Differentially Private Authorship Obfuscation Using AR-RE

Nico Manzonelli, Joseph Zuccarelli

13 May 2022

1 Introduction

There are many cases in which the author of some text document wishes to remain anonymous. For example, a journalist may wish to obscure their identity when publishing an unsettling political piece or a righteous political activist could want to spread their message on social media without being identified. In the past, authors could simply publish under a pseudonym to obscure their identity. However, with recent developments in the field of natural language processing (NLP), accurate computational authorship attribution methods can be used to re-identify anonymous text documents. As a defense to such attribution, researchers study authorship obfuscation, which uses computational methods to release text while concealing enough information to protect against authorship attribution [1, 2]. However, many obfuscation techniques rely on empirical evaluation and provide little to no theoretical guarantees. Therefore, this begs the question of whether it is feasible to release anonymized text documents using differential privacy (DP)? In the following paper, we review the ER-AE model for differentially private text generation and test the robustness of this model against a new dataset.

1.1 Authorship Verification

Digital text forensics is a field of text mining aimed at examining the originality and credibility of information in electronic documents and extracting information about the authors of said documents [3]. One of the most important tasks within this field is authorship attribution. Researchers frame the task of authorship attribution in many ways. For example, a simple form of the attribution problem is to successfully identify the author of an anonymous document from a small set of candidate authors. In this paper, we consider the *fundamental problem* of authorship attribution–authorship verification [4]. Given two pieces of text, authorship verification is defined as the task of determining whether the two texts have the same author.

We propose an authorship verification attack which is similar to a re-identification attack. In a re-identification attack, the adversary attempts to re-identify an individuals private data by using an auxiliary source. Authorship verification attacks follow a similar format. In an

authorship verification attack, we assume the adversary holds a piece of text from a known author (the auxiliary source) and attempts to verify if a corresponding piece of text has the same author (private data). Consider the following more formal definition. Suppose that there exists a document $\mathcal{D}_{\mathcal{U}}$ with unknown authorship. The adversary possesses a set of known candidate authors $\mathcal{A}$ and a set documents constructed by the known authors $\mathbb{D}_{\mathcal{A}}$. The adversary selects a document $\mathcal{D}_a \in \mathbb{D}_{\mathcal{A}}$ composed by author $a \in \mathcal{A}$. The attacker uses a model to compare the document within a known author, $\mathcal{D}_a$, to the unknown document, $\mathcal{D}_{\mathcal{U}}$, in order to determine if author a constructed $\mathcal{D}_{\mathcal{U}}$ (refer to Figure 1 for a visual depiction of an authorship verification attack) [5].

