# 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 allow to reveal privacy relevant author attributes such as gender, age and identify of 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 Adversarial Author Attribute Anonymity Neural Translation (A$^4$NT), to combat such text-based adversaries. We combine sequence-to-sequence language models used in machine translation and generative adversarial networks to obfuscate author attributes. Unlike machine translation techniques which need paired data, our method can be trained on unpaired corpora of text containing different authors. Importantly, we propose and evaluate techniques to impose constraints on our A$^4$NT to preserve the semantics of the input text. A$^4$NT learns to make minimal changes to the input text to successfully fool author attribute classifiers, while aiming to maintain the meaning of the input. We show through experiments on two different datasets and three settings that our proposed method is effective in fooling the author attribute classifiers and thereby improving the anonymity of authors.**

## 1. Introduction

Anonymity and privacy have been at the forefront of our discourse about the internet since the early days [1]. Ability to communicate anonymously helps people exercise their freedom of speech without fear of prosecution as well as organize social movements [2]. Anonymity could be vital when voicing a dissenting political opinion, or attempting to avoid harassment and discrimination. The latter two applications require stronger anonymity in the sense that it is not just the leak of the author's identity which compromises the author but even the unmasking of attributes like age, race or gender. Exposing these private attributes can expose people to harassment and discrimination [3, 4]. While protecting anonymity is not the solution to the root of discrimination and harassment, it can help avoid or reduce their occurrence.

Natural language processing (NLP) methods including stylometric tools allow to identify the authors of anonymous texts [5]–[7]. These methods can be used to break the anonymity of the author even when sufficient care has been taken to hide the identity in the content of the text. As discussed in [8] stylometric methods can successfully identify the authors who were protected with pseudonymity, with just 6,500 words written by the author. Since these methods rely solely on the content of the text and not the medium of communication, security measures like masking the IP addresses or posting patterns can still be vulnerable to attacks with NLP methods. NLP-tools have also been applied to determine authors' private attributes like age and gender [9]. These attribute classifiers perform better than classifiers to predict exact identity, since they deal with a smaller number of classes. Using such attribute classifiers, an adversary can easily obtain two out of three attributes (age and gender) listed in [10] required for exact identification. Combined with the fact that these methods have been shown to work on large scale datasets [11], they pose a serious privacy risk.

Prior works to obfuscate authorship and defend against NLP-methods has been largely restricted to semi-automatic solutions suggesting possible changes to the user [12] or hand-crafted transformations to text [13] which need re-engineering on different datasets [13]. This however limits the applicability of these defensive measures beyond the specific dataset it was designed on. To the best of our knowledge, generic text rephrasing using machine translation tools [14] is the only prior work offering a fully automatic solution to author obfuscation which can be applied across datasets. But as we show in our experiments, machine translation based obfuscation fails to sufficiently hide the identity and protect against attribute classifiers.

**Our work.** We propose an automatic model (A$^4$NT) to defend against NLP-methods. Our defense is inspired by the imitation model discussed in [15] and protects against various attribute classifiers by learning to imitate the writing style of a target class. For example, our model 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 learnt by adversarially training [16] 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 mis-classifying 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 [12, 13], 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 any author attribute ranging from the author's identify over gender to 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.

**Technical challenges:** We base our $A^4$NT network architecture on the sequence-to-sequence neural machine translation model [17]. A key challenge in learning to perform style transfer, compared to other sequence-to-sequence mapping tasks like machine translation, is the lack of parallel training data. Here, parallel/paired data refers to datasets with both the input text and its corresponding ground-truth output text. Some prior attempts to perform text style transfer required paired training data [18] 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) [16] framework. Using the GAN framework we 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 the proposed method 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".

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 transform the input text to a target style and fool the attribute classifiers. The network is trained without parallel data using 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 little to 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 will review prior work relating to four different aspects of our work – author attribute detection (our adversaries), authorship obfuscation (competing prior work), machine translation (basis of our $A^4$NT network) and generative adversarial networks (training framework we use).

**Authorship and attribute detection** Authorship attribution or stylometry is the task of identifying an author using the stylistic properties of the text. A machine learning approach where a set of text features are input to a classifier which learns to predict the author have been used in most recent author attribution models [6]. These methods have been shown to work well on large datasets [11], duplicate author detection [19] and even on non-textual data like code [20]. Sytlometric techniques can also be applied to determining various attributes of the author like age or gender [9].

Classical author attribution methods rely on a predefined set of features extracted from the input text [21]. Recently deep-learning methods have been applied to learn to extract the features directly from data [7, 22]. [22] uses a multi-headed character-level 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, [7] uses a character-level convolutional network to train an author classifier. Similar to these deep learning based approaches, we use a RNN based architecture to train our authorship classifiers.

