The Importance of Suppressing Domain Style in Authorship Analysis

Sebastian Bischoff [1] **Niklas Deckers** [2] **Marcel Schliebs** [3] **Ben Thies** [2]
Matthias Hagen [4] **Efstathios Stamatatos** [5] **Benno Stein** [6] **Martin Potthast** [7]

[1]Technical University of Munich, sebastian.bischoff@tum.de
[2]Humboldt-Universität zu Berlin, {niklas.deckers | ben.thies}@hu-berlin.de
[3]University of Oxford, marcel.schliebs@oii.ox.ac.uk
[4]Martin-Luther-Universität Halle-Wittenberg, matthias.hagen@informatik.uni-halle.de
[5]University of the Aegean, stamatatos@aegean.gr
[6]Bauhaus-Universität Weimar, benno.stein@uni-weimar.de
[7]Leipzig University, martin.potthast@uni-leipzig.de

Abstract

The prerequisite of many approaches to authorship analysis is a representation of writing style. But despite decades of research, it still remains unclear to what extent commonly used and widely accepted representations like character trigram frequencies actually represent an author's writing style, in contrast to more domain-specific style components or even topic. We address this shortcoming for the first time in a novel experimental setup of fixed authors but swapped domains between training and testing. With this setup, we reveal that approaches using character trigram features are highly susceptible to favor domain information when applied without attention to domains, suffering drops of up to 55.4 percentage points in classification accuracy under domain swapping. We further propose a new remedy based on domain-adversarial learning and compare it to ones from the literature based on heuristic rules. Both can work well, reducing accuracy losses under domain swapping to 3.6% and 3.9%, respectively.

1 Introduction

Authorship analysis refers to a group of tasks in which authors are modeled based on their writing style. The most commonly studied task is authorship attribution, where for a text of unknown authorship it has to be decided who of a given set of candidate authors has written it. Under the assumption that every author possesses a unique writing style, unconsciously encoded into their writing, the attribution is solved by picking the candidate whose writing style best matches the one found in the text in question. Unlike a text's topic, however, writing style has proven difficult to be modeled reliably—the key challenge of any authorship analysis.

The style of a text as perceived by human readers results not only from an author's personal traits, but also from customs an author adopts due to genre, register, type, and topic. These concepts are vague, and each can be defined broadly but also subdivided hierarchically, rendering them difficult to be operationalized. Style forms a continuum, and the challenge is thus to discover (combinations of) style markers more likely to be determined by an author's personality rather than by domain customs.

Due to the lack of large-scale datasets, most machine learning approaches to authorship attribution are still based on manual feature engineering. It is commonly agreed to avoid certain features like content words, which rather capture topic than style. Yet, style-capturing features are mostly based on intuition about what (combination of) quantifiable characteristic(s) might represent an author. While many have argued why their features should capture author style, such claims are hardly ever substantiated experimentally: Typically, experiments do not control for domain-related style components, foreclosing conclusions about the true capabilities of a feature set in capturing author style. What is worse, a feature set that captures author style more "clearly" may even go unremarked compared to one that also captures other domain characteristics, since, in the typical experimental setups, the latter has a better chance of performing well.

Our contributions address these shortcomings for the first time: (1) We devise the first evaluation setup to explicitly measure the capabilities of style representations (Section 3). (2) To enable many corresponding experiments as well as the application of deep learning, we compile a large corpus of 1.4 million authors, each of whom has written long monographs (5.8 million stories in 44 languages) in 10,328 different topical domains (Section 4). (3) We apply domain-adversarial training to train the first neural style encoder that suppresses domain-specific information (Section 5). (4) In a series of experiments, the new style encoders are compared to competitive baselines, showing that traditional character trigram models are extremely susceptible to capturing domain style instead of author style, whereas domain-adversarial learning and heuristic rules are not (Section 6).

2 Related Work

Given the large number of papers about authorship analysis in general and attribution in particular (extensively surveyed by Stamatatos (2009) and Neal et al. (2017)), we focus on cross-domain attribution and recent related deep learning advances.

2.1 Cross-Domain Attribution

Forensic linguists hardly ever encounter attribution problems where the domains of the text of unknown authorship and the writing samples of the candidate authors are the same. Despite its practical importance, however, cross-domain attribution has been studied much less compared to same-domain attribution. Mikros (2007) has been the first to investigate the topic dependence of features on a dataset of 200 Greek newspaper articles from two topics. Features like frequent function words and word length had a high correlation with topic; character trigrams were not studied. Stamatatos (2013) later showed character trigrams to be highly effective on 1004 articles from The Guardian across two genres and four topics. In a follow-up, Stamatatos (2017) suggested to reduce domain-specific information through a text distortion phase before the feature extraction; one of our baselines.

Kestemont et al. (2018) first proposed fanfiction as a source for cross-domain authorship problems. For their shared task at PAN, a sample of 40 attribution problems was compiled ranging from 5 to 20 candidate authors with 7 texts each. The best-performing approach relied on an ensemble classifier based on character n-grams and the text distortion approach of Stamatatos (2017). We scale this idea by crawling a fanfiction corpus comprising more than a million authors who wrote millions of stories across thousands of domains.

Besides topic, also genre and language have been investigated as domain variables. Stamatatos (2013) reports character trigrams to perform well across two genres—yet worse than for cross-topic attribution—and Overdorf and Greenstadt (2016) report a drop of attribution performance across three social media genres compared to same-genre attribution. Bogdanova and Lazaridou (2014) study attribution across languages, combining machine translation with cross-language features, and van der Goot et al. (2018) improve cross-language gender-prediction by bleaching text through transforming lexical strings into more abstract features.

Altogether, none of the above studies shed light on the question how well author style is separated from domain style components.

2.2 Deep Learning-Based Attribution

A handful of attribution studies investigated the usefulness of deep learning. Ruder et al. (2016) outperform several traditional state-of-the-art attribution approaches in same-domain settings using convolutional neural networks (CNNs). However, they caution that “fine-tuned word embeddings that are sensitive to topical divergence between authors boost CNN performance.” Boumber et al. (2018) propose another CNN approach designed for multi-label attribution tasks, but also take advantage of topic information through word embeddings.