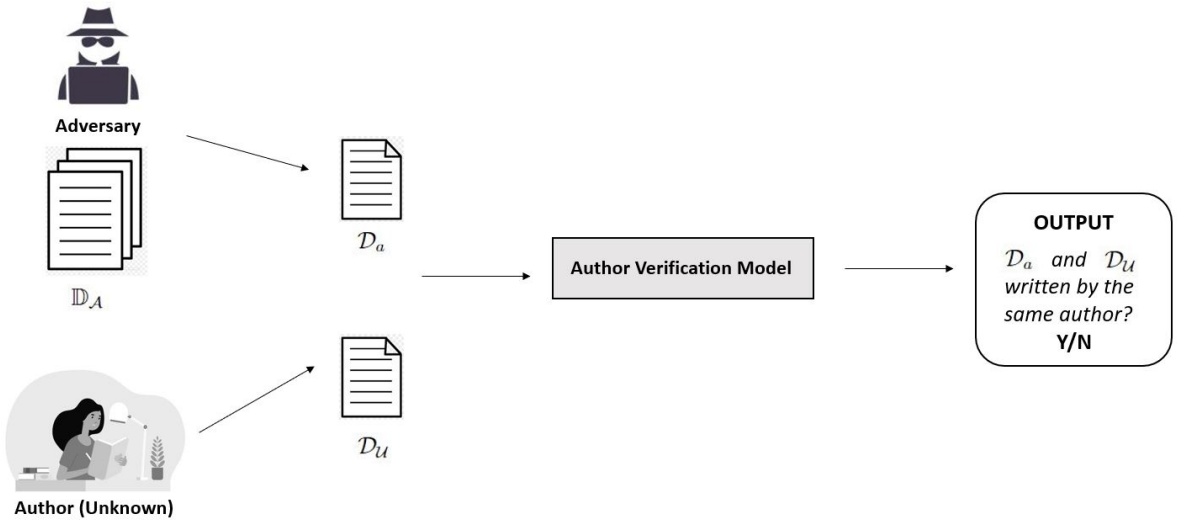

Figure 1: Authorship Verification Attack

1.2 Literature Review

Because text data sources are very abundant and NLP use cases are expanding as the machine learning research community grows, there are increased privacy concerns associated with releasing text data. In order to address such privacy concerns, researchers are beginning to apply DP to perform private computations over text data and releasing DP generated text. There exists some research in generating DP text under generalised forms of differential privacy [6, 7]. However, we implement a model that adheres to the pure form of differential privacy.

1.2.1 Differentially Private Text Generation

Before we introduce the ER-AE model, it is important to frame the core principles of DP in the context of text data, as introduced by Weggenmann and Kerschbaum [8]. Each text document, x, which could be a sentence, paragraph, etc., contained within a collection of

documents, X, is represented as a feature vector consisting of words from a fixed vocabulary. To define adjacency, we consider two datasets X and X' that each have one observation. In line with Weggenmann and Kerschbaum's definition concerning the strictest definition of adjacency, X is considered adjacent to X' if X' can be obtained by editing the record within X. In this case, an edit is considered a change at the word-level within the document. Note that this defintion of adjacently assumes that each document is anonymized independently which allows us to consider an X with only one document. In order to bound the sensitivity of text generation, we define a rating function, ρ, that takes an input text document, x, and an output text document, z, and returns a rating in $\mathbb{R}$. The sensitivity, Δ, of two adjacent documents is defined mathematically as follows:

$$\Delta = \max_{z \in Z} \max_{x \sim x'} \rho(x, z) - \rho(x', z).$$

By bounding the output of ρ to [0,1], the worst-case sensitivity is $\Delta = 1$. This allows researchers working in this space to satisfy the definition of DP text generation. Given the privacy parameter, ϵ, and the length of all text documents bound by l, a text generation function is differentially private if for each $\{x_i, \ldots, x_l\} = x \in X$ the generation function outputs $z = \{z_i, \ldots, z_l\}$ such that each z_i satisfies $(\epsilon, 0)$-DP. Therefore, the generated text document is $l(\epsilon, 0)$-DP via simple composition.

1.2.2 The ER-AE Model

The *Embedding Reward Auto Encoder* (ER-AE) model proposes using a seq2seq autoencoder with a two-set exponential mechanism and custom *REINFORCE* reward function to generate differentially private text (refer to Algorithm 1 for the fully defined generation procedure of ER-AE and Figure 2 for a visual depiction of the ER-AE model architecture) [9]. For modeling purposes, each text document $x = \{x_i, \ldots, x_l\}$ is converted to a sequence of word-embeddings such that $x = \{Em(x_i, \ldots, Em(x_l)\}$ where Em maps each word in the vocabulary set of V to an m- dimensional vector. The encoder takes each x and passes it into a recurrent neural network (RNN) with two layers of stacked bidirectional gated recurrent units (GRU) to generate the latent space, $E(x)$ [1]. The decoder, or generator, takes $E(x)$ and passes it to another RNN with two layers of stacked bidirectional GRUs. The autoencoder is trained traditionally via the *REINFORCE* reward function. Differential privacy is added in the inference phase; for each $z_i \in z$, the decoder calculates a logit weight for every candidate token $v \in V$, and applies the softmax function to generate $Pr[z_i = v]$. Given the resulting probability mass function, we choose a token set, T_i, using the two-set exponential mechanism tuned with the sensitivity of the rating function. Finally, by randomly sampling a token from T_i for each z_i, the algorithm generates a private text document, z. Because of the *REINFORCE* reward function the two-set exponential mechanism is more likely to

[1] Recurrent neural networks (RNNs) are a class of neural networks adapted to work with sequential data or time series data, such as text documents [10]. The gated recurrent unit (GRU) model is an advanced version of the standard RNN model that better regulates the flow of information within a network [11].

output a T_i with multiple meaningful replacement words because the model learns to assign higher probabilities to semantically similar words.

