Cross-Domain Authorship Attribution Using Pre-trained Language Models

Georgios Barlas(✉) and Efstathios Stamatatos

University of the Aegean, 83200 Karlovassi, Greece
barlasgeorgios@gmail.com, stamatatos@aegean.gr

Abstract. Authorship attribution attempts to identify the authors behind texts and has important applications mainly in cyber-security, digital humanities and social media analytics. An especially challenging but very realistic scenario is cross-domain attribution where texts of known authorship (training set) differ from texts of disputed authorship (test set) in topic or genre. In this paper, we modify a successful authorship verification approach based on a multi-headed neural network language model and combine it with pre-trained language models. Based on experiments on a controlled corpus covering several text genres where topic and genre is specifically controlled, we demonstrate that the proposed approach achieves very promising results. We also demonstrate the crucial effect of the normalization corpus in cross-domain attribution.

Keywords: Authorship Attribution · Neural network language models · Pre-trained language models

1 Introduction

Authorship Attribution (AA) is a very active area of research dealing with the identification of persons who wrote specific texts [12,20]. Typically, there is a list of suspects and a number of texts of known authorship by each suspect and the task is to assign texts of disputed authorship to one of the suspects. The basic forms of AA are closed-set attribution (where the list of suspects necessarily includes the true author), open-set attribution (where the true author could be excluded from the list of suspects), and author verification (where there is only one candidate author). The main applications of this technology are in digital forensics, cyber-security, digital humanities, and social media analytics [8,15].

In real life scenarios the known and the unknown texts may not share the same properties. The topic of the texts may differ but also the genre (e.g., essay, email, chat). Cross-domain AA examines those cases where the texts of known authorship (training set) differ with respect to the texts of unknown authorship (test set) in topic (cross-topic AA) or in genre (cross-genre AA) [19,22]. The main challenge here is to avoid the use of information related to topic or genre of

Published by Springer Nature Switzerland AG 2020
I. Maglogiannis et al. (Eds.): AIAI 2020, IFIP AICT 583, pp. 255–266, 2020.
https://doi.org/10.1007/978-3-030-49161-1_22

documents and focus only on stylistic properties of texts related to the personal style of authors.

Recently, the use of pre-trained language models (e.g., BERT, ELMo, ULMFiT, GPT-2) has been demonstrated to obtain significant gains in several text classification tasks including sentiment analysis, emotion classification, and topic classification [2,7,13,14]. However, it is not yet clear whether they can be equally useful for style-based text categorization tasks. Especially, in cross-topic AA, information about the topic of texts can be misleading.

An approach based on neural network language models achieved top performance in recent shared tasks on authorship verification and authorship clustering (i.e., grouping documents by authorship) [16,23]. This method is based on a character-level recurrent (RNN) neural network language model and a multi-headed classifier (MHC) [1]. So far, this model has not been tested in closed-set attribution which is the most popular scenario in relevant literature. In this paper, we adopt this approach for the task of closed-set AA and more specifically the challenging cases of cross-topic and cross-genre AA.

We examine the use of pre-trained language models (e.g., BERT, ELMo, ULMFiT, GPT-2) in AA and the potentials of MHC. We also demonstrate that in cross-domain AA conditions, the effect of an appropriate normalization corpus is crucial.

2 Previous Work

The vast majority of previous work in AA focus on the closed-set attribution scenario. The main issues is to define appropriate stylometric measures to quantify the personal style of authors and the use of effective classification methods [12,20].

A relatively small number of previous studies examine the case of cross-topic AA. In early approaches, features like function words or part-of-speech n-grams have been suggested as less likely to correlate with topic of documents [10,11]. However, one main finding of several studies is that low-level features, like character n-grams, can be quite effective in this challenging task [19,21]. Typed character n-grams provide a means for focusing on specific aspects of texts [17]. Interestingly, character n-grams associated with word affixes and punctuation marks seem to be the most useful ones in cross-topic AA. Another interesting idea is to apply structural correspondence learning using punctuation-based character n-gram as pivot features [18]. Recently, a text distortion method has been proposed as a pre-processing step to mask topic-related information in documents while keeping the text structure (i.e., use of function words and punctuation marks) intact [22].

There have been attempts to use language modeling for AA including traditional n-gram based models as well as neural network-based models [1,4,5]. The latter is closely related to representation learning approaches that use deep learning methods to generate distributed text representations [3,9]. In all these

cases, the language models are extracted from the texts of known authorship. As a result, they heavily depend on the size of the training set per candidate author.

3 The Proposed Method

An AA task can be expressed as a tuple (A, K, U) where A is the set of candidate authors (suspects), K is the set of known authorship documents (for each $a \in A$ there is a $K_a \subset K$) and U is the set of unknown authorship documents. In closed-set AA, each $d \in U$ should be attributed to exactly one $a \in A$. In cross-topic AA, the topic of documents in U is distinct with respect to the topics found in K, while in cross-genre AA, the genre of documents in U is distinct with respect to the genres found in K.