**Authorship obfuscation** Authorship obfuscation methods are adversarial in nature to stylometric methods of author attribution; they try to change the style of input text such that author identity is not discernible. The majority of prior works on author attribution are semi-automatic [12, 23], where the system suggests authors to make changes to the document by analyzing the stylometric features. The few automatic obfuscation methods have relied on general rephrasing methods like machine translation [14] or on a predefined set of modifications to text [24]. 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 [14]. Recent work [24] achieves obfuscation by moving the stylometric features of the text towards average values on the dataset applying pre-defined transformations on input text.

In contrast, our novel method achieves 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 on various datasets, without additional engineering effort.

**Machine translation** The task of style-transfer of text data shares significant 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. Therefore, we base our $A^4$NT network architecture on popular current machine translation systems.

Large end-to-end trainable neural networks have become a popular choice in machine translation [25, 26]. These methods are generally based on sequence-to-sequence recurrent models [17] 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 model [17] as opposed to character-level models [27, 28], as the character-level models require much larger computational resources due to increased sequence length. Neural machine translation systems are generally trained with large amounts of parallel training data. However, in our setting, we do not have parallel training data of the same text in different writing styles. We overcome the lack of parallel training data by casting the problem as matching style distributions instead of matching individual sentences. Specifically, we want our A$^4$NT network to take an input text from a source distribution to generate text whose style matches the target attribute distribution. This is learnt without parallel data using distribution matching methods.

**Generative adversarial networks** Generative Adversarial Networks (GAN) [16] 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, the generator reaches optimality when it mimics the target distribution perfectly [16].

GANs have been successfully applied to many image generation tasks including unconditional generation [29], image inpainting [30], and style transfer [31]. [31] applies the GAN framework to train two paired generators to transfer style of an image between two domains without paired data. The generators are trained by using discriminators to push the generated image towards the target distribution. In addition to the GAN loss, a "cycle loss" is used to keep the generated sample semantically close to the input image by penalizing the $l_1$ distance between the input and a reconstruction obtained by round-trip style transfer.

We use a similar framework to transfer styles between the input and the target attributes. However, unlike [31], we use the likelihood of correct reconstruction instead of the $l_1$ loss to impose the cycle constraint. This proposed loss addresses the sensitivity of $l_1$ to alignment of the text sequences and makes the training robust.

**GAN on text** The GAN framework is not widely adopted in text generation tasks since the discrete nature of the text output does not allow for backpropagation directly. A few recent works have applied GAN to text generation tasks [32]–[35] by using approximations to overcome this discreetness problem. Gumbel-softmax approximation, which is used in [34], tends to have lower variance estimation of gradients compared to the REINFORCE trick [36] based solutions used in [32, 33, 35] and consequently better convergence properties as empirically shown in [37]. Thus, we use gumbel-softmax approximation in our generator and utilize the GAN framework to train our A$^4$NT networks.

A recent approach to text style-transfer proposed in [38] also utilizes GANs to perform style transfer using unpaired data. However, the solution proposed in [38] changes the meaning of the input text significantly during style transfer and is applied on sentiment transfer task. In contrast, authorship obfuscation requires the generated text to preserve the semantics of the input. We address this by proposing and evaluating two methods to encourage the network to preserve the meaning of the input text.

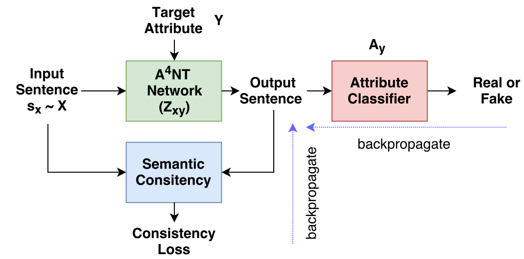


Figure 1: GAN framework to train our A$^4$NT network. Input sentence is transformed by the A$^4$NT network to match the style of the target attribute. This output is evaluated using the attribute classifier and semantic consistency loss. A$^4$NT is trained by backpropagating through these losses.

## 3. Author Attribute Anonymization

Private attributes of an anonymous author like identity, gender and age, can be de-anonymized by first extracting features from text using NLP-methods and then predict private attributes with some attribute classifier. We propose an author adversarial attribute anonymizing neural translation (A$^4$NT) network to defend against adversaries attacking these private attributes using NLP-methods. The proposed solution includes the A$^4$NT Network and the adversarial training scheme combined with semantic and language losses to learn to protect private attributes. The A$^4$NT network transforms the input text sequence from a source attribute class to mimic the style of a different attribute class, and thus fools the author attribute classifiers.

Technically, our A$^4$NT network is essentially solving a sequence to sequence mapping problem, similar to machine translation. Both tasks can be phrased as transforming an input text from the source domain (sequence of words) into a corresponding text sequence in a target domain. Hence, we design our A$^4$NT network based on the sequence-to-sequence neural language models [17], widely used in neural machine translation [25]. These models have proven effective when trained with large amounts of parallel data and are also deployed commercially [26]. If there were parallel aligned 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 parallel corpora are infeasible to obtain as it would require the same text written in multiple different styles.