Algorithm 1 Generation Procedure of ER-AE

Input: Text: x, Parameters: θ, Encoder: $E_\theta()$, Generator: $G_\theta()$, Privacy Budget: ε.
  Produce the latent vector: $E_\theta(x)$.
  Get probabilities of new tokens: $\Pr[\bar{x}] \leftarrow G_\theta(E_\theta(x))$.
  for $i \leftarrow 1$ to length of x **do**
    Build two candidate token sets based on $\Pr[\bar{x}_i] : S, O$. Apply exponential mechanism to choose token set: T. Randomly sample new i-th token from $T : \bar{x}_{dp}[i]$.
  end for
Output: Differentially Private Text: $\bar{x}_{dp}$.

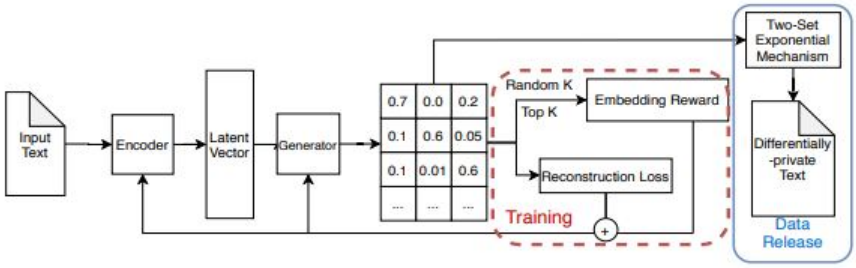

Figure 2: Overall Architecture of ER-AE

Bo et al. provide proofs for the privacy guarantees of their model and the two-set exponential mechanism in Appendix A and B [9]. They prove that each z_i satisfies $(\epsilon + ln(s))$-DP where s is the size of V. Therefore, each z satisfies $l(\epsilon + ln(s))$-DP. For a reasonably sized V and l, the authors use 20000 and 50 respectively; therefore, we see how there can be significant privacy loss for each privatized document. As a limitation of the model, the authors assert that ER-AE performs poorly for longer documents due to the privacy accounting issue.

1.3 Study Objectives

In this project, we implement the ER-AE model to generate differentially private text for authorship obfuscation. We aim to test the robustness of the ER-AE model by evaluating it's performance on a new dataset. More specifically, we systematically feed the model long-text documents with a clear method of privacy accounting and evaluate the model's results in terms of obfuscation performance and semantic preservation while varying the privacy parameter and the dimensionality of the word-embeddings used in training.

2 Implementation Details

2.1 Data

The dataset under observation in this project is the PAN dataset for authorship verification. PAN is an organization that hosts scientific events and shared tasks on digital text forensics and stylometry.[2] They have provided data for and hosted an authorship verification task challenge consistently since 2020. We requested and were approved for access to a PAN-owned dataset that contains 220k pairs of training text documents and 34k pairs of testing documents[12]. The text documents included in the dataset were gathered from fanfic.net, a long established forum for posting fan fiction [13].

In line with the adversarial model, we consider the training set and the ER-AE model trained on said dataset as the adversary's auxiliary information. In a real-world context, it is possible that the adversary's auxiliary information includes documents published by the target author, but under their true identity. In order to ensure that we properly met this assumption, we confirmed that there were instances of the same author occurring in the test and train set prior to performing any model evaluation. Additionally, due to computational cost, we randomly sampled 100 pairs of documents and their labels to function as our test-set.