Hassan et al. (2017) achieve 95% attribution accuracy on scientific papers via a supervised LSTM and lexical and syntactic features. However, since topic seems to not have been controlled during training, it is unclear whether writing style was actually learned. Recently, Ding et al. (2019) suggested a representation learning approach using a neural network that combines character, lexical, syntactical, and topical information. Again, an analysis of how well the representations separate an author’s writing style from domain style was beyond the scope of their study, which seems problematic since topic information was exploited.

2.3 Style Transfer

Given the success of style transfer for images (e.g., Gatys et al., 2016), many studies also try to translate or rather paraphrase a text from a source style to a desired target style, the two main problems being the lack of large-scale parallel training data (i.e., texts written in both styles), and the lack of reliable evaluation metrics (i.e., humans need to assess the transfer quality) (Fu et al., 2018). Style transfer studies typically do not model an author’s personal writing style, but consider more broad style characteristics like a text’s sentiment (Hu et al., 2017; Shen et al., 2017; Zhang et al., 2018b), dialects of English (formal vs. informal, Shakespearean vs. simple) (Jhamtani et al., 2017; Jin et al., 2019; Kabbara and Cheung, 2016), or political slant (Prabhumoye et al., 2018). Some text style transfer studies even suggest not to disentangle latent representations of style and content (Dai et al., 2019)—exactly the opposite of what we require from a writing style representation for attribution.

2.4 Adversarial Training

In author obfuscation, the task is to paraphrase a given text to render an author's style imperceptible (Potthast et al., 2016). Typically, another text from the author is used as a reference for style similarity; recent approaches employ neural models (Emmery et al., 2018) and heuristic search (Bevendorff et al., 2019) to render the given text dissimilar. Similarly, Elazar and Goldberg (2018) attempt to remove markers from a style representation to protect its author's demographic details, such as gender, age, etc., through adversarial training. Furthermore, Grießhaber et al. (2020) use adversarial learning as a regularizer to avoid overfitting when training features for deep neural networks in low-resource settings.

In this paper, we also tackle cross-domain attribution with adversarial learning. We repurpose the adversarial transfer learning approach by Ganin et al. (2016), which, by training on labeled data and adversarial training on unlabeled data, promotes features that are discriminative for the intended learning task but at the same time indiscriminative for the differences between the labeled and the unlabeled data, thus enabling a robust transfer. We observe that cross-domain attribution may be tackled in a similar fashion: by training on the texts with respect to their author labels, and adversarial training on the texts with respect to their domain labels, our approach promotes features that are discriminative for the task of authorship attribution but at the same time indiscriminative for the text domain differences. We adapt and improve the architecture to obtain substantial improvements, yielding effective cross-domain writing style representations.

3 Measuring Author Style

How can the capabilities of a writing style model in capturing author style be reliably measured? The most commonly carried out experiment in the literature answers this question only under near-perfect conditions, but may otherwise yield misleading results. We argue that a careful control of the text domain is less error-prone and more insightful.

3.1 Constructing Attribution Problems

An authorship attribution problem consists of a text x of unknown authorship, and $k > 1$ texts from known candidate authors, where x is to be attributed to the candidate whose writing style it matches. A typical scheme for problem instances for experiments for $k = 2$ authors looks as follows:

Scheme S_1	training		testing	
authors	A	B	A	B
domains	P	Q	P	Q

where A, B are authors and P, Q domains, and the vertical mapping denotes which author has written in which domain. For training, texts from A and B take turns as x; for testing, previously unseen texts from A and B are used as x. This scheme readily extends to $k > 2$ authors.

The vast majority of experiments in the literature are within-domain, i.e., P = Q. Here, ensuring that all texts are mutually from the same domain includes checking their topic, genre, register, idiolect, time period, etc. (Grieve, 2007). The rationale is to ensure that style characteristics whose variation is due to domain style rather than author style are randomly distributed across the texts of all authors. Otherwise, latent domain differences may bias a model trained on a given style representation. In that case, the catch is that the biased model performs *better* on the test data, not worse, since it exploits domain style on top of author style.

The efforts that must be taken to properly construct a within-domain attribution problem should not be underestimated. Take genre as an example domain, which is only vaguely defined and hierarchical in nature: restricting texts to fiction may not be enough if A is a romance author and B a thriller author. Outside fiction, such problems take different forms: for instance, person A may write mostly work-related emails, and B personal ones. Ensuring domain equality with respect to genre as well as the other domains presumes in-depth knowledge of each individual text, severely limiting scalability.

3.2 Domain Swapping vs. Author Style

To explicitly quantify the capabilities of a style model in capturing author style, we propose to contrast the performance achieved with Scheme S_1 with that of the following:

Scheme S_2	training		testing	
authors	A	B	A	B
domains	P	Q	Q	P

where P $\neq$ Q and the relation between authors and domains is swapped between training and test, which we call *domain swapping*. Given a style model and a performance measure, by computing the difference $\Delta(S_1, S_2)$ of the performance the model achieves in experiments as per Schemes S_1 and S_2, one can directly observe the proportion of performance a model achieves due to exploiting domain style as opposed to author style. In this cross-domain experiment, the requirement of within-domain experiments to tightly control all conceivable domains that may affect style besides the author is relaxed: A style model and the data used to train it may be developed domain by domain, allowing for incremental improvements.

4 Attribution Problems from Fanfiction

Following Kestemont et al. (2018), we employ fanfiction—fiction written by fans of another's work, reusing its characters and settings—as a large-scale source of ground truth. The original work a given fanfiction is about is called its fandom. For well-known fandoms, such as Harry Potter, many different authors have written fanfiction. Moreover, many authors contribute to more than one fandom. We operationalize fandoms as (topic) domains in our experiments. Each fandom refers to a different "universe" of storytelling, quite distinct from others, yet also quite consistent within itself.

Unfortunately, the datasets provided by Kestemont et al. (2018) lack the required domain labels and comprise only a few hundred texts, so that we resorted to crawling fanfiction.net instead. We rigorously cleaned the texts to remove any mentions of author names, notes, and disclaimers. Moreover, we excluded "crossovers" combining aspects from two or more fandoms. Altogether, we compiled 5,800,292 stories in 44 languages from 1,400,958 authors writing in a total of 10,328 different fandoms.This corpus is made available to also enable future cross-domain attribution research.

4.1 Sampling Training and Test Data

From this corpus we construct instances of attribution problems as outlined above. The amount of available problem instances depends on the desired experiment scheme as well as the following constraints: our focus is on the English subset of the corpus, and the most basic problems with two authors and two fandoms each. We leave experiments with more languages and authors to future work.