To overcome the lack of parallel data, we view 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 $s_y \sim Y$. We can use generative modeling techniques to train the A$^4$NT network $Z_{xy}(s_x)$ to generate samples close to the target distribution $Y$, using just unpaired samples from $X$ and $Y$.

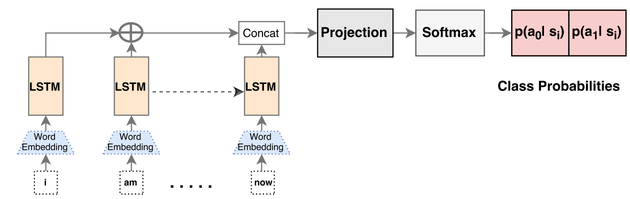


Figure 2: Block diagram of the attribute classifier network. The LSTM encoder embeds the input sentence into a vector. Sentence encoding is passed to linear projection followed by softmax layer to obtain class probabilities

We use the Generative Adversarial Network (GAN) framework to train our model (see Figure 1). The GAN framework consists of two models, a generator network which generates synthetic samples trying to mimic the target data distribution, and a discriminator network which tries to distinguish real data samples 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 generator samples. We use an attribute classifier network 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 3.4. In the following subsections we will describe each of these modules in detail.

### 3.1. 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 [7, 22] 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 two 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 [7, 22] for two reasons. We found that the word-level models give good performance on all three attribute-classification tasks we experiment with (see Section 5.1). Additionally, they are also significantly faster than character-level models, making it feasible to use these classifiers in GAN training described in Section 3.2.

Specifically, our attribute classifier network $A_x$ 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 $\{v_0, v_1, \cdots, v_{n-1}\}$. This word embedding layer encodes similarities between words into the word vectors and can help deal with large vocabulary sizes. The word vectors are randomly initialized and then learned from the data during 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 then passed through an LSTM network recursively 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 compute the LSTM output.

Now sentence embedding is obtained by concatenation of the final LSTM output and the mean of the LSTM outputs from other time-steps.

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

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. This is motivated by the success of such skip-connections in various applications of deep neural networks [42]–[44]. 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.

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

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

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

We experimented with more complex architectures – with more LSTM layers or a multi-layer perceptron instead of a linear layer – with little to no improvement over this simple model. The same network architecture can be used for different attribute classification tasks, including identity, age and gender as in this paper.

### 3.2. The A$^4$NT Network

One of the main goals while designing the A$^4$NT network is that it is trainable purely from data to obfuscate the

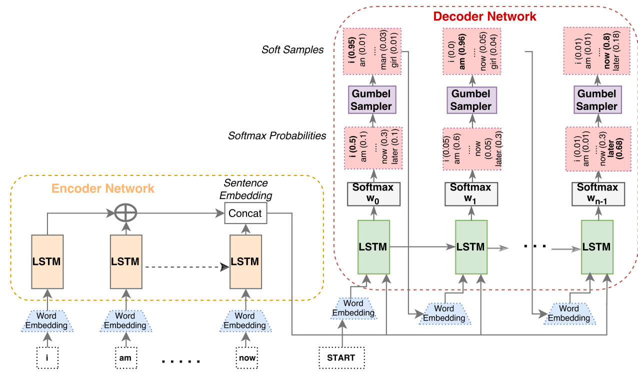


Figure 3: Block diagram of the A4NT network. First LSTM encoder embeds the input sentence into a vector. The decoder maps this sentence encoding to the output sequence. Gumbel sampler is used to obtain "soft" samples from the softmax distribution to allow backpropagation.

author attributes by learning to imitate the target style. This is a significant departure from prior works on author obfuscation [12, 24] 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.

To achieve this goal, we base our A4NT network $Z_{xy}$, shown in Figure 3, on a recurrent sequence-to-sequence neural translation model [17]. This class of models is popular in many sequence-to-sequence learning tasks such as machine translation [25], speech recognition [45], and part-of-speech (POS) tagging [46]. As seen from the wide-range of applications mapping text-to-text, speech-to-text, text-to-POS , the sequence-to-sequence neural network 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 [12, 24]. Instead, appropriate text transformations can be learnt directly from data. Additionally, since no hand-crated rules are needed, it allows us to easily apply the same A4NT network and training scheme to different datasets and settings.

The A4NT 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 sentence 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 3.1. The same architecture is used since the task here is also to embed the input sentence $s_x$ into a fixed size vector $E_G(s_x)$. However, in contrast to the attribute classifier, here the sentence embedding vector 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 the 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 for the next word. Sampling from this distribution outputs the next word.

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

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

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

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

In most applications of these sequence-to-sequence 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 A4NT network $Z_{xy}$ is trained in an unsupervised setting.

