Nomen est Omen - The Role of Signatures in Ascribing Email Author Identity with Transformer Neural Networks

Sudarshan Srinivasan, Edmon Begoli, Maria Mahbub, Kathryn Knight
Oak Ridge National Laboratory
Oak Ridge, USA
{srinivasanss, begolie, mahbubm, knightke}@ornl.gov

2021 IEEE Security and Privacy Workshops (SPW) | 978-1-6654-3732-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/SPW53761.2021.00049

***Abstract*—Authorship attribution, an NLP problem where anonymous text is matched to its author, has important, cross-disciplinary applications, particularly those concerning cyber-defense. Our research examines the degree of sensitivity that attention-based models have to adversarial perturbations. We ask, what is the minimal amount of change necessary to maximally confuse a transformer model? In our investigation we examine a balanced subset of emails from the Enron email dataset, calculating the performance of our model before and after email signatures have been perturbed. Results show that the model's performance changed significantly in the absence of a signature, indicating the importance of email signatures in email authorship detection. Furthermore, we show that these models rely on signatures for shorter emails much more than for longer emails. We also indicate that additional research is necessary to investigate stylometric features and adversarial training to further improve classification model robustness.**

***Index Terms*—natural language processing, authorship attribution, transformer-based networks, attention-based models, adversarial perturbation, digital forensics**

I. Introduction

Authorship attribution [1], a task of identifying the author of a given task, is an important area of natural language processing (NLP) with applications in digital forensics [2], criminal justice and law enforcement [3], informational assurance, but also linguistics [4], and even anthropology. In particular, author attribution is of increasing concern to cyber defense-focused operations dealing with cybersecurity [5]. For example, law enforcement or digital forensic operation could be interested in sources and tracking of misinformation, criminal or threatening content, fraudulent claims, blackmail, or phishing attempts.

More generally, authorship attribution is a technique in natural language processing that focuses on recognizing authors of anonymous texts. Given a corpus, author identification models learn about an author's writing style including sentence length, vocabulary, word context, uniqueness, and repetition, use of digits, parts-of-speech (POS), and word n-grams.

In NLP, an author attribution model learns specific features from a pool of documents and then is able to identify the authors of those documents based on these learned features, thus increasing that model's probability of detecting any fraudulent and misleading documents. With the evolution of NLP, author attribution techniques have also progressed from tedious rule-based techniques to faster, automated neural network-based models. However, as we are marching forward, we are more focused on testing the robustness of these techniques rather than just improving their prediction accuracy. In this paper, we attempt to explore this least-inspected area by using state-of-the-art BERT [6] classification models and Enron e-mail dataset [7].

After investigating BERT-based email classification models, we noticed that email signature removal produced higher confidence in classifying longer emails than shorter ones, indicating that the signature may be significant, especially for shorter emails. Thus, our hypothesis is that the removal or replacement of email signatures will hinder the performance of attention-based [8] email authorship detection models. As such, our main contributions are to examine the relationship between email content size (number of words) and the presence of an email signature (formal or informal), and providing a possible explanation on why transformer-based models are not robust against signature perturbations.

Furthermore, we wanted to examine the attention-based model's performance and sensitivity to adversarial perturbations – i.e., the minimal change of the content that maximally confuses the transformer model. In this process, we observed that these models are very sensitive to certain perturbations to the input. Specifically, in our task, we found out that there is a significant loss of model classification performance based on the presence or absence of a signature in short emails. These observations are important to the authorship attribution problems that deal with the short-content messages which are especially relevant in social media forensics. We present our findings in the rest of the paper, and we point out that these findings warrant further investigation in how the adversarial content perturbations can impact the effectiveness of transformer-based digital content forensics.

DOI 10.1109/SPW53761.2021.00049

II. Related Work

In this work we are particularly concerned with email authorship attribution and the robustness of the classification models to adversarial perturbations. As such our work relates to two different Spears: 1) Authorship attribution (specifically in emails); 2) Adversarial perturbations in text.

Authorship attribution is a well-known and well-studied problem [9], [10]. Author attribution using distributed language representation was done in [11]. Here the authors used a vector-space representation for the author and disputed text according to the words and their nearby context. They then used cosine similarity to determine the authorship of the text. They evaluated their model on historic text and found their approach to work well. Convolutional neural networks (CNN) was used for determining authorship of tweets in [12].

