Advancing the Use of Information Compression Distances in Authorship Attribution

Santiago Palmero Muñoz[1], Christian Oliva[2(✉)], Luis F. Lago-Fernández[2], and David Arroyo[1]

[1] Institute for Physical and Information Technologies "Leonardo Torres Quevedo" (ITEFI), Consejo Superior de Investigaciones Científicas (CSIC), Madrid, Spain
david.arroyo@csic.es
[2] Universidad Autónoma de Madrid, 28049 Madrid, Spain
{christian.oliva,luis.lago}@uam.es

Abstract. Detecting unreliable information in social media is an open challenge, in part as a result of the difficulty to associate a piece of information to known and trustworthy actors. The identification of the origin of sources can help society deal with unverified, incomplete, or even false information. In this work we tackle the problem of associating a piece of information to a certain politician. The use of inaccurate information is of great relevance in the case of politicians, since it affects social perception and voting behavior. Moreover, misquotation can be weaponized to hinder adversary reputation. We consider the task of applying a compression-based metric to conduct authorship attribution in social media, namely in Twitter. In specific, we leverage the Normalized Compression Distance (NCD) to compare an author's text with other authors' texts. We show that this methodology performs well, obtaining 80.3% accuracy in a scenario with 6 different politicians.

Keywords: Authorship recognition · Cyber-attribution · Normalised compression distance

1 Introduction

Communication through social media has become an essential part of people's lives. However, widespread misinformation and disinformation have become serious risks. Detecting unreliable information is a crucial challenge, especially when the actors behind information sources are unknown [29]. Inaccurate and fabricated content in social media comes from a variety of sources, usually as user-generated content or information scraped from the Internet and manually modified [17]. Attributing authorship for this type of information can help track the related sources and find their origin.

The proliferation of non-credible information is conspicuously hazardous in the case of politics. In the past decades, there is a significant number of examples

F. Spezzano et al. (Eds.): MISDOOM 2022, LNCS 13545, pp. 114–122, 2022.
https://doi.org/10.1007/978-3-031-18253-2_8

in which some politicians make use of ungenuine news and information to gain political advantage [13]. The capability of tracking information sources can be partially constructed by means of authorship attribution [16]. Our work will contribute to such a goal by tackling authorship attribution in Twitter for a given dataset which contains tweets from six US (United States) politicians. We propose a method to extract features from texts by comparing tweets using the Normalized Compression Distance (NCD) [8]. A K-dimensional space is constructed by selecting representatives or generators of each writing style, so that each new text is represented in this space by considering its NCD with respect to each generator. According to this representation, it is possible to train a classifier to conclude about the authorship of a text. In this work we consider the following Machine Learning (ML) models: Support Vector Machines (SVMs) and Multilayer Perceptrons (MLPs).

The remaining of the article is organized as follows. First, in Sect. 2 we discuss the state of the art about the application of NCD to Authorship Attribution in social media. In Sect. 3, we introduce the dataset used to validate our method. Then, in Sect. 4, we detail the NCD-based features construction. In Sect. 5, we briefly explain the ML used in our experiments. The results are discussed in Sect. 6, and the derived conclusions and highlights for future work are provided in Sect. 7.

2 Related Works

Research in authorship attribution has increased in recent years [2,16,24]. Some of the works propose the use of multivariate analysis in stylometry. Algorithms generate some vectors of frequencies which are then classified by clustering models [4–6,15]. Others authors introduce some labeled data to improve the obtained results with different ML algorithms: from traditional models, such as K-Nearest Neighbors [14,19,27], or Support Vector Machines [10,24], to Deep Learning algorithms for Natural Language Processing, such as LSTMs [21,28] or Convolutional Neural Networks [3,30].

Concerning Authorship Attribution in short texts from social media, the complexity of classifying the texts increases significantly. Some models like LSTMs have their performance heavily affected when dealing with this specific task [28], leaving the door open to the SVMs and CNNs. Regarding datasets, Twitter serves as a great benchmark as it allows to obtain a large pool of users and tweets from similar or different domains. Hence, most research focuses on this social media [1,3,11,20,24,30].

Within the techniques involving Authorship Attribution, compression distances have been used with remarkable results. There are many distance-based metrics, such as the Conditional Complexity of Compression (CCC) [22], the Normalized Compression Distance (NCD) [8], or the Compression Dissimilarity Measure (CDM) [33], and also a large variety of compressors, such as PPMd, Gzip, BZip2, Zip, or LZW [12]. The most remarkable compression methods are profile-based, i.e., those that concatenate all available text from a known author

and then compare an unseen text with this [33]. Other works use instance-based methods, i.e., they estimate the distance to all the available text and then group these distances using clustering methods.

