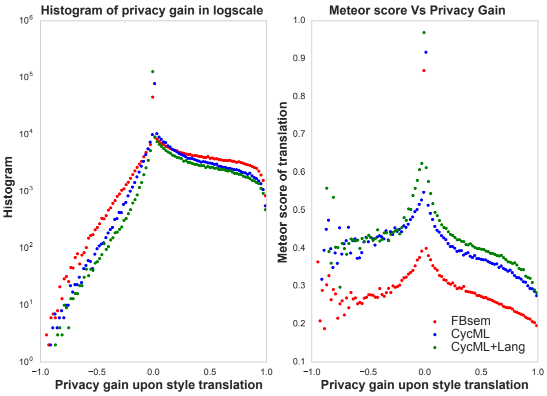

Figure 10: Histogram of privacy gain (left side) is shown alongside comparison of meteor score vs privacy gains.

score of the A^{4}NT output against the difficulty of the input text. We see that the meteor is high for sentences already across the decision boundary. These are easy cases, where the A^{4}NT networks need not intervene. As the input gets more difficult, the meteor score of the A^{4}NT output drops, as the network needs to do more changes to be able to fool the attribute classifier. The *CycML+Lang* model fares better than the other two models, with consistently higher meteor across the difficulty spectrum.

Figure 10 shows the histogram of privacy gain across the test set. Privacy gain is the difference between the attribute classifier score on the input and the A^{4}NT network output. We see that majority of transformations by the A^{4}NT networks leads to positive privacy gains, with only a small fraction leading to negative privacy gains. This is promising given that this histogram is over all the 500k sentences in the test set. Meteor score plotted against privacy gain shown in Figure 10, again confirms that large privacy gains comes with a trade-off of loss in semantics.

7 Conclusions

We presented a novel fully automatic method for protecting privacy sensitive attributes of an author against NLP based attackers. Our solution consists of the A^{4}NT network which learns to protect private attributes with novel adversarial training of a machine translation model. The A^{4}NT network achieves this by learning to perform style-transfer without paired data.

A^{4}NT offers a new data driven approach to authorship obfuscation. The flexibility of this end-to-end trainable model means it can adapt to new attack methods and datasets. Experiments on three different attributes namely age, gender and identity, showed that the A^{4}NT network is able to effectively fool the attribute classifiers in all the three settings. We also show that the A^{4}NT network also performs well against multiple unseen classifier architectures. This strong empirical evidence suggests that the method is likely to be effective against previously unknown NLP adversaries.

We developed a novel solution to preserve the meaning of input text using likelihood of reconstruction. Semantic similarity (quantified by meteor score) of the A^{4}NT network remains high for easier sentences, which do not contain obvious give-away words (school, work, husband etc.), but is lower on difficult sentences indicating the network effectively learns to identify and apply the right magnitude of change. The A^{4}NT network can be operated at different points on the privacy-effectiveness and semantic-similarity trade-off curve, and thus offers flexibility to the user. The experiments on the political speech data show the limits to which style transfer based approach can be used to hide attributes. On this challenging data with very distinct styles by the two authors, our method effectively fools the identity classifier but achieves this by altering the semantics of the input text.

Acknowledgment

This research was supported in part by the German Research Foundation (DFG CRC 1223). We would also like to thank Yang Zhang, Ben Stock and Sven Bugiel for helpful feedback.

References

[1] P. Juola *et al.*, “Authorship attribution,” *Foundations and Trends® in Information Retrieval*, 2008.

[2] E. Stamatatos, “A survey of modern authorship attribution methods,” *Journal of the Association for Information Science and Technology*, 2009.

[3] S. Ruder, P. Ghaffari, and J. G. Breslin, “Character-level and multichannel convolutional neural networks for large-scale authorship attribution,” *arXiv preprint arXiv:1609.06686*, 2016.

[4] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler, “Automatically profiling the author of an anonymous text,” *Communications of the ACM*, 2009.

[5] R. Overdorf and R. Greenstadt, “Blogs, twitter feeds, and reddit comments: Cross-domain authorship attribution,” *Proceedings on Privacy Enhancing Technologies*, 2016.

[6] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, “On the feasibility of internet-scale author identification,” in *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012.

[7] P. Juola. (2013) How a computer program helped show J.K. rowling write a cuckoos calling. [Online]. Available: https://goo.gl/mkZai1

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

[9] K. Ikeda, G. Hattori, C. Ono, H. Asoh, and T. Higashino, "Twitter user profiling based on text and community mining for market analysis," *Know.-Based Syst.*, 2013.

[10] A. Makazhanov, D. Rafiei, and M. Waqar, "Predicting political preference of twitter users," *Social Network Analysis and Mining*, 2014.