2.2 Application to Long-Text Documents

Instead of increasing l to fit the desired length for each long document, we break down each long document into a series of short documents by sentence. Then, we apply ER-AE to generate differentially private sentences (removing sentences that exceed the preset length l) and recombine the newly privatized sentences to form one long document of text. Via the composition theorem of differential privacy, the resulting document is $n * l * (\epsilon + ln(s))$-DP where n is the number of sentences with length $\leq l$ in the document at hand.

The authors of ER-AE use pre-trained BERT embeddings of dimension (768,) [14]. However, we opted to use GloVe embeddings to vary the dimensions of the word embeddings used for model evaluation [15]. In each evaluation, we test the model trained on word embeddings with dimensions (100,), and (300,).

3 Evaluation

First, we train 2 ER-AE models on a collection of sentences from the train-set. We set the size of the vocabulary, s, and the max length of a sentence, l, to 50 for each model and vary the dimensions of the GloVe word embeddings. Then, for each model, we vary epsilon and apply AR-AE to each document to evaluate model performance based on authorship obfuscation and semantic preservation.

[2] https://pan.webis.de

3.1 Metrics

3.1.1 Authorship Obfuscation

In order to measure the performance of the ER-AE model in authorship obfuscation, we implement a mock-adversary that uses naive PAN baseline prediction model to attempt to re-idenitfy the author of privatized documents. This approach first represents documents as TF-IDF weighted bag-of-character-four-grams, then optimizes and re-scales the cosine similarity between two documents to generate a pseudo-probability that represents the likelihood that the two documents have the same author [3]. Successful authorship obfuscation is measured as a decrease in the classifiers roc-auc score from the baseline: 0.564.

3.1.2 Semantic Preservation

Similar to Bo et. al, we use Google's pre-trained Universal Sentence Embedding model to transform sentences into one latent vector that represents the sentences' semantics [4]. To measure semantic preservation for each document, we take the average cosine similarity between the latent vector for each sentence in the original document and each sentence of the DP document. Then, we average all of the document-average-cosine similarities and record the standard deviation.

3.2 Results

Prior to evaluating the AE-RE model's performance at different values for the document-wideprivacy budget, we first print a few DP-generated sentences to inspect generated sentence quality. Table 1 displays DP-generated text from random sentences within a random document in the test-set at various values of ϵ_s, the privacy budget for the sentence. Note that each word in the sentence is generated with privacy loss parameter equal to $\frac{\epsilon_s}{50}$.

Original Sentence: But instead of heeding her warning, he instead turns his gaze back to the ocean.

ϵ_s	GloVe 100d	GloVe 300d
1	wills respectful of research of ignoring activated stops undertake dimed eyes back to cherubs air sunlight	ravens vibrations dividing dropped her random owls he kay at his gaze back shouldered the ocean
5	strategy instead of bother his l chips he disregard turning sherlocks gaze back to the obvious	junior of many pitch dulled computer steadfast he turns turns his gaze back to the ocean
10	plainly bred of travel his goblin aggravation he impossibly imitate projects gaze back to the cave surface	ravens unacceptable of icy her warning assemble he turns villains her gaze back to the ocean
∞	aeris instead of towards her alarm abruptly not makes the his gaze back the the ocean	intelligence instead of heading her warning he turns turns at his gaze go to the ocean

Table 1: Example Generated Sentences

As shown in Table 1, for all values of epsilon and each GloVe dimension presented, ER-AE generates a different sentence free of punctuation. Note that at each epsilon the 300-dimensional embeddings typically capture more of the original message (each use the terms "gaze" and "ocean"). However, the model appears to be struggling to generate meaningful text with added privacy. As we can see when $\epsilon_s = \infty$, even our non-DP text generation using ER-AE struggles to produce great output. Therefore, for the experiments outlined

[3] https://github.com/pan-webis-de/pan-code/tree/master/clef20/authorship-verification
[4] https://tfhub.dev/google/universal-sentence-encoder/4

below, we expect the noise of non-DP text generation process to significantly contribute to our results. However, overall we expect ER-AE with higher embedding dimesionality to perform better in terms of semantic preservation, but worse in terms authorship obfuscation due to the model's ability to retain key information from the original sentence.