To address the lack of parallel training data, we use the GAN framework to train the A4NT network. In the GAN framework, the A4NT 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. Using the GAN framework also gives the benefit of directly optimizing our A4NT network 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 use the GAN framework, we need to be able to differentiate through the samples generated by $Z_{xy}$. The A4NT network produces stochastic output that is composed of a sequence of discrete tokens, and thus is not differentiable in a straight forward way.

**Differentiability of discrete samples:** To obtain an output sentence sample $\tilde{s_y}$ from the A4NT network $Z_{xy}$, we can sample 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 $Z$ within a GAN framework as sampling from multinomial distribution, $p(\tilde{w}_t|s_x)$, is not differentiable.

To make sampling differentiable we follow the approach used in [34] and use the Gumbel-Softmax approximation [37] 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 w.r.t 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 A4NT network enabling end-to-end GAN training. Further details on gumbel-softmax approximation can be found in [37, 47].

**Splitting decoder:** To transfer styles between attribute pairs, $x$ and $y$, in both directions, we found it ineffective to use

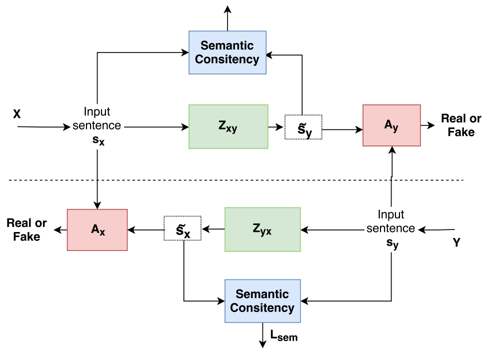


Figure 4: Illustrating use of GAN framework and cyclic semantic loss to train a pair of A4NT networks.

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 use the same encoder to embed the input sentence but different decoders for style transfer between each ordered pair of attributes. That is, A4NT network $Z_{xy}$ from $x$ to $y$ is composed as $Z_{xy} = [E_G; D_{xy}]$ and the network $Z_{yx}$ from $y$ to $x$ is composed as $Z_{yx} = [E_G; D_{yx}]$. Encoder $E_G$ is shared between $Z_{xy}$ and $Z_{yx}$. Sharing the encoder allows the two networks to share a significant number of parameters and enables the attribute specific decoders to deal with words found only in the vocabulary of the other attribute group using shared sentence and word embeddings.

### 3.3. Style Loss with GAN

To train our A4NT network to imitate the style of the target attribute class we use the Generative Adversarial Network framework [16]. Using the GAN framework allows us to train our two A4NT networks $Z_{xy}$ and $Z_{yx}$ 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 illustrates the GAN training mechanism we use to train the two A4NT networks. Given a sentence $s_x$ written by author with attribute $x$, the A4NT 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 sentence 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 A4NT network from $y$ to $x$, with $x$ and $y$ swapped.

The loss functions used to train the A4NT network and the attribute classifiers in this setting is given by:

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

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

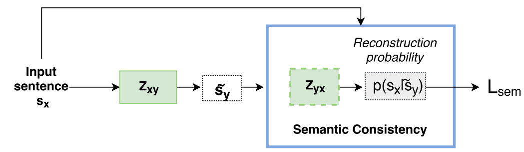


Figure 5: Semantic consistency in A4NT networks is enforced by maximizing cyclic reconstruction probability.

The two networks $Z_{xy}$ and $A_y$ are adversarially competing with each other when minimizing the above loss functions. At optimality it is guaranteed that the distribution of samples produced by $Z_{xy}$ is identical to the distribution of $y$ [16]. However, we want the A4NT network to only imitate the style of $y$, while keeping the content from $x$. Thus, we explore different means to enforce the semantic consistency between the the input sentence and the A4NT output.

### 3.4. Preserving Semantics

We want the output sentence, $\tilde{s_y}$, produced by $Z_{xy}(s_x)$ not only to fool the attribute classifier, but also to preserve the meaning of the input sentence $s_x$. We use a semantic loss $L_{\text{sem}}(\tilde{s_y}, s_x)$ to quantify the meaning lost or changed during the anonymization by A4NT . Simple approaches like matching words in $\tilde{s_y}$ and $s_x$ can severely limit the effectiveness of anonymization, as it will even penalize synonyms or alternate phrasing. In the following subsection we will discuss two different approaches to define $L_{\text{sem}}$, and later in Section 5 we compare these approaches quantitatively.

**3.4.1. 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 $\tilde{s_y}$. 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 A4NT 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 [31]. Here they use $l_1$ distance to compare the reconstructed image and the original image to impose semantic relatedness penalty on style transfer. 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 in this work, is to forgo the reconstruction and just compute 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 A4NT network $Z_{yx}$ using the word distribution probabilities at the output. This cyclic loss computation is illustrated in Figure 5. Accordingly, 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 $L_{\text{sem}}$, the higher the reconstruction probability and thus more meaning of the input sentence $s_x$ is preserved in the style-transfer output $\tilde{s_y}$.