The input size of our neural style encoder is presently 500 words, which is frequently considered to be about the minimum sufficient length to measure author style (e.g., Koppel and Schler, 2004). Although it is possible in principle to train a style encoder with such a small amount of text per author per domain, in our pilot experiments, we determined 100,000 words per author per domain (in the form of 500 word chunks) to be a sensible lower bound (Section 6.1). Moreover, to maximize reliable testing, we desire at least another 50,000 words per author per domain. Additionally, we employ a validation set of 50,000 words per author per domain for hyperparameter optimization and early stopping on our neural networks. We also make sure to never split chunks from a single fanfiction between training, validation and testing set. Regarding all of the above constraints for Scheme S_1 with P $\neq$ Q, the corpus allows for drawing (with replacement) a total of 1,260,082,646 problem instances. Regarding Scheme S_2 with P $\neq$ Q, a total of 93,238 problem instances can be drawn. In our evaluation, each experiment is repeated with at least ten distinct pairs of P, Q.

5 Domain-Invariant Style Encoder

Let x denote a text, $\mathbf{x}$ its two-dimensional representation, and X as well as $\mathbf{X}$ the sets of texts and text representations, respectively. An author mapping $a : X \rightarrow Y_a$ maps texts to their authors, and a domain mapping $d : X \rightarrow Y_d$ to their domains.

In what follows, we detail the input text representation, the adversarial training architecture, two style encoder variants, how we deal with the unbalanced class distributions as well as how we combine author and domain loss functions.

5.1 Input Text Representation

An input text x is a token sequence of 500 tokens. It is represented as a 2013$\times$500 matrix $\mathbf{x}$. The matrix is composed of 500 one-hot vectors, one for each token in x with 2013 dimensions each. 2001 dimensions represent the most frequently used words in the Brown corpus (Francis, 1965), 11 represent punctuation marks, and one one-hot vector $(1, 0, 0, \ldots)$ represents all other tokens in x. Beforehand, all punctuation marks are reduced to their ASCII equivalents. We refrained from introducing

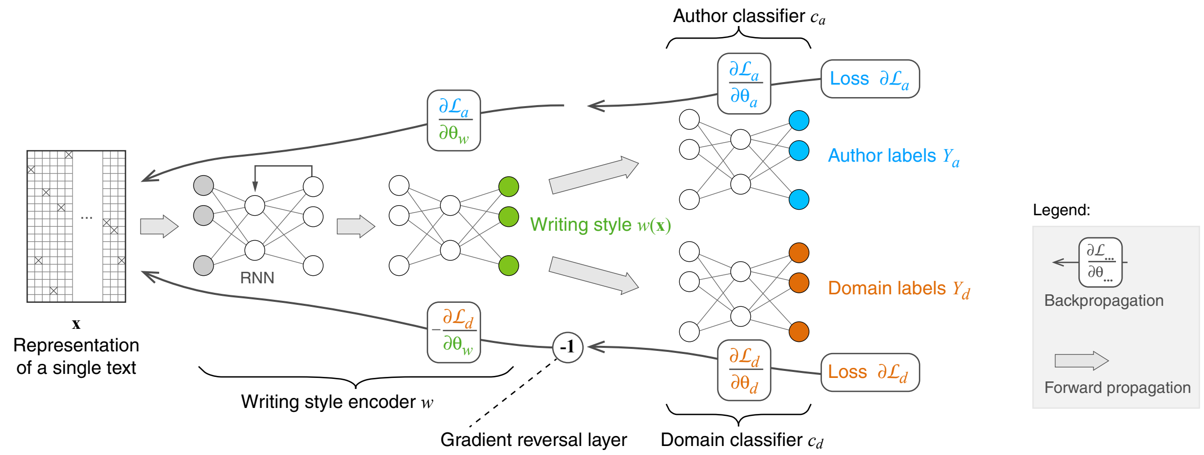

Author classifier c_a
$\frac{\partial \mathcal{L}_a}{\partial \theta_a}$
Loss $\partial \mathcal{L}_a$
$\frac{\partial \mathcal{L}_a}{\partial \theta_w}$
Author labels Y_a
...
RNN
Writing style $w(\mathbf{x})$
Legend:
$\frac{\partial \mathcal{L}_{...}}{\partial \theta_{...}}$
Backpropagation
Forward propagation
Domain labels Y_d
$\mathbf{x}$
Representation of a single text
$\frac{\partial \mathcal{L}_d}{\partial \theta_w}$
-1
$\frac{\partial \mathcal{L}_d}{\partial \theta_d}$
Loss $\partial \mathcal{L}_d$
Writing style encoder w
Gradient reversal layer
Domain classifier c_d

Figure 1: Architecture of our network including writing style encoder, author classifier and domain classifier. The texts are fed into the network as $\mathbf{x}$, a recurrent neural network is used to process the sequential property of a text. The final hidden state is used as a fixed-size summarization of the sequence (Goodfellow et al., 2016, p. 371) for the following fully connected layers to yield the writing style representation $w(\mathbf{x})$ (with parameters θ_w) of the full writing style encoder. The gradient can therefore only be obtained on the final hidden state which makes it difficult to optimize. This text representation is fed into the author classifier c_a (parameters θ_a) and domain classifier c_d (parameters θ_d). When backpropagating, we reverse the gradients for θ_d before the writing style encoder.

variables for these figures: it is our current trade-off between the maximal text length x, the size of its representation $\mathbf{x}$, and the batch size $m = 400$ of text representations $\mathbf{X}_{\text{batch}} \subset \mathbf{X}$ that we could simultaneously fit into our graphics card's RAM for training. We sought to maximize the number of examples in a batch while allowing for a reasonably-sized representation of each individual text.

5.2 Adversarial Learning Architecture

As illustrated in Figure 1, our goal is to find a writing style encoder $w : \mathbf{X} \rightarrow \mathbf{W}$ which maps text representations to style representations $\mathbf{W}$, so that an author classifier $c_a : \mathbf{W} \rightarrow Y_a$ is successful in mapping the style representations to authors, and a domain classifier $c_d : \mathbf{W} \rightarrow Y_d$ is *unsuccessful* in mapping the same style representations to domains.