Bagnall introduced an AA method[1] [1] and obtained top positions in shared tasks in authorship verification and authorship clustering [16,23]. The main idea is that a character-level RNN is produced using all available texts by the candidate authors while a separate output is built for each author (MHC). Thus, the recurrent layer models the language as a whole while each output of MHC focuses on the texts of a particular candidate author. To reduce the vocabulary size, a simple pre-processing step is performed (i.e., uppercase letters are transformed to lowercase plus a symbol, punctuation marks and digits are replaced by specific symbols) [1].

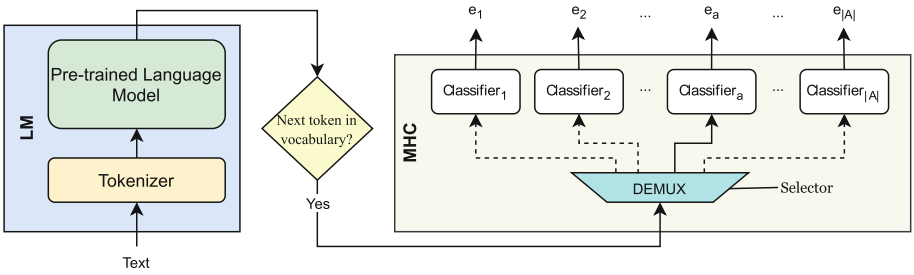


Fig. 1. The proposed model consists of two parts, the language model (LM) and the multi-headed classifier (MHC). The DEMUX layer in MHC part functions as a demultiplexer, its state is defined by the selector. During training phase the selector is defined by the author of the input text and during calculation of normalization vector or test phase the input of DEMUX is connected to all its outputs.

The model, as shown in Fig. 1, consists of two parts, LM and MHC. LM consists of a tokenization layer and the pre-trained language model. MHC comprises a demultiplexer which helps to select the desirable classifier and a set of $|A|$ classifiers, where $|A|$ is the number of candidate authors. Each classifier

[1] https://github.com/pan-webis-de/caravel.

has N inputs, where N is the dimensionality of the LM's representation, and V outputs, where V is the size of the vocabulary. The vocabulary is created using the most frequent tokens. The output of LM is a representation of each token in text. If the token exists in vocabulary its representation propagates to MHC, otherwise is ignored (despite the fact that the representation is not further useful, the calculations that took place in LM to produce the representation are mandatory to update the hidden states of the pre-trained language model). If the sequence of input tokens is modified, the representation is also affected.

The function of LM remains the same during training, calculation of normalization vector n and test phase. The MHC layer during training propagates the LM's representations only to the classifier of the author a which is the author of the given text. Then the cross-entropy error is back-propagated to train MHC. During the test phase (as well as the calculation of normalization vector n explained below) the LM's representation is propagated to all classifiers.

The MHC calculates the cross-entropy $H(d, K_a)$ for each input text d and the training texts of each candidate author K_a. The lower cross-entropy is, the more likely for author a to write document d. However, the scores obtained for different candidate authors are not directly comparable due to different bias at each head of MHC. To handle this problem, a normalization vector n is used which is equal to zero-centered relative entropies produced by using an unlabeled normalization corpus C [1]:

$$n = \frac{1}{|C|} \sum_{d_i \in C} H(d_i, K_a) \tag{1}$$

where $|C|$ is the size of the normalization corpus. Note that in cross-domain conditions it is very important for documents in C to include documents belonging to the domain of d. Then, the most likely author a for a document $d \in U$ is found using the following criterion:

$$\arg\min_a \left(H(d, K_a) - n\right) \tag{2}$$

In this paper, we extended Bagnall's model in order to accept tokens as input and we propose the use of a pre-trained language model to replace RNN in the aforementioned AA method. The RNN proposed by Bagnall [1] is trained using a small set of documents (K for closed-set AA). In contrast, pre-trained language models have been trained using millions of documents in the same language. Moreover, RNN is a character-level model while the pre-trained models used in this study are token-level approaches. More, specifically, the following models are considered:

- *Universal Language Model Fine-Tuning* (ULMFiT): It provides a contextual token representation obtained from a general domain corpus of millions of unlabeled documents [7]. It adopts left-to-right and right-to left language modeling in separate networks and follows auto-encoder objectives.
- *Embeddings from Language Models* (ELMo): It extracts context-sensitive features using a left-to-right and a right-to-left language modeling [13]. Then,

the representation of each token is a linear combination of the representation of each layer.
- *Generative Pretrained Transformer 2* (GPT-2): It is based on a multi-layer unidirectional *Transformer* decoder [24]. It applies a multi-headed self-attention operation over the input tokens followed by position-wise feed-forward layers [14].
- *Bidirectional Encoder Representations from Transformer* (BERT): It is based on a bidirectional *Transformer* architecture that can better exploit contextual information [2]. It masks a percentage of randomly-selected tokens which the language model is trained to predict.