3 Politicians Dataset

In this work, we use a dataset containing tweets from different politicians. We base our preprocessing on [24]. We remove those tweets marked as retweet by the metadata and tweets following the old retweet convention, which included the characters RT. We also remove non-English tweets and those with less than four tokens. Finally, tweet tags, which include usernames and hashtags, urls, numbers, dates and timestamps are replaced with the tokens REF, TAG, URL, NUM, DAT, TIM, respectively [24,26]. These replacements, especially tweet tags, are needed to avoid creating models unsuitable for authorship attribution [20].

The dataset contains approximately 1.25×10^6 tweets from 545 US politicians[1]. From the original dataset, we choose the six users who have the highest number of tweets. To break ties in the top six, we use alphabetical order. After preprocessing, the final dataset has a total of 16000 instances. For the evaluation, we generate five partitions by splitting the dataset in 80% training - 20% test. This dataset is publicly available, which facilitates reproducibility for further works and its users have not been selected by Twitter search heuristics.

4 Feature Construction: NCD Attribute Vectors

The Normalized Compression Distance (NCD) [8] is a compresion-based metric that calculates the similarity between two texts by means of a distance. Thus, two texts are similar when the distance between them is small, and they are different when their distance is large. We describe the process of generating attributes using this metric.

From text strings T, we create a set of K attribute generators $G = \{g_1, g_2, ..., g_K\}$. Each generator g_i is the concatenation of some strings from T, which are not used in the rest of the generators in G. These generators are not balanced regarding size and the number of strings in each of them can vary. However, G contains an even number of generators of each class that equals $K/6$. The procedure used in our experiments transforms text strings from T into numerical attributes by using a variant of the NCD. This variant, called the normalized conditional compressed information distance [32], is defined as:

$$D(g_j, t_i) = \frac{C(g_j :: t_i) - C(g_j)}{C(t_i)}, \tag{1}$$

where g_j is a generator and t_i is a single text string from T. $C(x)$ is the *gzip* compressed size of x, and the operator :: is the concatenation of strings. Then,

[1] The original dataset can be downloaded from https://www.reddit.com/r/datasets/comments/6fniik/over_one_million_tweets_collected_from_us/.

text strings from T have its distance computed to each of the generators in G forming a new set of data I. Each instance in I is an attribute vector with dimension K where each element is $i_{ij} = D(g_j, t_i)$. Finally, I is used to train a ML model. Generators G can be created not only from text strings within the original dataset T, but also using other strings from external sources. The rest of this section explains the previous [31,32] and novel procedures we have conducted to generate subsets G and I under a multi-class authorship scenario.

4.1 Disjoint Subsets

Initial set T is divided into T_{tr} and T_{test}. Then, T_{tr} is divided again into T_G and T_I. Subset G is created from T_G, and I_{tr} and I_{test} are created following the functions $D(G, T_I)$ and $D(G, T_{test})$, respectively. This way, strings that are used to create the generators G do not appear in I. Thus, G and I are disjoint subsets. For all the experiments done with this procedure we used 80% of the instances from T_{tr} to create T_G.

4.2 All Data for Training

Following the idea in [32], the Disjoint approach might leave a small subset of data for training, which could be a handicap. In this procedure, the initial set T is only divided into T_{tr} and T_{test}. Subset G is created from T_{tr}. I_{tr} and I_{test} are created following the functions $D(G, T_{tr})$ and $D(G, T_{test})$, respectively. Note that this time G and I_{tr} share T_{tr}. Every string from T_{tr} contributes to the creation of one generator. Consequently, each instance of I_{tr} has an attribute which is close to 0, because of the distance to this generator, adding some bias.

5 ML Models and Evaluation

In this section, we describe the Machine Learning (ML) models used in this work: Support Vector Machines (SVM) [9], which are one of the most robust prediction models, and Multilayer Perceptrons (MLP) [25], the most common feedforward neural network. The NCD attribute vectors will be the input to these models. We use the standard Scikit-Learn [23] implementation for SVM, and the Keras [7] implementation for MLP. A description of the evaluation is included.