We train our neural network batch-wise using $\mathbf{X}_{\text{train}} \subset \mathbf{X}$ to predict both Y_a and Y_d, minimizing the two classifier's loss functions:

$$\mathcal{L}_a(c_a(w(\mathbf{X}_{\text{train}}, \theta_w), \theta_a), Y_a);$$

$$\mathcal{L}_d(c_d(w(\mathbf{X}_{\text{train}}, \theta_w), \theta_d), Y_d),$$

where for w, c_a, and c_d their respective parameters $\theta_w, \theta_a, \theta_d$ are updated with respect to their joint loss $w(\mathbf{X}_{\text{train}}, \theta_w)$, $c_a(\mathbf{X}_{\text{train}}, \theta_a)$, and $c_d(\mathbf{X}_{\text{train}}, \theta_d)$ for the training texts $\mathbf{X}_{\text{train}}$.

By setting up the backpropagation of the prediction losses for both classifiers so that the obtained prediction performance for the author labels Y_a is maximized while that for the domain labels Y_d is minimized, the encoder learns to encode author style while avoiding domain style. This is accomplished by reversing (negating) the gradient of the domain classifier when propagated back to the style encoder. A gradient descent of negated gradients equals a gradient ascent of the original gradients. Therefore, while the optimizer updates the weights in the domain classifier to better predict the domain, at the same time, the weights in the style encoder are updated such that the style vector becomes less helpful for domain prediction.

Once the neural network has been trained using $\mathbf{X}$, we retain only its encoder: it takes a representation $\mathbf{x}$ of a given (previously unseen) text x as input and derives the style vector $w(\mathbf{x})$, which is well-suited to predict x's author label y_a but unsuited to predict x's domain label y_d. The encoder suppresses domain style while retaining author style, rendering $w(\mathbf{x})$ a domain-invariant author style vector for text x.

Dropout (Srivastava et al., 2014) is used for regularization and its resemblance of ensemble learning (Baldi and Sadowski, 2013). We use dropout on the input and all fully-connected layers. Batch normalization (Ioffe and Szegedy, 2015) ensures better learning dynamics. This is especially important in our case because the interaction of the two different losses makes the problem hard to optimize. Contrary to the results of Ioffe and Szegedy (2015), we cannot reduce the dropout rate and still get the same performance. We use Adam (Kingma and Ba, 2014) as an optimizer with Xavier initialization (Glorot and Bengio, 2010) for the output layers with sigmoid activation functions and He initialization (He et al., 2015) for fully connected layers with ReLU activation functions.

5.3 Style Encoder Variants

We consider two variants of our style encoder. The first one (Encoder 1), also illustrated in Figure 1, exploits the sequential nature of our text representation $\mathbf{x}$ by feeding it into a recurrent neural network (RNN) in the form of an LSTM (Hochreiter and Schmidhuber, 1997). The representation resulting from the LSTM is then fed into a convolutional layer to obtain the writing style representation. Although this architecture does improve over a convolutional layer in isolation, we observe that only the loss of the representation originating from the LSTM's final step is taken into account.

As a second variant (Encoder 2), in order to extract more information for the optimization of the style encoder's weights, we do not only use the loss when predicting the author on the full text representation $\mathbf{x} = \mathbf{x}_{1:500}$ (i.e., one-hot encoded token 1 to token 500), but the losses from n predictions based on the style representations obtained after the LSTM has read $\mathbf{x}_{1:\frac{1}{n}500}, \mathbf{x}_{1:\frac{2}{n}500}, \ldots, \mathbf{x}_{1:500}$. The individual losses are combined as follows:

$$\sum_{i=1}^{n} \lambda_i \mathcal{L}_a(a(w(\mathbf{X}_{1:\frac{i}{n}500}, \cdot), \cdot), Y_a),$$

weighting the prediction on $\mathbf{x}_{1:\frac{1}{n}500}$ the least and the one on $\mathbf{x}_{1:500} = \mathbf{x}$ the most. This way of training our style encoder can also be viewed as parameter sharing between individual networks trained on texts of length $\frac{1}{n}500, \ldots, 500$.

5.4 Unbalanced Author and Domain Classes

Unbalanced class distributions are often balanced using approaches like oversampling an underrepresented class or undersampling an overrepresented class. The rationale is to balance the influence of each class (i.e., in terms of macro accuracy). However, these sampling techniques are not directly applicable to problems formed of pairs $(y_d, y_a) \in (Y_d, Y_a)$ of labels. Changing the distribution of Y_d also changes that of Y_a, and vice versa. We therefore use the macro author accuracy and correct the author loss by weighting the texts from author A with m/n_{A}, where m is the number of authors and n_{A} is the number of texts written by author A. Likewise, the domain accuracy and domain loss are calculated.

It should be noted that not only the total numbers of activity n_{A} and n_{D} vary, but also the activity $n_{\mathrm{A,D}}$ of authors A in domains D. While this does not affect the validity of the accuracy calculations, it may affect the effectivity of the adversarial learning: if the correlation between authors and domain is 1, i.e., if author labels can be mapped to domain labels, the performance of the author classifier also correlates with the performance of the domain classifier. This problem is outlined in the next section.

5.5 Combining Author and Domain Loss

To obtain a joint loss function $\mathcal{L}(\ldots, (Y_a, Y_d))$, it might seem natural to add author loss and domain loss, since both losses are to be minimized. However, in our case, author and domain are correlated, which requires an adaptation of the loss function: Ideally, the domain accuracy is not reduced to 0, but to a random guess based on the writing style. This yields a lower bound of accuracy $1/n_{\mathrm{D}}$, which may be higher due to correlations between author and domain, presuming we achieve a high author accuracy. When simply adding the losses, the domain loss $\mathcal{L}_d(\ldots, Y_d)$ will be decreased below that boundary at the expense of author accuracy.