An attention-based hierarchical neural network is used for capturing writing style in [13], which is then used to attribute authorship. Pre-trained language models are used for cross-domain authorship attribution in [14]. They combine a character-level RNN with a multi-head attention architecture and pre-trained language models to detect authors in cross-domain documents. LI et al., introduced a target-dependent method for authorship attribution, where they include information about the author directly in their model for author attribution of user reviews. They used a pre-trained BERT model and a CNN to extract features from the text and proposed a fusion method which was used to predict the author of the review. In [16], the authors perform authorship attribution on different datasets including the Enron email corpus, by fine-tuning pre-trained BERT models. In addition to the text, they include several stylistic and hybrid features such as length of text, number of words, and others as input to their model. Their results indicate high performance for many authors as resulting of input augmentation with stylistic features.

Sylometry-based authorship attribution is an important methodology where in authorship is detected based on linguistic writing style. Important contributions to this type of authorship attribution include [17], [18]. In [17], the authors apply stylometry to a multi-authored set of documents and use support vector machine (SVM) for classifying the authors of these documents. They define several different scenarios and show cases where their classifiers yielded good and bad performances. In [18], Greenstadt talks about the field of adversarial stylometry and how the author applied it to different datasets such as open source projects, underground hacker forums, and tweets.

Adversarial text [19] is a relatively new and rapidly growing field which involves adversarial attacks on machine learning models and associated defenses. A particular type of adversarial attack, known as, adversarial perturbations involves manipulating the input text to deteriorate the performance of the model. A common approach is to find the most words that are influential to the model prediction and manipulate them to force the model to misclassify the input. This strategy has been adopted by many include TextBugger [20], TextFooler [21], and DeepWordBug [22]. While the basic strategy across these works is the same, they differ in the way they determine the most important words of the input that are influential to the prediction by employing different word importance scoring functions. For a comprehensive review of adversarial text, we refer to [19].

In this work, we combine email authorship attribution with adversarial perturbation focusing only email signatures and email content length to determine what effect these have on the robustness of the model.

III. Dataset Details & Preprocessing

In this work, we used a subset of the famous Enron email dataset. The Enron corpus was made public during the legal investigation concerning the Enron corporation. The corpus consists of more than half a million messages with over 150 users. For our work we downloaded the version provided by Kaggle [2].

In order to get good model performance to test our adversarial perturbation algorithm, we chose a balanced subset of emails from the dataset. Specifically, we wanted to maximize the number of emails per author while also keeping the emails-per-author ratio similar for each author. By attacking a well-performing model, we can judge the quality of the attack in terms of performance and the robustness of the trained classifier.

During our preprocessing we only included emails that were *sent* by authors which were conveniently located in the *sent* folder for each author in the original dataset. We avoided conversations, email forward chains, and attachments contained within the text of the email. We also removed *to* and *from* addresses and URLs embedded within the email.

Table I: Dataset details by Author

Author Name	# Emails	# Train	# Test
Allen Phillip	1075	860	215
Chris Dorland	1000	800	200
Darron Giron	1081	865	216
Jeffrey Shankman	1172	938	234
Kevin Presto	908	727	181
Kimberly Watson	953	762	191
Louise Kitchen	1023	818	205
Lynn Blair	994	795	199
Mark Haedicke	1079	863	216
Michelle Cash	1090	872	218
Total	**10375**	**8300**	**2075**

We wanted to maximize the number of emails per author while also keeping the emails-per-author ratio similar for each author. After sorting the emails by author, we identified 10 authors that met our criteria, resulting in a subset of 10,375 emails. Table I shows the details of the dataset by author.

[2] https://www.kaggle.com/wcukierski/enron-email-dataset#

IV. Approach

Since the original transfer-based language model BERT was introduced, there have been many variants of it. In order to analyze the robustness of the transformer-based model, we selected 8 models for two reasons: 1) diversity - models vary in their handling of cases and the number of parameters in them; 2) performance in standard NLP tasks. The models we selected are ALBERT (*albert-base-v2*) [23], BERT (*bert-base-cased*, *bert-base-uncased*) [6], DeBERTa (*microsoft/deberta-base*) [24], DistilBERT (*distilbert-base-cased*, *distilbert-base-uncased*) [25], RoBERTa (*roberta-base*) [26], and SqueezeBERT (*squeezebert/squeezebert-uncased*) [27]. More information about the pretrained models are available in HuggingFace's documentation website [3].