[11] H. Grassegger and M. Krogerus. (2017) The data that turned the world upside down. [Online]. Available: https://motherboard.vice.com/en_us/article/mg9vvn/how-our-likes-helped-trump-win

[12] M. Brennan, S. Afroz, and R. Greenstadt, "Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity," *ACM Transactions on Information and System Security (TISSEC)*, 2012.

[13] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012.

[14] A. W. McDonald, S. Afroz, A. Caliskan, A. Stolerman, and R. Greenstadt, "Use fewer instances of the letter" i": Toward writing style anonymization." in *Privacy Enhancing Technologies*. Springer, 2012.

[15] D. Castro, R. Ortega, and R. Muñoz, "Author Masking by Sentence Transformation—Notebook for PAN at CLEF 2017," in *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*, Sep. 2017.

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

[17] A. Caliskan and R. Greenstadt, "Translate once, translate twice, translate thrice and attribute: Identifying authors and machine translation tools in translated text," in *2012 IEEE Sixth International Conference on Semantic Computing*, Sept 2012.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014.

[20] W. Xu, A. Ritter, B. Dolan, R. Grishman, and C. Cherry, "Paraphrasing for style," *Proceedings of COLING 2012*, 2012.

[21] S. Afroz, A. C. Islam, A. Stolerman, R. Greenstadt, and D. McCoy, "Doppelgänger finder: Taking stylometry to the underground," in *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 2014.

[22] A. Caliskan-Islam, R. Harang, A. Liu, A. Narayanan, C. Voss, F. Yamaguchi, and R. Greenstadt, "De-anonymizing programmers via code stylometry," in *USENIX Security Symposium*, 2015.

[23] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Transactions on Information Systems (TOIS)*, 2008.

[24] D. Bagnall, "Author identification using multi-headed recurrent neural networks," *arXiv preprint arXiv:1506.04891*, 2015.

[25] G. Kacmarcik and M. Gamon, "Obfuscating document stylometry to preserve author anonymity," in *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, 2006.

[26] G. Karadzhov, T. Mihaylova, Y. Kiprov, G. Georgiev, I. Koychev, and P. Nakov, "The case for being average: A mediocrity approach to style masking and author obfuscation," in *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 2017.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[28] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[29] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, "Style transfer from non-parallel text by cross-alignment," *To appear in NIPS*, 2017.

[30] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[31] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[32] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017.

[33] S. J. Oh, M. Fritz, and B. Schiele, "Adversarial image perturbation for privacy protection–a game theory perspective," in *International Conference on Computer Vision (ICCV)*, 2017.

[34] S. Samanta and S. Mehta, "Towards crafting text adversarial samples," *arXiv preprint arXiv:1707.02812*, 2017.

[35] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep text classification can be fooled," *arXiv preprint arXiv:1704.08006*, 2017.

[36] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[37] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker, "Effects of age and gender on blogging." in *AAAI spring symposium: Computational approaches to analyzing weblogs*, 2006.

[38] A. A. Morgan-Lopez, A. E. Kim, R. F. Chew, and P. Ruddle, "Predicting age groups of twitter users based on language and metadata features," *PLOS ONE*, 08 2017.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[41] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[42] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly transcribe foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[43] X. Ma and E. Hovy, "End-to-end sequence labeling via bidirectional LSTM-CNNs-CRF," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[44] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, and B. Schiele, "Speaking the same language: Matching machine to human captions by adversarial training," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[45] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[46] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[47] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[48] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015.

[49] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[50] Pytorch framework. [Online]. Available: http://pytorch.org/

[51] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, 2012.

[52] J. T. Woolley and G. Peters. (1999) The american presidency project. [Online]. Available: http://www.presidency.ucsb.edu

[53] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.

[54] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. ACL, 2014.

[55] Z. Li, X. Jiang, L. Shang, and H. Li, "Paraphrase generation with deep reinforcement learning," *arXiv preprint arXiv:1711.00279*, 2017.

[56] D. Elliott, S. Frank, L. Barrault, F. Bougares, and L. Specia, "Findings of the second shared task on multimodal machine translation and multilingual image description," in *Proceedings of the Second Conference on Machine Translation*, 2017.

[57] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe, "Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016.

[58] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

A Differentiability of discrete samples

We obtain an output sentence sample $\tilde{s_y}$ from the A^{4}NT network Z_{xy} by sampling from the distribution $p(\tilde{w}_t|s_x)$, shown in (5), repeatedly until a special 'END' token is sampled. This naive sampling though is not suitable for training Z_{xy} within a GAN framework as sampling from multinomial distribution is not differentiable.