Instead, we employ another loss combination method based on the lower bound of the domain accuracy given the current author accuracy. When passing a single text through the current state of the network, it provides a vector $\mathbf{p}$ of prediction probabilities for each author. Assuming that all domain information is eliminated from the style representation, the author predictions might still be used to perform a prediction of the domains. The column-normalized matrix $\mathbf{N}$ of activity values $n_{\mathrm{A,D}}$ contains the conditional probabilities $P(D|A)$, i.e., the probability for each author A to write in some domain D. This matrix is applied on the predicted author probabilities $P(A)$ to receive the unconditional domain probabilities $P(D)$ as a vector $\mathbf{q}$:

$$P(D) = \sum_{A} P(D|A) \cdot P(A), \quad \text{or} \quad \mathbf{q} = \mathbf{N} \cdot \mathbf{p}.$$

As the domain classifier is optimized, the network reaches a domain macro accuracy performing at least as good as the macro accuracy of the predictions implied by $\mathbf{q}$ (calculated by selecting the most probable domain for each text). We call this lower

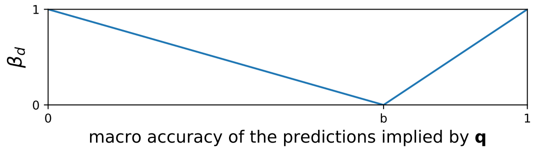


Figure 2: Determination of β_d for loss combination.

bound b. In order to perform as good as b, the domain classifier reproduces the author classification and empirically estimates the matrix $\mathbf{N}$.

In mid-training of the network, the style representation might still contain some domain information. Thus, the domain macro accuracy of the network will be higher than b. The larger this difference is, the more important it gets to eliminate domain information. This can be achieved by increasing the portion β_d of the domain loss according to Figure 2. Finally, the combined loss is defined as an affine combination of author and domain loss:

$$\begin{aligned}\mathcal{L}(\ldots,(Y_a,Y_d)) &= \beta_d\,\mathcal{L}_d(\ldots,Y_d)\\ &\quad+(1-\beta_d)\,\mathcal{L}_a(\ldots,Y_a).\end{aligned}$$

6 Evaluation

This section reports on a series of experiments to study whether and to what extent character trigram representations capture domain style information, and, whether and to what extent domain style can be successfully suppressed. Regarding the latter, we compare heuristic rules that have been applied in the literature with our new domain-adversarial learning approach and our two writing style encoder variants. With our experiment series, we want to shed light on the following questions:

1. Does domain-adversarial learning compete with traditional approaches to authorship attribution in a traditional setting?
2. What is the impact of domain swapping on all models under investigation? In particular: Can domain style be successfully suppressed?
3. What is the generalizability of our style encoders across fandoms?

Experimental Setup The experiments pertaining to Question 1 establish that all models under investigation are on par with each other. This includes the reconciliation of the training requirements of the different machine learning paradigms.

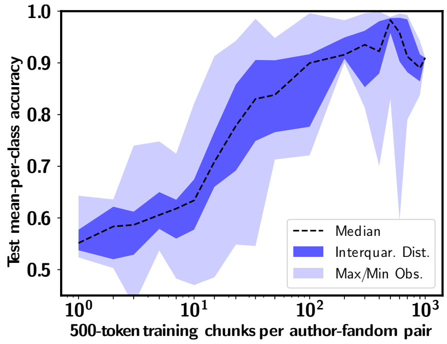


Figure 3: Author macro accuracy for test set over several text lengths.

Key goal is the creation of a setup in which all models are provided with the same amount of information for training. Regarding the experiments pertaining to Question 2, achieving this kind of fairness while at the same time meeting the requirements of the domain swapping experiment *and* those of domain-adversarial learning in terms of scale of training data presented a unique challenge. With all constraints combined, the fanfiction data we compiled provided for a sufficient amount of training and test data, allowing for ten repetitions of every experiment.

The models under investigation include: (1) A standard character trigram representation in conjunction with the three learning algorithms support vector machine (SVM), naive Bayes (NB), and random forest (RF). (2) A reproduction of the rule-based domain style suppression approach by Stamatatos (2018), which uses character trigrams after applying text distortion algorithms: replacing tokens with multiple asterisks (MA) as per their length, or with a single asterisk (SA); retaining only exterior characters (EX) of words in a dictionary, or their last two (L2) characters. (3) The two writing style encoder variants (Encoder 1 and Encoder 2) introduced above. In pilot experiments, all models have been meticulously optimized with regard to their respective parameters.

As performance measure, we employ the mean macro accuracy over at least ten problem instances for every experiment.

6.1 Traditional Authorship Attribution

This experiment investigates the performance of all models within the following setup:

	training				test			
author	A	A	B	B	A	A	B	B
fandom	P	Q	P	Q	P	Q	P	Q

	(a)		(b)			(c)			(d)
Classifier	**Traditional Attribution**		**Zero-Knowledge Swapping**			**Class-Imbalance Swapping**			**Cross-Fandom Generalization**
	default	suppression	normal	swapped	Δ	normal	swapped	Δ	
SVM	96.8%	–	98.9%	67.4 %	−31.5%	99.3%	83.9%	−15.4%	92.6%
NB	95.3%	–	100%	44.6 %	−55.4%	100%	51.5%	−48.5%	91.7%
RF	95.2%	–	97.4%	50.6%	−46.8%	97.3%	49.9%	−47.4%	87.5%
SVM MA	–	96.7%	97.2%	90.1%	−7.1%	98.0%	94.1%	−3.9%	94.8%
SVM SA	–	97.3%	97.4%	89.2%	−8.2%	98.1%	94.5%	−3.6%	95.2%
SVM EX	–	96.4%	98.6%	67.3%	−31.3%	99.0%	88.4%	−10.6%	93.9%
SVM L2	–	96.9%	98.8%	69.0%	−29.8%	99.0%	89.1%	−9.9%	91.7%
Encoder 1	95.5%	95.1%		–		90.8%	86.9%	−3.9%	92.8%
Encoder 2	95.0%	94.5%		–		91.0%	85.7%	−5.3%	91.4%

Table 1: Mean macro accuracies on testing sets for classification of two authors who wrote in two fandoms. (a) Results of the experiment in Section 6.1, where *default* pertains to training the classifier to just predict the authors, and *suppression* to training it in a way to reduce the domain style. (b) Results of the experiment in Section 6.2. (c) Results of the experiment in Section 6.2. The results reported here for our network are based on adversarial training. (d) Results of the experiment in Section 6.3 The results reported here for our network are based on adversarial training.

In this setup, two authors A and B have written in two fandoms P and Q, and an equal amount of their writing in both fandoms is used for training and testing, which corresponds most closely to a traditional attribution experiment from the literature. However, to allow for domain-adversarial learning, two fandoms must be present. Recall that this setup has been instantiated ten times without replacement from our fanfiction corpus for different pairs of authors A, B, and fandoms P, Q.