**3.4.2. Semantic Embedding Loss.** Another approach to measuring the semantic loss is to embed the two sentences, $\tilde{s_y}$ and $s_x$, into a semantic space and compare two embedding vectors using standard distances in this vector space. The idea is that a semantic embedding method embeds 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 [48]

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 [48, 49]. We use the universal sentence embedding model from [48], pre-trained on the Stanford natural language inference dataset [50]. This model was shown to perform well on several transfer learning tasks on different datasets, including sentiment classification, paraphrase detection and semantic relatedness [48].

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:

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

### 3.5. Smoothness with Language Model 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 loss 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 [51]. The networks are trained by optimizing the loss functions described with stochastic gradient descent using the RMSprop algorithm [52]. 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 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.

## 4. Experimental Setup

We test our A$^4$NT network on obfuscation of three different binary 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). The restriction to binary attributes brings certain advantages and challenges. It simplifies the experimental setting and allows faster iteration with fewer models to train. On the flip-side it simplifies the task of attribute classifiers with only two classes, making them harder to fool.

### 4.1. Datasets

We run our experiments on two different real world datasets: Blog Authorship corpus [53] 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 can work across very different language corpora.

**Blog dataset:** The blog dataset is a large collection of micro blogs from blogger.com collected by [53]. 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. In our work 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. Note that the age-groups 19-22 are missing in the original dataset. Since this dataset consists of free form text written while blogging, sentence boundaries are not well marked with the period symbol. We

| **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.

use the Stanford CoreNLP tool to segment the documents into sentences. All numbers are replaced with the NUM token.

**Political speech dataset:** To test the limits of how far style imitation based anonymization can help protect author's identity, we also test our model on two well known political figures with very different verbal styles. We collected the transcripts of political speeches (transcribed from the actual speech) of Barack Obama and Donald Trump made available by the The American Presidency Project[1]. While the two authors talk about similar topics they have highly distinctive verbal styles and vocabularies in their speeches, 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 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 [54] to identify and replace these entities with entity labels.

The comparison of the two datasets can be found in Table I. The blog dataset is much larger in size 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 obfuscation (blog-age), gender obfuscation (blog-gender) and identity obfuscation (speech dataset). Detailed analysis of our model presented in Section 5.2 is done on the validation split of the blog dataset, in the blog-age setting, containing 2,799 documents and 518,268 sentences.

## 4.2. Evaluation Methods

We evaluate our models w.r.t. 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. We use classifiers trained independently on real data and not the ones used in GAN training. The classifier used in GAN training are not used, to avoid evaluating the A$^4$NT network against the specific classifier it has learnt to fool. In a real-world setting, we will not have access to the parameters of the attribute-classifier used by the adversary to train our A$^4$NT network and it should be able to do well against adversaries it has not trained on.

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

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

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

To quantify semantic similarity, we use the Meteor [55] metric. It is used in machine translation and image captioning to evaluate the similarity of the candidate text using a reference text. Meteor compares the candidate text to one or more references by matching n-grams, allowing for soft matches using synonym and paraphrase tables. We compute the Meteor score between the generated and input text and use this 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 a human evaluation study on a subset of the test data of 745 sentences. We ask human annotators on Amazon Mechanical Turk to judge the semantic similarity of the generated text from our models.

## 4.3. Baselines

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

**Autoencoder** We can 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 change writing style of a piece of text without hand designed rules is to use machine translation software. The idea is to pass the text through machine translation from a source language through multiple intermediate languages until finally translating back to the source language. The hope is that through this round-trip the style of the text has changed, whereas the meaning is preserved. This was used

1. http://www.presidency.ucsb.edu

| 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 classification networks in different settings. All attribute classifiers perform well and better than the document-level random chance baselines (0.62 for speech), (0.53 for age), and (0.50 for gender)

in the PAN authorship obfuscation challenge recently [14] as a solution to mask author identities automatically.

Following this idea we use the Google machine translation service[2] to perform round-trip translation on our input sentences. We have tried a varying number of intermediate languages, results of which will be discussed in Section 5 Since Google limits 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.

## 5. Experimental Results

We test our model on the three settings discussed in section 4 with the goal to understand if the proposed A$^4$NT network is able to sufficiently fool the attribute classifiers to protect the anonymity of the author attributes. Through quantitative evaluation done in Section 5.1, 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 3.4 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 5.2, 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 3.4.2 and is referred to as *FBsem*. The second uses the reconstruction likelihood loss discussed in Section 3.4.1 instead, and is denoted by *CycML*. Finally, *CycML+Lang* uses both cyclic maximum likelihood and the language smoothing loss described in Section 3.5.