To make sampling differentiable we follow the approach used in [44] and use the Gumbel-Softmax approximation [45] to obtain differentiable soft samples from $p(\tilde{w}_t|s_x)$. The gumbel-softmax approximation includes two parts. First, the re-parametrization trick using the gumbel random variable is applied to make the process of sampling from a multinomial distribution differentiable with respect to the probabilities $p(\tilde{w}_t|s_x)$. Next, softmax is used to approximate the arg-max operator to obtain "soft" samples instead of one-hot vectors. This makes the samples themselves differentiable. Thus, the gumbel-softmax approximation allows differentiating through sentence samples from the A^{4}NT network enabling end-to-end GAN training. Further details on gumbel-softmax approximation can be found in [45, 58].

B Human evaluation

Rating	Instruction
5	The two sentences are completely equivalent, as they mean the same thing.
4	The two sentences are mostly equivalent, but some unimportant details differ.
3	The two sentences are roughly equivalent, but some important information differs/missing.
2	The two sentences are not equivalent, but share some details.
1	The two sentences are not equivalent, but are on the same topic
0	The two sentences are completely dissimilar

Table X: The zero to five scale and corresponding instructions used to conduct the user study of absolute semantic similarity between the input and the output sentence.

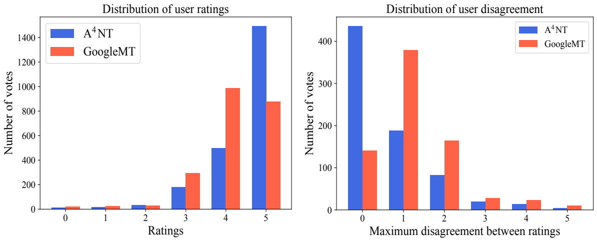

Figure 11: Comparing distribution of ratings obtained by our model and the GoogleMT baseline in the absolute semantic similarity user study. Left figures shows the distribution of the ratings, whereas the figure on the right shows the distribution of maximum difference between user ratings for each sentence.

Both the user studies presented in Section 6.1.2 were conducted on Amazon Mechanical Turk platform (AMT). The workers were based in the united states and were required to have Mechanical Turk masters qualification, which is given by the AMT platform to workers producing high quality work. The workers were also required to have a minimum approval rating of 95% in their prior assignments on AMT. All the workers who participated in the two user studies were compensated through the AMT platform. The workers were paid an average of 0.02$ for each sentence evaluation task, which took a median of twelve seconds complete. Both the studies were conducted on the human-eval test set containing 745 test sentences and each sentence was evaluated by three unique users. We did not collect any personal information from the users. A total of 18 unique users participated in the user study measuring absolute semantic similarity, with each user rating on an average 176.25 sentences. In the relative semantic similarity evaluations, a total of 70 unique users participated with each user evaluating on average 55.6 pairs of sentences.

Relative evaluation: In the first evaluation we show each user the input sentence and the modified sentences from different models and ask the users to pick the sentence which best preserves the meaning of the input text. This task was titled "Pick semantically similar sentence from a list" on AMT and was description provided was "Pick from the given list, the sentence closest in meaning to the provided reference sentence". Each time a model's output sentence is picked by a user, we consider it as ranked first. For sentences were one or more of the models produce output sentence identical to the input, we directly award those models rank one for these sentences. Finally, we compare the models based on the percentage of instances they were ranked first as presented in Section 6.1.2. We found good agreement between the users on this task. All the three users rating each sentence agreed 62% of the time in this task, compared 25% chance of agreement if the three users were randomly voting.

Absolute evaluation: We also evaluated the semantic similarity of the edited text to the input on an absolute scale of zero to five. Each user is shown the input sentence and the edited sentence and is asked to rate the semantic similarity on zero (no similarity) to five (identical) scale. This task was titled "Rate the similarity of two sentences on a scale" on AMT and was description provided was "You are presented with two sentences. Rate how similar they are in meaning on a scale of 0 to 5" along with the rating guide in Table X. Again, if a model produces identical output sentence to the input, we award a rating of five automatically. The models are compared using the average rating they obtain as presented in Section 6.1.2. To evaluate the agreement between the three user ratings for each sentence, we plot the distribution of ratings and distribution of maximum difference between the three ratings in Figure 11. We can see that the most of the ratings are distributed between four and five. Also the users tend to rate the sentences similarly, with the maximum difference between user ratings mostly distributed between zero and one. We see that users tend to agree more on our A^{4}NT model compared to the GoogleMT baseline. This is due to the fact that our model preserves many more sentences identical compared to the GoogleMT baseline.