Important variables in this regard are the size $|x|$ of an individual text x used for training or test, and the number $|X_{\mathrm{train}}|$ of such texts X_{train} per author and per fandom. Regarding the former, due to the constraints imposed by the adversarial learning approach (see Section 5.1), $|x|$ cannot exceed 500 tokens, which must be propagated to all other models for reasons of fairness. Regarding the latter, however, deep learning models require large amounts of training data when trained from scratch, which, too, must also be extended to all other models. To determine the amount of text required to reliably train our model, we train and test it on varying numbers of text chunks of 500 tokens each. Using the above experimental setup, we determine the macro accuracies over the number $|X_{\mathrm{train}}|$ of available chunks for training. The results can be seen in Figure 3, where mean performance exceeds 90% accuracy at 200 chunks (100,000 words, the length of a book), which we choose as least amount of training text for each problem instance drawn from our corpus in all subsequent experiments, equally distributed across authors and fandoms.

Table 1a shows the accuracies of all models in this experimental setup, and Figure 4 the respective performance distributions. We distinguish models that actively suppress domain style from ones that do not (default). For Encoders 1 and 2, we disabled adversarial learning once to also supply performance values in a default learning setup. Otherwise, we observe that our fandom prediction accuracy converges to 50% in adversarial learning in about two hours on a single GTX 1080.

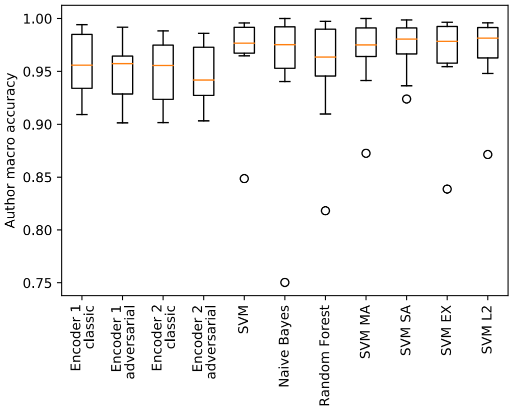


Figure 4: Distribution of accuracies as per experiment Section 6.1.

As can be seen, all models achieve very good accuracies between 94% and 97%. This performance is due to the amount of training data available; furthermore, the commonly held belief that 500 token chunks are sufficient to measure style is corroborated. Interestingly, neither does adding heuristic rules for domain suppression to the SVM change its a priori performance a lot (compare SVM with its variants MA, SA, EX, and L2), nor does adding adversarial training affect the performance of our writ-

	training		normal test		swapped test	
fandom	R	S	R	S	R	S
author A	200	0	max	0	0	max
author B	0	200	0	max	max	0

Table 2: Zero-knowledge domain swapping: During training, a model has no access to the one relation between authors and fandoms, whereas during swapped testing, the situation is reversed.

	training		normal test		swapped test	
fandom	T	U	T	U	T	U
author E	600	10	max	0	0	max
author F	10	600	0	max	max	0

Table 3: Class-imbalance domain swapping: During training, a model has highly imbalanced access to both relations between authors and fandoms, whereas during swapped testing, the situation is reversed.

ing style encoder. Despite their slight advantage in mean performance, the SVM models have a higher chance of severe misclassification, as the outliers in Figure 4 show. As outlined in Section 3, a setup like this does not reveal whether the performance difference is due to more or less usage of domain style information as opposed to author style, so that no conclusion about the relative effectiveness of the two suppression paradigms (heuristic rules vs. adversarial learning) can be drawn. Likewise, one cannot conclude that SVM-based models work "better" than our style encoder in terms of representing author style, since we cannot rule out that the better-performing models only perform better due to exploitation more domain style information.

6.2 Domain Swapping

This experiment investigates the performance of all models within the following setup:

	training		test			
			normal		swapped	
author	C	D	C	D	C	D
fandom	R	S	R	S	S	R

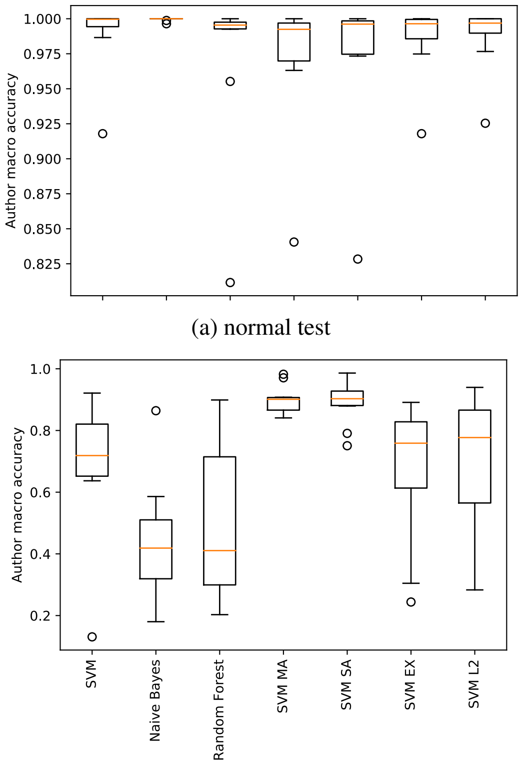
It contrasts a traditional authorship attribution situation with our novel domain swapping experiment. Two authors C and D have written texts in both domains R and S. For training, the models can learn from only one relation between authors and domains C-R and D-S, whereas for testing, either the same relation is used ("normal"), or a swapped relation C-S and D-R. This way, model bias that is due to an undesired exploitation of domain style rather than the desired representation of author style can be measured.

We consider two kinds of domain swapping experiments: (1) zero-knowledge swapping, and (1) class-imbalance swapping. The first variant, as shown in Table 2, maximizes the potential for confusion during training: the models never see an author in writing in the other author's respective fandom. However, this setup forecloses domain-adversarial learning, since the adversarial component cannot be trained in the absence of information about the domain to be suppressed. The second variant, as shown in Table 3, relaxes the first variant by allowing for many examples of one author-fandom relation and only a few ones of the reverse relation during training, while swapping the imbalance for testing. This allows for adversarial training while approximating zero-knowledge swapping.

Zero-Knowledge Swapping Table 1b shows the accuracies of all except the Encoder models for zero-knowledge swapping alongside the accuracy delta between normal and swapped testing, and Figure 5 shows the respective performance distributions. The character trigram models which do not apply any measure to suppress domain style suffer severe drops of accuracy under swapping: The naïve Bayes model, which under normal conditions achieves a perfect accuracy, drops 55.4 percentage points, falling even below random performance, so that reversing all of its decisions would yield a better performance. Similarly, the random forest-based and the SVM-based trigram models drop 46.8 and 31.5 percentage points, respectively.