Next, we evaluate each model's performance in terms of authorship obfuscation. Figure 3 displays ER-AE authorship obfuscation performance for each GloVe embedding dimension while varying the document wide privacy budget. The y-axis of Fig. 3 represents authorship obfuscation, which we define as the average decrease in the mock-adversary's model AUC score across all pairs of documents in the test-set, where one document is generated using ER-AE. As mentioned in Section 3.1.1, the baseline adversary AUC score is 0.564, which is calculated on the raw test-set.

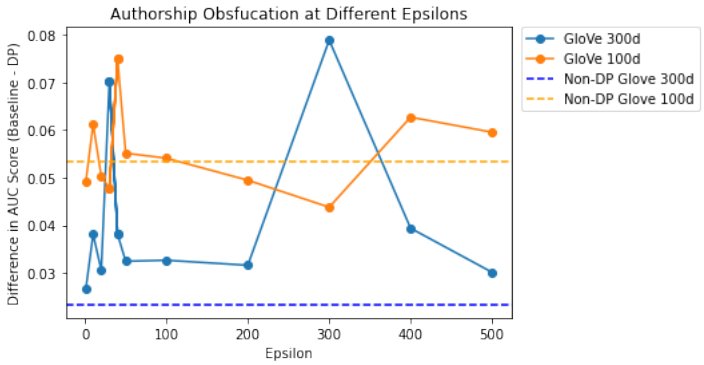

Figure 3: Authorship Obfuscation Performance for Varying Word Embedding Dimensions

Figure 3 highlights that models using higher dimensional embeddings typically perform worse at obfuscating the author for most epsilons. Intuitively, this makes sense because there is higher quality text generation with higher dimensional embeddings, which in turn causes an authorship verification model to perform better.

Making generalizations across epsilons is difficult. We expect that at larger values of epsilon there is more privacy loss and subsequently less effective authorship obfuscation. However, we do not observe the expected trend. The dashed-lines on Figure 3 indicate authorship obfuscation quality when we set $\epsilon = \infty$ to mimic generating text from our model in a non-DP manner. In this case, ideally the authorship obfuscation would be zero, which indicates that our model generates very similar text. At the least, we expect that authorship obfuscation with $\epsilon = \infty$ should be lower than any ϵ generated by the model, even at extremely high epsilons. We observe that this case for the 300 dimensional GloVe embeddings, but not with the 100 dimensional embeddings. This suggests that there is likely a significantly amount of noise associated with generating text using the model trained on 100 dimensional embeddings, and that introducing more noise via DP yields random, unintelligible results for any privacy budget. Although the model trained using 300 dimensional

embedding with $\epsilon = \infty$ has almost zero difference in AUC score, we still do not observe the expected trend in authorship obfuscation. We hypothesize that this discrepancy is due to two factors. First, due to computational limitations, as we limited our tests set to 100 document pairs and generated new documents for one trail at each epsilon. Second, because the documents contain many sentences, the privacy budget is stretched thin at both the sentence and word-level. The average document has 347 sentences padded to length 50, which means that on average each word is generated with privacy loss of $\frac{\epsilon}{17350}$.

Figure 4 displays ER-AE semantic preservation performance for each GloVe embedding dimension while varying the privacy budget of the entire document. Note that the y-axis represents semantic preservation, which we define as the average semantic preservation across all sentences in all generated documents from the test-set.