A. Email Author Classification

We trained all of the 8 models following the same steps. First we downloaded the pretrained model for sequence classification and replaced the top layer with a 10-output linear layer corresponding to the 10 authors of the dataset. We then proceeded to train the entire model using the training data. Specifically, we do not *freeze* the bottom layers as that did not give very good performance. We use the same hyperparameters for all of the models with a learning rate of $1e^{-5}$, weight decay of $1e^{-2}$, and a maximum sequence length of 128 tokens. The value for maximum sequence length was determined empirically as more than 99% of the emails fall below this value. All of the models were trained on 80% of the dataset and tested on the remaining 20% of the data.

B. Signature Detection & Perturbation

Our signature detection algorithm targets the email signatures of the authors. Prior to analysis, we examined a representative sample of emails from each author to identify any signature patterns. Notably, we found that author signatures varied both among and within author email text. For example, one author might choose to sign her email with only a first name or just by using her first and last initials.

In order to account for the variety in signatures, we created a list of possible signature choices for each other. Choices include the author's full name, last name, first name, initials, and in some cases an abbreviated version of their first name. For example, a possible signature choice for author *Kimberly Watson* was *Kim*.

For detecting whether an email was signed by one of the authors, we scanned the email and compared its contents against the author signature choices. Assuming that email signatures tend to appear at the end of the email, our algorithm looked for signature choices in the last portion of email. By empirically determining that the last 20% of emails contain author signatures, we were able to check for signatures in both long and short emails. Once we determined the part of the email containing the signature, we used regular expressions to isolate the words in the strings to compare it against author signature choices.

[3] https://huggingface.co/transformers/pretrained_models.html

Table II: Model Performance on Unperturbed Emails

Model Name	All Emails	Signed Emails	Unsigned Emails
albert-base-v2	85.6	97.6	67.4
bert-base-uncased	88.0	98.2	72.6
bert-base-cased	87.8	98.4	71.5
deberta-base	88.4	98.2	73.4
distilbert-base-uncased	86.9	97.9	70.1
distilbert-base-uncased	87.1	98.6	69.7
roberta-base	87.8	98.6	71.3
squeezebert-uncased	87.0	98.0	70.2

After signature detection, we perturb the email by changing the signature in two ways: 1) *signature removal* – the signature is removed completely, which reduces the amount of information available to model; 2) *signature replacement* – the signature is replaced with a random string which has the *same length* as the signature, which changes the type of information available to the model. Both of these methods manipulate the email without access to the model's internal state, we perform an *untargeted black box* attack. While targeted attacks are possible using this algorithm, we will consider them for our future work.

V. Results & Discussion

We used a 80-20 split with a fixed seed for training and testing our models. This resulted in 8,300 emails for training and 2,075 emails for testing. We made sure that the author-to-emails ratio was similar in both the training and test sets. For performance measurement we use accuracy, which is the percentage of the correctly classified samples. All the results that we show below pertain to the test set.

A. Model Performance

Table II shows the accuracy of author detection of all of the models. In addition to showing the performance of the entire test set, this table also shows the performance of the models on the signed and unsigned emails separately. We see that all the models performed similarly, getting an average accuracy of 87.3% on all emails, 98.2% on signed emails, and 70.8% on unsigned emails.

Table III shows the model performance of all of the models on emails with their signatures perturbed (either removed or replaced). The values in the parenthesis indicate the drop in accuracy from performance of the model with the unperturbed emails as input.

We see that when the models are fed emails with signatures removed, there is an average drop of 36.1% and 59.8% accuracy on all emails and signed emails respectively. When the models are fed with emails that have their signatures replaced by a random string of the same length, there is an average drop of 36.3% and 56.4% accuracy on all emails and signed emails respectively.

The results from table III indicate that, there is a similar drop in accuracy when the signature is either removed or

replaced. Interestingly, when looking at only signed emails, signature removal has a bigger drop in accuracy than compared to signature replacement by an extra 3.4%. This could be because of the change in length of the email that results from removing content from it.