5.1 Machine Learning Models

Support Vector Machine. This ML approach bases its classification on finding a hyperplane (a decision boundary) that separates the classes with maximum margin [9]. It uses kernel functions to map a non-linearly separable N-dimensional dataset onto a new high dimensional space in which linear separability is more plausible. In this work, we use linear and radial basis function (RBF) kernels.

Multilayer Perceptron. This model is the most common kind of artificial neural network, where neurons are hierarchically organized in layers, with feed-forward connections between adjacent layers. Hidden layers (neither input nor output) provide the computational processing to determine the most probable class. In this work, we use a Rectifier neural network with a single hidden layer, which is an MLP with Rectifier Linear Unit (ReLU) activation function.

5.2 Evaluation

The models' evaluation consists of a cross-validation with the five partitions described in Sect. 3. We consider the differences between the attribute vectors generation procedures described in Sect. 4. For tuning the optimal hyperparameters, we follow the grid search detailed in Table 1. We train both SVMs (linear and RBF kernels) until convergence, and the MLP for 500 epochs. To minimize the cross-entropy loss we use the Adam optimizer [18], applying checkpointing to get the best validation loss. We also add dropout and L2 regularization to every layer.

Table 1. Grid search tuning for training linear SVMs, RBF SVMs, and MLPs

Model	Hyperparam	Values
Linear SVM	C	$range[10^{-4}{-}10^{4}]$
RBF SVM	C	$range[10^{-4}{-}10^{4}]$
	γ	$range[10^{-4}{-}10^{4}]$
MLP	Units	[100, 200]
	Learning rate	$range[10^{-4}{-}10^{-1}]$
	L2 regularization	[0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]
	Dropout	[0, 0.1, 0.2, 0.3]
	Epochs	500

6 Results

In this section, we show our results after following the procedures described in Sect. 4 with the models and evaluation criteria detailed in Sect. 5. In all our analyses, we use accuracy (ACC) and balanced accuracy (BAC) to compare the performance of our proposals. We train SVM with Linear and RBF kernels and MLP, applying the grid search detailed in Table 1. The three methods have been tested with the K values 18, 36, 72, and 144. We show in Fig. 1 the test accuracy for each model with the best hyperparameter settings and K value, and in Tables 2 and 3, we show more detailed results for both procedures.

There are some ideas to extract from these results. First, there is a slight increase in the performance of the MLP model against the SVM in the two approaches. In addition, regarding the procedure of the features generation, the

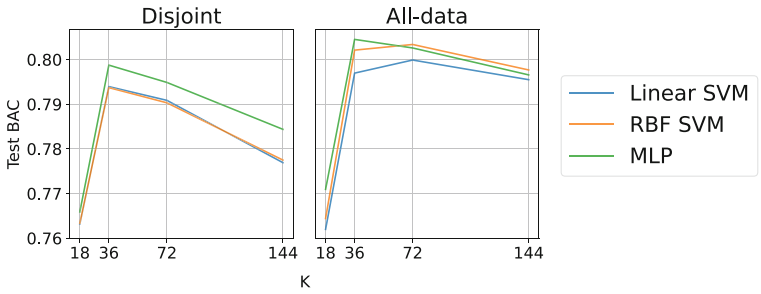

Fig. 1. Test Balanced Accuracy (BAC) versus K value for procedures *Disjoint* and *All-data* with the three models.

Table 2. Best results obtained with the *Disjoint* approach for all the models

K	Model	Hyperparams	AccTest	BACTest
36	*Linear SVM*	$C = 1$	0.798 ± 0.005	0.794 ± 0.006
36	*RBF SVM*	$C = 10$ $\gamma = 0.1$	0.798 ± 0.005	0.794 ± 0.004
36	*MLP*	$LR = 0.001$ $L2 = 0$ $Dropout = 0.1$	$\mathbf{0.803 \pm 0.004}$	$\mathbf{0.799 \pm 0.003}$

Table 3. Best results obtained with the *All-data* approach for all the models

K	Model	Hyperparams	AccTest	BACTest
72	*Linear SVM*	$C = 1$	0.804 ± 0.007	0.800 ± 0.008
72	*RBF SVM*	$C = 1$ $\gamma = 10$	0.806 ± 0.007	0.803 ± 0.006
36	*MLP*	$LR = 0.001$ $L2 = 0.001$ $Dropout = 0.1$	$\mathbf{0.809 \pm 0.007}$	$\mathbf{0.805 \pm 0.006}$

best option is All-data. However, it is worth to mention that SVMs require a much lower computational cost for training than the neural network. The MLP's need for the hyperparameters search described in Table 1 makes the SVM choice more suitable.