This leads us to the conclusion that the traditional way of building author style models is highly susceptible to learning domain style instead. Unless the domains are carefully controlled—which imposes severe practical limitations—these models are prone to pick up domain artifacts or be fooled by adversaries.

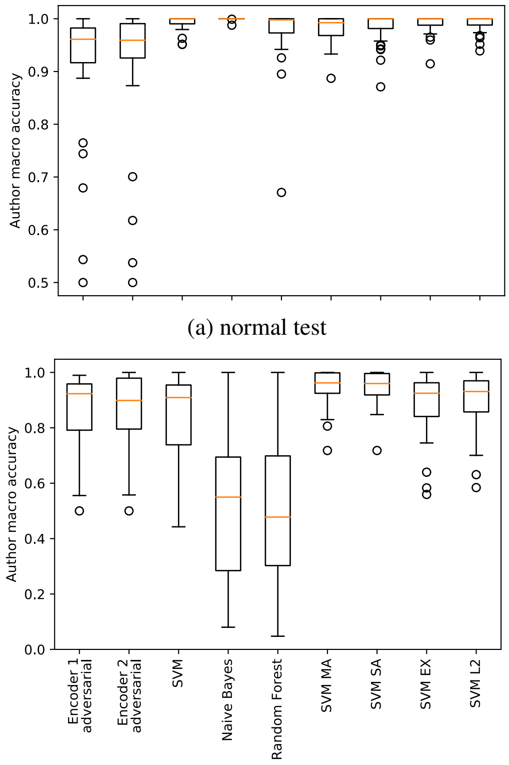
Regarding the SVM-based models that apply heuristic rules to suppress domain style, their performance varies from similarly high drops in performance to much more sensible drops of 7.1 and 8.2 percentage points for the MA and the SA rules. Regarding the performance distributions in Figure 5, in a normal test, all models perform quite



(a) normal test

(b) swapped test

Figure 5: Author macro accuracy for (a) normal test set and (b) swapped test set as constructed in Section 6.2.



(a) normal test

(b) swapped test

Figure 6: Author macro accuracy for (a) normal test set and (b) swapped test set as constructed in Section 6.2.

consistently, whereas, in the swapped test, the performance distributions has a large variance with the exception of the SVM MA and SA.

Class-Imbalance Swapping Table 1c shows the accuracies of all models for class-imbalance swapping alongside the accuracy delta between normal and swapped testing, and Figure 6 shows the respective performance distributions. The impact of domain swapping on the NB and RF models is comparable to that in the previous experiment. The SVM model, however, reduces its drop from 31.5 to only 15.4 percentage points. Consequently, also the SVM-based models that apply domain suppression drop much less than with zero-knowledge swapping. Regarding our Encoder models, their a priori performance under normal conditions more strongly deviates from the performance of the SVM-based models (around 90% vs. 99% accuracy). This is likely due to the effect of active domain suppression by adversarial training: The other models can exploit domain knowledge to achieve their high performance, whereas the Encoders cannot do so to the same extent. Moreover, the adversarial training component is perhaps penalized due to the high class imbalance. The drop in accuracy, between normal and swapped testing is the same as for the SVM-based models, which tells us that there is no relative disadvantage of adversarial training compared to heuristic rules: This is important, since it shows that domain style suppression can be learned instead of requiring handcrafted rules from experts.

6.3 Cross-Fandom Authorship Attribution

This experiment investigates the performance of all models within the following setup:

	training				test	
author	A	A	B	B	A	B
fandom	P	Q	P	Q	R	R

In this setup, two authors A and B have written in three fandoms P, Q, and R. The training set comprises an equal amount of text from both authors in the fandoms P and Q, while the test set is exclusively composed of equal amounts of text from both authors in fandom R. This setup allows gives insights into the generalizability of the style encoder across fandoms, when the test fandom is unknown at the time of training. Table 1d shows the accuracies of all models for this setup, and Figure 7 shows the respective performance distributions. As can

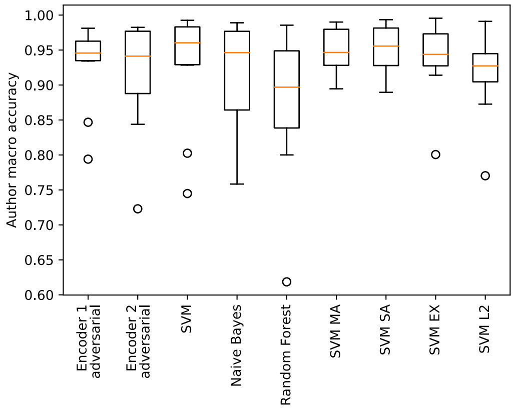


Figure 7: Author macro accuracy for test set as constructed in Section 6.3.

be seen, all models except for RF, tend to achieve a comparable accuracy of around 92%, which shows that all models generalize across fandoms, and our style encoder in particular.

In a second generalization experiment, we investigate the performance of all models within the following setup:

	retraining		test	
author	C	C	D	D
fandom	R	S	R	S

In this setup, two authors C and D have written in two fandoms R and S. However, the model is trained only equal amounts of text from author C in both fandoms, while being tested on equal amounts of text from author D in both fandoms. The main purpose of our proposed style encoder is to extract an author's writing style from given texts in a fandom-invariant manner: i.e., the fandom of a text must not be predictable, even if the network has not been trained on this specific fandom. In this experiment, we therefore take model trained for the experiment in Section 6.1 and extract the style of texts from new authors and fandoms. We then retrain the part of the network that uses the style vector to classify fandoms. The mean fandom accuracy for adversarially trained Encoder 1 is 51.6% and that for Encoder 2 is 49.9%, indicating the intended failure: The extracted style contains no usable fandom information.