### 5.1. Quantitative Evaluation

Before discussing 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 3.1 on all three settings. Table II shows the F1-scores of the trained attribute classifiers on the training and the validation splits of the blog and the speech datasets. Document-level scores are obtained from accumulating the class log-probability scores on each sentence in a document before picking the maximum scoring class as the predicted label. We also tried hard voting to accumulate sentence level decisions, and observed that the results with hard voting follow the same trend across datasets and splits (including results with A$^4$NT network in the following subsection).

We see that on the smaller political speech dataset, the attribute classifier is able to easily discriminate between the two author identities, 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). In all three tasks, the performance on sentence-level is worse than on document-level classification. 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 classifier F1-score as our primary metric when evaluating the effectiveness of A$^4$NT networks.

**5.1.1. Quantifying Privacy Gains from A$^4$NT .** We can now evaluate the performance of our A$^4$NT network using the attribute classifiers presented in the previous section. To obfuscate the authorship of a document, the A$^4$NT 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$^4$NT network $Z_{Teen-Adult}$. This style-transferred document is input to the attribute classifier of the source attribute and F1-score of this classifier is computed. This measures the effectiveness of the style transfer in terms of privacy. Additionally, meteor score is computed between the source sentence and the A$^4$NT output, to measure the semantic similarity preserved.

Table III shows the result of this evaluation on three settings. On the small speech dataset all methods, including the autoencoder baseline described in Section 4.3, successfully fool the attribute classifier. They all obtain F1-scores below the chance-level, with our A$^4$NT networks doing slightly better. However the meteor scores of all models is significantly lower than in the blog dataset, indicating significant amount of semantics 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$^4$NT models however do much better, with all of them being able to drop the F1-score below the random chance level.

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 sentence.

2. https://translate.google.com/

| **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 | 0.68 | 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$^4$NT FBsem | *0.43* | *0.08* | 0.40 | *0.45* | *0.39* | 0.53 | *0.12* | *0.00* | 0.28 |
| A$^4$NT CycML | *0.49* | *0.20* | 0.57 | *0.41* | *0.44* | **0.79** | *0.11* | *0.00* | **0.29** |
| A$^4$NT 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.

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 meteor 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 human evaluation discussed in Section 5.1.2 and also produces better qualitative samples as will be seen in Section 5.2.

**Generalization to other classifiers:** A question that arises from the experiments in table Table III is if the A$^4$NT networks are only able to fool the classifier architectures it was trained with in the GAN setup or can they generalize to unseen classifier architectures. This is important if the system is to be applied to protect the privacy of author attributes against unseen adversaries.

To test this we train a second attribute classifier network on the blog-age setting. This network consists of a convolutional layer on top of the LSTM network described in Section 3.1. Results from evaluating the text generated by the A$^4$NT networks using this "holdout" classifier are shown in Table IV. First we note that the holdout classifier has similar performance to the LSTM classifier on the original text, achieving 0.87 document-level F1-score. We can see from this table that all three A$^4$NT networks generalize well and are able to drop the document-level F1-score of the holdout classifier below the random chance level. They perform slightly worse than on the LSTM classifier they have seen, but are still well below the random chance level. This confirms that the transformations applied by the A$^4$NT networks are not specific to the classifier they are trained with, but can also generalize to other adversaries.

We conclude that the proposed A$^4$NT 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.

**Different operating points :** Our A$^4$NT 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.

| **Model** | LSTM Classifier | | Holdout Classifier | |
|---|---|---|---|---|
| | Sent F1 | Doc F1 | Sent F1 | Doc F1 |
| Original text | 0.74 | 0.88 | 0.65 | 0.87 |
| Autoencoder | 0.69 | 0.85 | 0.71 | 0.84 |
| A$^4$NT FBsem | *0.43* | *0.08* | *0.45* | *0.11* |
| A$^4$NT CycML | *0.49* | *0.20* | *0.52* | *0.26* |
| A$^4$NT CycML+Lang | *0.54* | *0.32* | 0.56 | *0.43* |

TABLE IV: Performance of style transfer anonymization in fooling the classifiers, on blogdata (age)

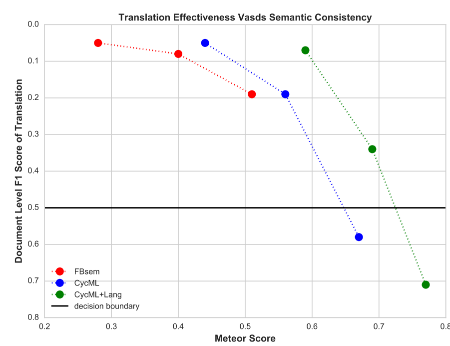


Figure 6: Operating points of three A$^4$NT models on testset

We illustrate this in Figure 6. Here five samples are obtained from each A$^4$NT model for each sentence in the test set. By choosing the sentence with minimum, maximum or random meteor scores, 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. The best samples (in meteor) from the *FBsem* model are still significantly worse than the worst samples from *CycML+LangLoss*.