Both the results from table II and table III indicate the following about all the models on the test set: 1) Without signature perturbation, accuracy on signed emails is considerably higher than accuracy on unsigned emails; 2) When signature is perturbed, accuracy on signed emails plummet; 3) Both signature removal and signature replacement affect the models' accuracy. Taken together, we can infer that email signatures play a vital role in email authorship detection.

B. *Effect of Email Length and Signature*

We wanted to investigate how the length of emails and email signatures interact with each other to affect model prediction. To this end, we chose *bert-base-uncased* to further analyze its output. We chose the model due to its lowest drop in accuracy on emails with their signatures removed. We also chose the signature removal perturbation as part of this analysis, given its slightly higher negative effect on the model's performance.

Table IV shows the average length of the email in number of tokens produced by the tokenizer, the percentage of signed emails, and the average difference in confidence of the model's prediction of that author. The *confidence difference* is the difference between the model's prediction probability of the target author when the input is an unperturbed and when the input is a signature-removed email. Intuitively, the confidence difference illustrates the average of the percentage difference in confidence loss that the model loses when faced with a signature-perturbed email for a particular author.

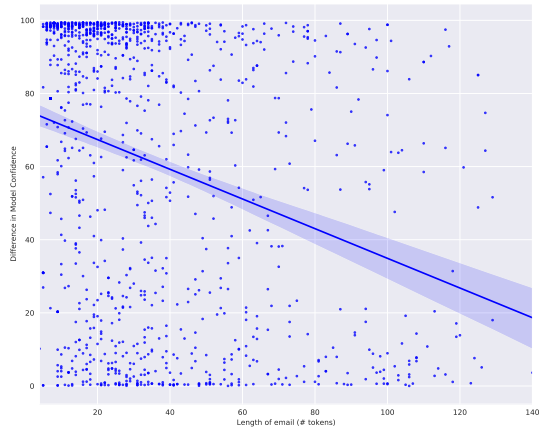

Figure 1: Model confidence difference of target author between unperturbed and signature removed emails vs email length

We can immediately see that there is a negligible change in the model's confidence of predicting the author "Kevin Presto", who seems to write longer emails and rarely signs them. In general, we can see that, both email length and email signatures seem to have an effect on the model's confidence in predicting the target author. We do note that there are certain anomalies. In particular, the confidence drop of the model is less in certain authors who have more signed emails. We think that this could be due to the fact that we are only concerned with email signatures and not authoritative style.

In order to better understand the relationship between length of email and the confidence difference, we plotted the confidence differences as a function of the email length, shown in figure 1. In this figure, the x-axis is the email length in number of tokens and the y-axis is the confidence difference of the model for all of the emails in the test set. From the fitted line we can see that there is a *downward trend* in the confidence difference. This reflects the fact that when emails are longer, the model's confidence in predicting the target label does not change significantly when predicting both unperturbed and signature-removed emails. This indicates that, for longer emails, the model does not rely on the signatures for prediction as much it does for short emails.

C. *Effect of Signature Removal*

While we have established that the models tend to rely heavily on email signatures for predicting the author, we were curious to see why this was happening. In order answer this question, we looked at the model architecture. In particular, BERT-based models are trained as language models and use a special token called the [CLS] token for classification. The [CLS] token is considered to be a representation of the input document [6]. For classifying the document, the [CLS] token is fed to a linear classifier to predict the class.

We plot the [CLS] vectors of the top four authors who had the highest confidence difference between their predictions with unperturbed and signature-perturbed emails. Figure 2, shows a plots of the [CLS] vectors of the input emails reduced to 2-dimensions using T-SNE [28].

Figure 2a, shows the visualization of the [CLS] vectors produced by the model that has been tested on unperturbed emails. We can see how the classes are *linearly separable*, indicating that a linear classifier will not have trouble in classifying them. Figure 2b, shows the [CLS] vectors produced by the successful perturbations of the model tested on signature-removed emails. Here, we see that the linear separability is completely lost. Due to this, the linear classifier sitting on top of the BERT architecture has trouble in correctly classifying these classes. Figure 2c, shows the [CLS] vectors produced by the *failed* perturbations of the model tested on signature-removed emails. We can see that despite certain outliers, linear separability is restored, thus enabling the linear classifier to correctly classify these data points, nullifying the perturbation. This indicates that signature perturbations to the input emails disrupts the linear separability of the [CLS] vector output by BERT models. Since these [CLS] vectors are used for classification, the classifier's performance goes down since it is unable to separate the classes easily.