The main function of our proposed network is that it extracts a writing style from given texts, which should be fandom-invariant, i.e., the fandom of a story must not be predictable, even if the network has not been trained on this specific fandom. For this purpose, we take the network trained in Section 6.1 and extract the style of texts from new authors and fandoms. We then retrain the part of the network that uses the style vector to classify fandoms, as shown in the following table:

7 Conclusion and Future Work

Representing writing style poses many theoretical and practical problems: Not only is there no clear definition of what, precisely, constitutes an author's writing style, and what separates it from other kinds of writing style due to domains like genre, register, and topic; as we reveal in this paper: the traditional way of modeling writing style is highly susceptible to representing domain style instead of author style. This revelation is due to a new kind of experimental setup that is afforded by a new, large-scale corpus that we build as part of this work. The corpus comprises large amounts of long texts written by many different authors in many different domains, allowing for the first time to study the relation of author style and domain style for at least one domain of interest.

We demonstrate that basic models employing character trigrams as features are reduced to near-random performance under our experimental setup. We further show that two approaches to reduce domain style yield promising results: on the one hand, a heuristic approach of handcrafted rules to select features, and on the other, a new domain-adversarial learning approach that learns to extract writing style while suppressing domain style. Both appear to be working equally well. However, our adversarial learning approach is presumably more adaptable to new situations.

On a more critical note, the amount of data required to train our model to high performance is considerable, and it stands to reason that such amounts of data cannot be easily compiled in practical situations, and even less so when more than one domain has to be suppressed. In this regard, the good performance of the handcrafted rules at least open the door to a less resource-intensive approach. Nevertheless, the amount of training data required to train a neural network may border on impracticality, unless we can create pre-trained style models that need only be fine-tuned, e.g., like BERT. Future work in this direction is necessary, and perhaps the directions we have shown as well as the data we have compiled will help to go into this direction.

References

Pierre Baldi and Peter J Sadowski. 2013. Understanding dropout. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2814–2822. Curran Associates, Inc.

Janek Bevendorff, Benno Stein, Matthias Hagen, and Martin Potthast. 2019. Bias Analysis and Mitigation in the Evaluation of Authorship Verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6301–6306. Association for Computational Linguistics.

Dasha Bogdanova and Angeliki Lazaridou. 2014. Cross-language authorship attribution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2015–2020, Reykjavik, Iceland. European Language Resources Association (ELRA).

Dainis Boumber, Yifan Zhang, and Arjun Mukherjee. 2018. Experiments with convolutional neural networks for multi-label authorship attribution. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation. *arXiv:1905.05621 [cs]*.

S. H. H. Ding, B. C. M. Fung, F. Iqbal, and W. K. Cheung. 2019. Learning stylometric representations for authorship analysis. *IEEE Transactions on Cybernetics*, 49(1):107–121.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial Removal of Demographic Attributes from Text Data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21.

Chris Emmery, Enrique Manjavacas Arevalo, and Grzegorz Chrupała. 2018. Style Obfuscation by Invariance. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 984–996, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

W. Nelson Francis. 1965. A standard corpus of edited present-day american english. *College English*, 26(4):267–273.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. *AAAI Conference on Artificial Intelligence*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030.

L. A. Gatys, A. S. Ecker, and M. Bethge. 2016. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

J. Grieve. 2007. Quantitative Authorship Attribution: An Evaluation of Techniques. *Literary and Linguistic Computing*, 22(3):251–270.

Daniel Grießhaber, Ngoc Thang Vu, and Johannes Maucher. 2020. Low-resource text classification using domain-adversarial learning. *Computer Speech & Language*, 62:101056.

Saeed-Ul Hassan, Mubashir Imran, Tehreem Iftikhar, Iqra Safder, and Mudassir Shabbir. 2017. Deep stylometry and lexical & syntactic features based author attribution on plos digital repository. In *Digital Libraries: Data, Information, and Knowledge for Digital Lives*, pages 119–127, Cham. Springer International Publishing.

K. He, X. Zhang, S. Ren, and J. Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models.

In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19, Copenhagen, Denmark. Association for Computational Linguistics.

Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. 2019. Unsupervised Text Style Transfer via Iterative Matching and Translation. *arXiv:1901.11333 [cs]*.

Jad Kabbara and Jackie Chi Kit Cheung. 2016. Stylistic Transfer in Natural Language Generation Systems Using Recurrent Neural Networks. In *Proceedings of the Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods*, pages 43–47, Austin, TX. Association for Computational Linguistics.

Mike Kestemont, Michael Tschuggnall, Efstathios Stamatatos, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast. 2018. Overview of the author identification task at PAN-2018: Cross-domain authorship attribution and style change detection. In *Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018*, pages 1–25.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*.

Moshe Koppel and Jonathan Schler. 2004. Authorship verification as a one-class classification problem. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, page 62, New York, NY, USA. ACM.

Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. 2006. Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 659–660, New York, NY, USA. ACM.

George K. Mikros. 2007. Investigating Topic Influence in Authorship Attribution. In *SIGIR '07 Amsterdam. Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*.

Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. 2017. Surveying stylometry techniques and applications. *ACM Computing Surveys*, 50(6):86:1–86:36.

Rebekah Overdorf and Rachel Greenstadt. 2016. Blogs, Twitter Feeds, and Reddit Comments: Cross-domain Authorship Attribution. *Proceedings on Privacy Enhancing Technologies*, 2016(3):155–171.

Martin Potthast, Matthias Hagen, and Benno Stein. 2016. Author Obfuscation: Attacking the State of the Art in Authorship Verification. In *Working Notes Papers of the CLEF 2016 Evaluation Labs*, volume 1609 of *CEUR Workshop Proceedings*. CLEF and CEUR-WS.org.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv preprint arXiv:1609.06686*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6830–6841. Curran Associates, Inc.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556.

Efstathios Stamatatos. 2013. On the Robustness of Authorship Attribution Based on Character N-Gram Features. *Journal of Law & Policy*, pages 421–439.

Efstathios Stamatatos. 2017. Authorship attribution using text distortion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1138–1149, Valencia, Spain. Association for Computational Linguistics.

Efstathios Stamatatos. 2018. Masking topic-related information to enhance authorship attribution. *Journal of the Association for Information Science and Technology*, 69(3):461–473.

Rob van der Goot, Nikola Ljubešić, Ian Matroos, Malvina Nissim, and Barbara Plank. 2018. Bleaching Text: Abstract Features for Cross-lingual Gender Prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–389.

Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. 2019. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5310–5319.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *Proceedings* *of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 585–596, USA. Curran Associates Inc.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018a. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pages 335–340, New York, NY, USA. ACM.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018b. Style Transfer as Unsupervised Machine Translation. *arXiv:1808.07894 [cs]*.