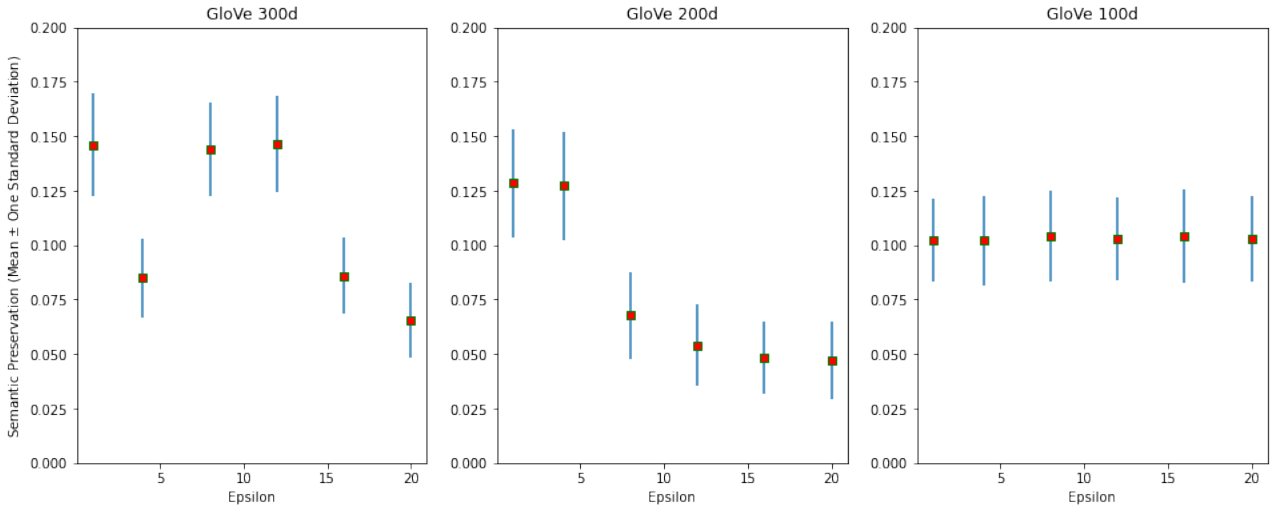

Figure 4: Semantic Preservation Performance for Varying Word Embedding Dimensions

Once again, we do not observe a clear trend in terms of the change in semantic preservation as we vary epsilon, which is possibly due to the privacy accounting issues mentioned above. However, it is worth noting that semantic preservation at the best epsilons decreases as we decrease the embedding dimension. This indicates that the model trained with higher embeddings captures more semantic meaning in training.

4 Conclusion

4.1 Discussion

Evaluating ER-AE on a new dataset with different embeddings allows us to make some key observations in regards to DP text generation. First, like all text generation models,

ER-AE is very sensitive to the training data and preprocessing steps. Because our implementation uses GloVe embeddings and our training set consists of many obscure (think fan-fiction specific) terms, the generated sentence quality suffers in our implementation. The authors of ER-AE use BERT embeddings on text corpora that contain more common everyday language. Therefore, the original implementation drastically outperforms our implementation in terms of semantic preservation.

While our implementation on long text suffers in producing meaningful text and preserving semantic meaning, we see that on a limited test set it performs well in authorship obfuscation. This is likely due to the fact that the DP generated text is almost unintelligible. We find that for our choices of epsilon ({1,10,20,40,50,100,200,300,400,500}) we do not find much difference in authorship obfuscation because of privacy accounting issues. The authors of ER-AE realize the expected relationship between privacy loss and authorship preservation on their shorter length documents, which indicates that long documents suffer from stretching the privacy budget too thin across epsilons (even up to 500). However, when working with privacy loss parameters that are large, we lose meaningful interpretations of privacy in general, which is a limitation of all DP text generating models.

4.2 Limitations and Future Work

One major limitation of our implementation is hardware based. Because we lacked the computing power to train models for many epochs and evaluate results on larger test sets for many iterations over more epsilons, our findings are difficult to interpret. As discussed previously, the model did not perform as expected, which could be a function of the computational limitations or the nature of generating longer DP text documents.

Our method is also limited in privacy budget accounting methods. Theoretically, the privacy loss of systematically privatizing each sentence in the full document should be in the worst-case equal to the privacy loss of increasing l to the length of the full document. This implies that systematically privatizing each sentence would result in privacy savings if the full document had at least one sentence with length $< l$. However, for modeling purposes, we must pad each sentence to length l with "pad" characters, which forces the privacy loss to be equal to that of the privacy loss of increasing l. To approach this problem, we could train multiple ER-AE models at varying l's and dynamically apply the model to sentences in the full document. If n_i is the number of sentences with max length l_i, the resulting document will be $\sum_i n_i{*}l_i * (\epsilon + ln(s))$-DP which would have less privacy loss than the system implemented in the original model. The main limitation to this approach is that training ER-AE is computational costly, and building/evaluating many models is unrealistic without a high-performance system.