**5.1.2. 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 human evaluations to judge the semantic 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 dataset in the blog-age setting. 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. 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 where there is no majority from human evaluators are excluded.

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 4.3.

For the machine translation baseline, we obtain style-transferred texts from four different language round-trips. We started with English $\rightarrow$ German $\rightarrow$ French $\rightarrow$ 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 human evaluation 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 human evaluation for our style-transfer models on two different operating points of $0.5$ F1-score and $0.66$ F1-scores, to obtain human judgments at two different levels of privacy effectiveness. The results are shown in Table V. 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 5.1 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 human evaluation between the *CycML+Lang* model operating at $0.79$ and the Google machine translation baseline with 3 hops. The operating point was chosen so that the two models were closest to each other in privacy effectiveness and meteor score. Results are shown in Table VI. We can see that our model wins over the GoogleMT baseline by approximately $16\%$ ($59.46\%$ vs $43.76\%$ rank1). 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.

## 5.2. Qualitative Analysis

In this section we will look at some qualitative examples of anonymized text produced by our A$^4$NT model and try to identify strengths and 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.

**5.2.1. Examples of Style Transfer for anonymization.** Table VII 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 VII).

**# 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 VII. 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 are able to fool the attribute classifiers thereby providing privacy to the author attribute against NLP adversaries.

**# 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 VII. 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 style, with changes like addition of "diz" (this) and replacing of "think" with "relized" (realized). These changes are learned entirely from the data, and would

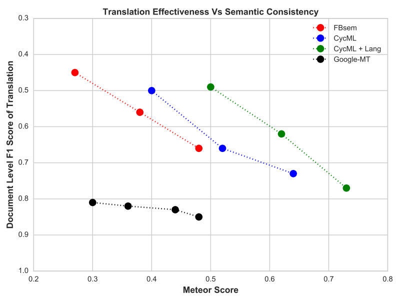


Figure 7: Privacy effectiveness and semantic consistency of our $A^4$NT networks and the Google Machine Translation baseline on the human eval test set

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

TABLE V: Human evaluation to judge semantic similarity of generated sentences to input. Three variants of our model are compared. The numbers show the % times the model was ranked first. Can add to more than 100% as multiple models can have rank-1 when they keep the input sentence identical

| Comparison | $A^4$NT CycML + Lang | GoogleMT |
|---|---|---|
| Operating point | 0.79 | 0.85 |
| % Rank 1 | **59.46** | 43.76 |

TABLE VI: Human evaluation comparing our best model to the Google Machine Translation baseline.

| # | 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 VII: 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.

be very hard to encode explicitly in a rule based system due to great 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 VII. For example, when an adult author mentions his "wife", the $A^4$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 many cases. Some example style transfers from Obama to Trump's style are shown in Table VIII. We see that the model inserts hyperbole ("better than anybody", "horrible horrible", "crooked"), references to "media" and "system", all salient characteristics 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

| 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 VIII: Qualitative examples of style transfer on the speech dataset from Barack Obama to Donald Trump's style

fool them the A$^4$NT network learns to make drastic changes to the input text.

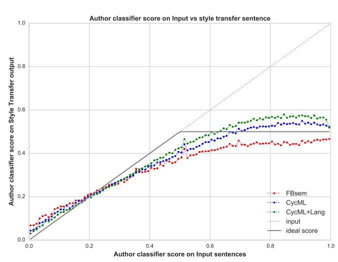


Figure 8: Output Privacy vs Privacy on Input.


Figure 9: Meteor score plotted against input difficulty.

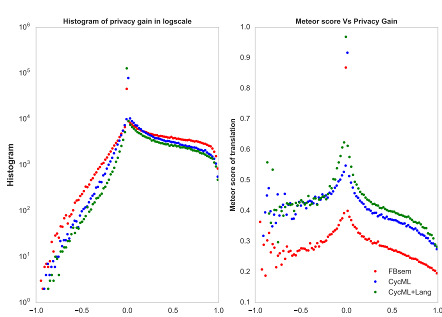


Figure 10: Histogram of privacy gain (left side) is shown alongside comparison of meteor score vs privacy gains.

**5.2.2. Performance Across Input Difficulty.** Figure 8 compares the attribute classifier score on the input sentence and the A$^4$NT output. Ideally we would want all theA$^4$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 $\sim 0.45$).

Now, we analyze how much of input semantics is preserved with increasing difficulty. Figure 9 plots the meteor score of the A$^4$NT output against the difficulty of input text. We see that the meteor is high and stable for sentences already across the decision boundary. These are easy cases, where the A$^4$NT networks do not need to intervene. As the input text 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 score 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 sentences 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 higher loss in semantics.