Table III: Model Performance on Signature Perturbed Emails

	All Emails		Signed Emails	
Model Name	**Signature Removed**	**Signature Replaced**	**Signature Removed**	**Signature Replaced**
albert-base-v2	47.4 (-38.3)	51.1 (-34.6)	34.3 (-63.3)	40.4 (-57.2)
bert-base-uncased	55.4 (-32.6)	51.2 (-36.8)	44.1 (-54.0)	37.2 (-61.0)
bert-base-cased	51.2 (-36.6)	53.0 (-34.8)	37.9 (-60.5)	40.8 (-57.6)
deberta-base	51.2 (-37.2)	52.6 (-35.8)	36.8 (-61.5)	39.0 (-59.2)
distilbert-base-cased	48.4 (-38.7)	50.3 (-36.8)	34.4 (-64.2)	37.6 (-61.0)
distilbert-base-uncased	51.6 (-35.3)	47.5 (-39.4)	39.3 (-58.6)	32.7 (-35.2)
roberta-base	51.8 (-36.0)	52.2 (-35.5)	39.1 (-59.5)	39.7 (-58.8)
squeezebert-uncased	52.7 (-34.3)	50.1 (-36.9)	41.3 (-56.7)	37.0 (-61.1)

(a) (b) (c)

Figure 2: T-SNE Visualization of CLS Vector of BERT models: (a) Unperturbed email input; (b) Successful signature perturbation; (c) Unsuccessful signature perturbation

Table IV: Average drop in confidence by author in the test set by *bert-base-uncased*

Author Name	**Percentage Signed**	**Average # Tokens**	**Average Difference in Confidence**
Allen Phillip	55.81	71.27	30.25
Chris Dorland	66.50	41.37	47.68
Darron Giron	87.50	48.64	64.11
Jeffrey Shankman	38.03	45.25	42.79
Kevin Presto	3.87	110.14	-0.10
Kimberly Watson	76.96	39.49	35.37
Louise Kitchen	13.66	79.71	19.87
Lynn Blair	94.97	36.70	89.16
Mark Haedicke	79.63	26.38	79.57
Michelle Cash	82.11	56.78	47.64

VI. Conclusion & Future Work

Automatic email authorship classification is an important problem in various domains. In this paper, using a subset of the Enron email corpus, we have shown how certain machine learning models rely heavily on email signatures to predict the author of the email. By manipulating the signatures, we see that the model performance has plummeted by almost 60%. Furthermore, we have shown that this reliance on signatures is more acute in shorter emails. By inspecting the confidence differences between models tested on unperturbed and signature perturbed emails, we have shown that as the length of an email increases, the model gets more confident in predicting the target author. We have also shown why this is happening in transformer based models by analyzing the [CLS] vector that is used for classification. We see that signature perturbation disrupts the linear separability of the individual classes, thus making it hard for the linear classifier to correctly classify the author.

For future work we plan to look into: 1) more email datasets to see if the same trend ensues; 2) stylometric features that are important for author classification; 3) cloaking authors using adversarial stylometry [29]; 3) defense techniques such as adversarial training [30] to improve the robustness of authorship attribution models.

Acknowledgment

This manuscript has been in part co-authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy.

References

[1] Patrick Juola. “Authorship Attribution”. In: 1.3 (Dec. 2006), pp. 233–334. ISSN: 1554-0669. DOI: 10.1561/1500000005.

[2] Anderson Rocha, Walter J Scheirer, Christopher W Forstall, et al. “Authorship attribution for social media forensics”. In: *IEEE transactions on information forensics and security* 12.1 (2016), pp. 5–33.

[3] Sabine Ehrhardt and J Visconti. “Authorship attribution analysis”. In: *Handbook of Communication in the Legal Sphere* (2018), pp. 169–200.

[4] Patrick Juola. “Authorship attribution, constructed languages, and the psycholinguistics of individual variation”. In: *Digital Scholarship in the Humanities* 33.2 (2018), pp. 327–335.

[5] Saed Alrabaee, Mourad Debbabi, Paria Shirani, et al. “Authorship Attribution”. In: *Binary Code Fingerprinting for Cybersecurity*. Springer, 2020, pp. 211–230.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv e-prints*, arXiv:1810.04805 (Oct. 2018). eprint: 1810.04805.