Another avenue for future work relates to the DP text generation model itself. Instead of relying on the two-set exponential mechanism used to sample from the decoder, we could

train the decoder using differentially private optimization methods (i.e., DP-SGD). With a model trained using differentially private optimization methods, we can generate text the traditional way by finding the argmax of $Pr[z_i = v]$ for each $v \in V$. The use of DP-SGD removes the use for the complex two-set exponential mechanism. However, whether or not the proposed model will yield any privacy savings while preserving utility is an open question for further research.

References

[1] Asad Mahmood et al. "A Girl Has No Name: Automated Authorship Obfuscation using Mutant-X." In: *Proc. Priv. Enhancing Technol.* 2019.4 (2019), pp. 54–71.

[2] Martin Potthast, Matthias Hagen, and Benno Stein. "Author Obfuscation: Attacking the State of the Art in Authorship Verification." In: *CLEF (Working Notes)*. 2016, pp. 716–749.

[3] Martin Potthast et al. "A decade of shared tasks in digital text forensics at PAN". In: *European Conference on Information Retrieval*. Springer. 2019, pp. 291–300.

[4] Ph.D Moshe Koppel, Ph.D Jonathan Schler, and Ph.D Shlomo Argamon. "Authorship Attribution: What's Easy and What's Hard?" In: *Journal of Law and Policy* 21 (2013). URL: https://brooklynworks.brooklaw.edu/jlp/vol21/iss2/4/.

[5] Oren Halvani, Christian Winter, and Lukas Graner. "Assessing the applicability of authorship verification methods". In: *Proceedings of the 14th International Conference on Availability, Reliability and Security*. 2019, pp. 1–10.

[6] Natasha Fernandes, Mark Dras, and Annabelle McIver. "Author Obfuscation Using Generalised Differential Privacy". In: *CoRR* abs/1805.08866 (2018). arXiv: 1805.08866. URL: http://arxiv.org/abs/1805.08866.

[7] Natasha Fernandes, Mark Dras, and Annabelle McIver. "Generalised differential privacy for text document processing". In: *International Conference on Principles of Security and Trust*. Springer, Cham. 2019, pp. 123–148.

[8] Benjamin Weggenmann and Florian Kerschbaum. *SynTF: Synthetic and Differentially Private Term Frequency Vectors for Privacy-Preserving Text Mining*. 2018. DOI: 10.48550/ARXIV.1805.00904. URL: https://arxiv.org/abs/1805.00904.

[9] Haohan Bo et al. "ER-AE: Differentially-private Text Generation for Authorship Anonymization". In: *CoRR* abs/1907.08736 (2019). arXiv: 1907.08736. URL: http://arxiv.org/abs/1907.08736.

[10] Tomas Mikolov et al. "Recurrent neural network based language model". In: *Interspeech*. Vol. 2. 3. Makuhari. 2010, pp. 1045–1048.

[11] Kyunghyun Cho et al. "On the properties of neural machine translation: Encoder-decoder approaches". In: *arXiv preprint arXiv:1409.1259* (2014).

[12] Sebastian Bischoff et al. "The Importance of Suppressing Domain Style in Authorship Analysis". In: *CoRR* abs/2005.14714 (May 2020). URL: https://arxiv.org/abs/2005.14714.

[13] Mike Kestemont et al. "Overview of the authorship verification task at PAN 2021". In: *CLEF*. 2021.

[14] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. DOI: 10.48550/ARXIV.1810.04805. URL: https://arxiv.org/abs/1810.04805.

[15] Jeffrey Pennington, Richard Socher, and Christopher Manning. “Glove: Global Vectors for Word Representation”. In: vol. 14. Jan. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.