# 6. Conclusions

We have presented a novel 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 fool attribute classifiers by transforming the input text to imitate style of a target attribute. The network learns to perform these privacy targeted transformations automatically, by adversarially training against the attribute classifiers. The A$^4$NT network is trained end-to-end on non-parallel data and thus can be applied easily to new datasets.

Experiments on three different attributes namely age, gender and identity, showed that the proposed A$^4$NT network is able to effectively fool the attribute classifiers in all the three settings. Moreover, we show that the A$^4$NT network also performs well against unseen classifier architectures. This implies that the method is likely to be effective against previously unknown NLP adversaries.

We proposed a novel method to preserve the semantic similarity 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 don't reveal the attributes with obvious give-away words (school, work, husband etc.), but is lower on difficult sentences implying 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 dataset show the limits to which style transfer based approach can be used to hide attributes. On this challenging dataset 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.

Although, we tested our model on binary attributes, it can be extended to $m$-ary valued attributes. One simple method would be to pick a fixed target class (based on difficulty) for each input class and train $m$ separate style-transfer networks. Instead, methods to condition the style-transfer network effectively on input-target class pairs could be investigated, allowing the use of a single network to transfer styles between arbitrary attribute pairs.

## Acknowledgment

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

## References

[1] J. S. Donath *et al.*, "Identity and deception in the virtual community," *Communities in cyberspace*, vol. 1996, pp. 29–59, 1999.

[2] G. LaMonica and T. Fixel. (2011) Starting a revolution with technology. [Online]. Available: http://edition.cnn.com/2011/TECH/innovation/06/17/mesh.technology.revolution/index.html

[3] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, C. Parnin, and J. Stallings, "Gender differences and bias in open source: Pull request acceptance of women versus men," *PeerJ Computer Science*, vol. 3, p. e111, 2017.

[4] R. E. Steinpreis, K. A. Anders, and D. Ritzke, "The impact of gender on the review of the curricula vitae of job applicants and tenure candidates: A national empirical study," *Sex roles*, vol. 41, no. 7, pp. 509–528, 1999.

[5] P. Juola *et al.*, "Authorship attribution," *Foundations and Trends® in Information Retrieval*, vol. 1, no. 3, pp. 233–334, 2008.

[6] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the Association for Information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[7] S. Ruder, P. Ghaffari, and J. G. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *arXiv preprint arXiv:1609.06686*, 2016.

[8] J. R. Rao, P. Rohatgi *et al.*, "Can pseudonymity really guarantee privacy?" in *USENIX Security Symposium*, 2000, pp. 85–96.

[9] S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler, "Automatically profiling the author of an anonymous text," *Communications of the ACM*, vol. 52, no. 2, pp. 119–123, 2009.

[10] L. Sweeney, "Simple demographics often identify people uniquely," *Health (San Francisco)*, vol. 671, pp. 1–34, 2000.

[11] 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, pp. 300–314.

[12] 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*, vol. 7384. Springer, 2012, pp. 299–318.

[13] 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.

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

[15] 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)*, vol. 15, no. 3, p. 12, 2012.

[16] 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, pp. 2672–2680.

[17] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[18] W. Xu, A. Ritter, B. Dolan, R. Grishman, and C. Cherry, "Paraphrasing for style," *Proceedings of COLING 2012*, pp. 2899–2914, 2012.

[19] 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, pp. 212–226.

[20] 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.

[21] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 2, p. 7, 2008.

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

[23] 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, pp. 444–451.

[24] 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, pp. 173–185.

[25] 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.

[26] 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.

[27] J. Chung, K. Cho, and Y. Bengio, "A character-level decoder without explicit segmentation for neural machine translation," *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[28] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," *Transactions of the Association for Computational Linguistics (TACL)*, vol. 3, pp. 365–378, 2017.

[29] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, 2015, pp. 1486–1494.

[30] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] 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.

[32] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky, “Adversarial learning for neural dialogue generation,” *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 21472159, 2017.

[33] L. Yu, W. Zhang, J. Wang, and Y. Yu, “SeqGAN: sequence generative adversarial nets with policy gradient,” *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2016.

[34] 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.

[35] Z. Yang, W. Chen, F. Wang, and B. Xu, “Improving neural machine translation with conditional sequence generative adversarial nets,” *arXiv preprint arXiv:1703.04887*, 2017.

[36] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, no. 3-4, 1992.

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

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

[39] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, 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, pp. 1532–1543.

[42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[43] J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber, “Recurrent highway networks,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[44] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, “Densely connected convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[45] 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.

[46] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 1064–1074.

[47] 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.

[48] 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.

[49] 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, pp. 3294–3302.

[50] 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.

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

[52] 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*, vol. 4, no. 2, pp. 26–31, 2012.

[53] 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*, vol. 6, 2006, pp. 199–205.

[54] 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, pp. 363–370.

[55] 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, pp. 376–380.