Concerning the K values, both $K = 36$ and $K = 72$ show the best performance. The value of K is directly related to the size of the generators, so it should not make their size larger than 32 KB [22,31]. The reason for this is that the sliding window of the *LZ77* algorithm, used in *gzip*, can only reference the last 32 KB[2]. Therefore, the point is to adjust the K value to be as higher as

[2] For more information visit https://datatracker.ietf.org/doc/html/rfc1951.

Table 4. Relation between the number of generators K and the size of each generator for the *Disjoint* procedure (left), and the *All-data* procedure (right).

Disjoint		All-data	
K	**size (bytes)**	**K**	**size (bytes)**
18	31948.37±32.42	18	31945.34±33.43
36	28505.17±3225.63	36	31378.09±1182.62
72	14513.80±2015.85	72	18169.24±2491.86
144	7231.36±1004.27	144	9057.20±1245.30

possible while making generators of size as close as possible to 32 KB. We show in Table 4 the relation between the number of generators K and the size of each generator for both the *Disjoint* (left) and *All-data* (right) procedures.

Looking at the tables, considering the size of the generators, the best values of K are 18 and 36. However, with $K = 18$ the models do not have enough information to perform well, and this is observed in the accuracy (see Fig. 1). Finally, increasing the number of generators beyond 36 reduces their size, and this also affects the models' accuracy.

7 Conclusions and Future Work

In this paper we have proposed a method to conduct authorship attribution by combining compression metrics and ML. The method has been tested on a dataset with tweets from six US politicians. We have analyzed the possibility of concluding about the politician behind a certain tweet just by measuring the NCD of this tweet with respect to a set of K writing style representatives. Such a comparison enables the classification of the tweet on the ground of ML models properly trained using a K-dimensional space of representation. This space is constructed upon K representatives or generators extracted from the original dataset. As ML models, we have used SVMs and MLPs. In our experiments we have evaluated the selection of adequate values of K, and the possibility of using as much data as possible to train and validate our models. Indeed, we distinguish two scenarios with regard to data preparation. First, we consider that none of the text samples included in the generators set are included in the text samples used to train and validate our model. Second, we consider the use of samples from the generators set as samples in the training and validation sets. In other words, we enable data reutilization. We reach the conclusion that the best option for our NCD-based authorship attribution is to use MLP, consider a $K = 36$ dimensional representation space and to re-use generators data for training and validation.

For future work, we have to bear in mind the explainability shortcomings of MLP, along with its computational burden. Our next steps in NCD-based authorship attribution will target at improving the construction of the training dataset and study the possibility of replacing MLP by SVM. Moreover, we have

to take into account that in this paper we have used an implementation of the NCD using *gzip*. Additional work is required to consider alternatives to this compressor, which eventually could lead to overcome the limitations associated to the sliding window of the *LZ77* algorithm (e.g., ppmz or bzip2).

Acknowledgements. This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 872855 (TRESCA project), from Grant PLEC2021-007681 (project XAI-DisInfodemics) funded by MCIN/AEI/ 10.13039/501100011033 and by European Union NextGeneration EU/PRTR, from Comunidad de Madrid (Spain) under the project CYNAMON (no. P2018/TCS-4566), cofunded with FSE and FEDER EU funds, and from Spanish projects MINECO/FEDER TIN2017-84452-R and PID2020-114867RB-I00 (http://www.mineco.gob.es/).