[7] Klimt, Bryan and Yang, Yiming. “The Enron Corpus: A New Dataset for Email Classification Research”. In: *Machine Learning: ECML 2004*. Springer Berlin Heidelberg, 2004, pp. 217–226.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017, pp. 5998–6008.

[9] Efstathios Stamatatos. “A survey of modern authorship attribution methods”. In: *Journal of the American Society for information Science and Technology* 60.3 (2009), pp. 538–556.

[10] Sara El Manar El and Ismail Kassou. “Authorship analysis studies: A survey”. In: *International Journal of Computer Applications* 86.12 (2014).

[11] Mirco Kocher and Jacques Savoy. “Distributed language representation for authorship attribution”. In: *Digital Scholarship in the Humanities* 33.2 (2018), pp. 425–441.

[12] Prasha Shrestha, Sebastian Sierra, Fabio A González, et al. “Convolutional neural networks for authorship attribution of short texts”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. 2017, pp. 669–674.

[13] Fereshteh Jafariakinabad and Kien A Hua. “Style-aware neural model with application in authorship attribution”. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE. 2019, pp. 325–328.

[14] Georgios Barlas and Efstathios Stamatatos. “Cross-domain authorship attribution using pre-trained language models”. In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer. 2020, pp. 255–266.

[15] Yang LI, Wei ZHANG, and Chen PENG. “Target-dependent method for authorship attribution”. In: *Journal of Computer Applications* 40.2 (2019), pp. 473–478.

[16] Maël Fabien, Esa ú Villatoro-Tello, Petr Motlicek, et al. “BertAA: BERT fine-tuning for Authorship Attribution”. In: *2020 17th International Conference on Natural Language Processing (ICON)*. Dec. 2020.

[17] Edwin Dauber, Rebekah Overdorf, and Rachel Greenstadt. “Stylometric Authorship Attribution of Collaborative Documents”. In: *Cyber Security Cryptography and Machine Learning*. 2017, pp. 115–135.

[18] Rachel Greenstadt. “Using Stylometry to Attribute Programmers and Writers”. In: *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. 2017, p. 91.

[19] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, et al. “Adversarial attacks on deep-learning models in natural language processing: A survey”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.3 (2020), pp. 1–41.

[20] J Li, S Ji, T Du, et al. “TextBugger: Generating Adversarial Text Against Real-world Applications”. In: *26th Annual Network and Distributed System Security Symposium*. 2019.

[21] Di Jin, Zhijing Jin, Joey Tianyi Zhou, et al. “Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment”. In: *arXiv e-prints*, arXiv:1907.11932 (July 2019). eprint: 1907.11932.

[22] Ji Gao, Jack Lanchantin, Mary Lou Soffa, et al. “Black-box generation of adversarial text sequences to evade deep learning classifiers”. In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2018, pp. 50–56.

[23] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, et al. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: *arXiv e-prints*, arXiv:1909.11942 (Sept. 2019). eprint: 1909.11942.

[24] Pengcheng He, Xiaodong Liu, Jianfeng Gao, et al. “DeBERTa: Decoding-enhanced BERT with Disentangled Attention”. In: *arXiv e-prints*, arXiv:2006.03654 (June 2020). eprint: 2006.03654.

[25] Victor Sanh, Lysandre Debut, Julien Chaumond, et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *arXiv e-prints*, arXiv:1910.01108 (Oct. 2019). eprint: 1910.01108.

[26] Yinhan Liu, Myle Ott, Naman Goyal, et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv e-prints*, arXiv:1907.11692 (July 2019). eprint: 1907.11692.

[27] Forrest N. Iandola, Albert E. Shaw, Ravi Krishna, et al. “SqueezeBERT: What can computer vision teach NLP about efficient neural networks?” In: *arXiv e-prints*, arXiv:2006.11316 (June 2020). eprint: 2006.11316.

[28] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

[29] Michael Brennan, Sadia Afroz, and Rachel Greenstadt. “Adversarial stylometry: Circumventing author-

ship recognition to preserve privacy and anonymity". In: *ACM Transactions on Information and System Security (TISSEC)* 15.3 (2012), pp. 1–22.

[30] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, et al. "Adversarial Training for Free!" In: *arXiv e-prints*, arXiv:1904.12843 (Apr. 2019). eprint: 1904.12843.