References

1. Alonso-Fernandez, F., Belvisi, N.M.S., Hernandez-Diaz, K., Muhammad, N., Bigun, J.: Writer identification using microblogging texts for social media forensics. IEEE Trans. Biomet. Behav. Identity Sci. **3**(3), 405–426 (2021)
2. Aykent, S., Dozier, G.: AARef: exploiting authorship identifiers of micro-messages with refinement blocks. In: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1044–1050. IEEE (2020)
3. Aykent, S., Dozier, G.: Author identification of micro-messages via multi-channel convolutional neural networks. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 675–681. IEEE (2020)
4. Baayen, H., Halteren, H., Neijt, A., Tweedie, F.: An experiment in authorship attribution, January 2002
5. Binongo, J.N.G.: Who wrote the 15th book of OZ? An application of multivariate analysis to authorship attribution. Chance **16**(2), 9–17 (2003)
6. Burrows, J.F.: Word-patterns and story-shapes: the statistical analysis of narrative style. Liter. Linguist. Comput. **2**(2), 61–70 (1987)
7. Chollet, F., et al.: Keras. http://keras.io (2015)
8. Cilibrasi, R., Vitanyi, P.: Clustering by compression. IEEE Trans. Inf. Theory **51**(4), 1523–1545 (2005)
9. Cortes, C., Vapnik, V.: Support vector networks. Mach. Learn. **20**, 273–297 (1995)
10. Diederich, J., Kindermann, J., Leopold, E., Paass, G.: Authorship attribution with support vector machines. Appl. Intell. **19**, 109–123 (2003). https://doi.org/10.1023/A:1023824908771
11. Fourkioti, O., Symeonidis, S., Arampatzis, A.: Language models and fusion for authorship attribution. Inf. Process. Manag. **56**(6), 102061 (2019)
12. Halvani, O., Winter, C., Graner, L.: On the usefulness of compression models for authorship verification. In: Proceedings of the 12th International Conference on Availability, Reliability and Security, pp. 1–10 (2017)
13. Hameleers, M., Minihold, S.: Constructing discourses on (un)truthfulness: attributions of reality, misinformation, and disinformation by politicians in a comparative social media setting. Commun. Res. (2020)
14. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, 2nd edn. Inference and Prediction. Springer, New York (2009). https://doi.org/10.1007/978-0-387-21606-5

15. Holmes, D., Robertson, M., Paez, R.: Stephen crane and the New York tribune: a case study in traditional and non-traditional authorship attribution. Comput. Human. **35**, 315–331 (2001)
16. IARPA: Human Interpretable Attribution of Text using Underlying Structure (HIATUS) Program (2022)
17. Jursenas, A., Karlauskas, K., Ledinauskas, E., Maskeliunas, G., Rondomanskas, D., Ruseckas, J.: The Role of AI in the Battle Against Disinformation (2022)
18. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
19. Kjell, B., Addison Woods, W., Frieder, O.: Information retrieval using letter tuples with neural network and nearest neighbor classifiers. In: 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century. vol. 2, pp. 1222–1226 (1995)
20. Layton, R., Watters, P., Dazeley, R.: Authorship attribution for twitter in 140 characters or less. In: 2010 Second Cybercrime and Trustworthy Computing Workshop, pp. 1–8. IEEE (2010)
21. Oliva, C., Palmero-Muñoz, S., Lago-Fernández, L.F., Arroyo, D.: Improving LSTMs' under-performance in authorship attribution for short texts. In: Proceedings of the European Interdisciplinary Cybersecurity Conference (EICC) (2022)
22. Oliveira, W., Jr., Justino, E., Oliveira, L.S.: Comparing compression models for authorship attribution. Forensic Sci. Int. **228**(1–3), 100–104 (2013)
23. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
24. Rocha, A., et al.: Authorship attribution for social media forensics. IEEE Trans. Inf. Forensics Secur. **12**(1), 5–33 (2017)
25. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation, pp. 318–362. MIT Press, Cambridge, MA, USA (1986)
26. Schwartz, R., Tsur, O., Rappoport, A., Koppel, M.: Authorship attribution of micro-messages. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1880–1891 (2013)
27. Selj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the Conference Pacific Association for Computational Linguistics PACLING 2003, September 2003
28. Shrestha, P., Sierra, S., González, F.A., Montes, M., Rosso, P., Solorio, T.: 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, vol. 2, Short Papers, pp. 669–674 (2017)
29. Theophilo, A., Giot, R., Rocha, A.: Authorship attribution of social media messages. IEEE Trans. Comput. Soc. Syst. 1–14 (2021)
30. Theóphilo, A., Pereira, L.A., Rocha, A.: A needle in a haystack? Harnessing onomatopoeia and user-specific stylometrics for authorship attribution of micro-messages. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2692–2696. IEEE (2019)
31. de la Torre-Abaitua, G., Lago-Fernández, L.F., Arroyo, D.: A compression-based method for detecting anomalies in textual data. Entropy **23**(5), 618 (2021)
32. de la Torre-Abaitua, G., Lago-Fernández, L.F., Arroyo, D.: On the application of compression-based metrics to identifying anomalous behaviour in web traffic. Log. J. IGPL **28**(4), 546–557 (2020)
33. Veenman, C.J., Li, Z.: Authorship verification with compression features. In: CLEF (Working Notes) (2013)