4 Experiments

4.1 Corpus

We use the CMCC corpus introduced in [6] and also used in previous cross-domain AA works [19,22]. CMCC is a controlled corpus in terms of genre, topic and demographics of subjects. It includes samples by 21 undergraduate students as candidates authors (A), covering six genres (blog, email, essay, chat, discussion, and interview) and six topics (catholic church, gay marriage, privacy rights, legalization of marijuana, war in Iraq, gender discrimination) in English. To ensure that the same specific aspect of the topic is followed, a short question was given to subjects (e.g., Do you think the Catholic Church needs to change its ways to adapt to life in the 21th Century?). In two genres (discussion and interview) the samples were audio recordings and they have been transcribed into text as accurately as possible maintaining information about pauses, laughs etc. For each subject, there is exactly one sample for each combination of genre and topic. More details about the construction of this corpus are provided in [6].

4.2 Experimental Setup

In this study, our focus is on cross-topic and cross-genre AA. In cross-topic, we assume that the topic of training texts (K) is different from the topic of test texts (U) while all texts (both K and U) belong in the same genre. Similar to [22] and [19], we perform leave-one-topic-out cross-validation where all texts on a specific topic (within a certain genre) are included in the test corpus and all remaining texts on the remaining topics (in that genre) are included in the training corpus. This is repeated six times so that all available topics to serve exactly once as the test topic. Mean classification accuracy over all topics is reported.

Similar to cross-topic, in cross-genre we perform leave-one-genre-out cross-validation as in [22], where all texts on a specific genre (within a certain topic) are included in the test corpus and all remaining texts on the remaining genres (in that topic) are included in the training corpus. The number of available genres is also six like topics, and though we repeat the leave-one-genre-out cross-validation

Table 1. Dimensionality of representation (N) for each language model in this study.

Model	RNN	BERT	ELMo	GPT-2	ULMFiT
N	149	768	1024	768	400

six times and report the mean classification accuracy. In both scenarios, cross-topic and cross-genre, the candidates authors set A consists of 21 undergraduate students as mentioned in Sect. 4.1.

All the examined models use a MHC on top of a language modeling method. First, we study the original Bagnall's approach where a character-level RNN is trained over K. Then, each one of the pre-trained language models described in previous section. In our experiments, all of the pre-trained LMs was fine-tuned for the specific AA task with MHC as classifier without further training the language model , since our goal is to explore the potential of pre-trained models obtained from general domain corpora.

In MHC, each author corresponds to a separate classifier with N inputs and M outputs, where N is the dimensionality of text representation, Table 1, and M is equal to vocabulary size V. During training, each classification layer is trained only with the documents of the corresponding author. The vocabulary is defined as the most frequent tokens in the corpus. These are less likely to be affected by topic shifts and the reduced input size increases the efficiency of our approach. The selected values of V are 100, 500, $1k$, $2k$ and $5k$. Each model used its own tokenization stage except from ELMo (where ULMFiT's tokenization was used). Note that RNN is a character-level model while all pre-trained models are token-based.

Since RNN is trained from scratch for a corpus of small size, it is considerably affected by initialization. As a result, there is significant variance when it is applied several times to the same corpus. To compensate this, we report average performance results for 10 repetitions. Regarding the training phase of each method, we use 100 epochs for RNN and examine four cases for the pre-trained models: the minimal training of 1 epoch and the cases of 5, 10 and 20 epochs of training.

4.3 Results on Cross-Topic AA

Table 2 presents the leave-one-topic-out cross-validation accuracy results for each one of the six available genres as well as the average performance over all genres for each method. Two cases are examined: one using the (unlabeled) training texts as normalization corpus ($C = K$) and another where the (unlabeled) test texts are used as normalization corpus ($C = U$). The former means that C includes documents with distinct topics with respect to the document of unknown authorship while the latter ensures that there is perfect thematic similarity. As can be seen, the use of a suitable normalization corpus is crucial to enhance the performance of the examined methods.

Table 2. Accuracy results (%) on Cross-Topic AA. The reported performance of baseline models is taken from the corresponding publications.

‡	LM	V	epochs	Blog	Email	Essay	Chat	Disc.	Interv.	Avg.
$C = K$	RNN	–	100	47.94	44.37	41.00	73.41	75.71	72.54	59.16
	BERT	$2k$	1	57.14	49.21	60.32	84.92	79.37	80.16	68.52
	BERT	$5k$	1	53.97	52.38	58.73	86.51	77.78	78.57	67.99
	ELMo	$2k$	1	56.35	55.56	56.35	80.95	72.22	76.98	66.40
	ELMo	$5k$	1	55.56	53.17	57.14	82.54	70.64	76.19	65.87
	GPT-2	$2k$	20	60.32	57.94	54.76	76.98	63.49	79.37	65.48
	GPT-2	$5k$	20	58.73	59.52	61.11	84.13	63.49	76.98	67.33
	ULMFiT	$2k$	10	50.00	43.65	52.38	79.37	72.22	71.43	61.51
	ULMFiT	$5k$	20	46.83	40.48	50.79	80.16	69.84	70.64	59.79
$C = U$	RNN	–	100	61.67	56.43	68.36	81.27	**86.90**	84.52	73.19
	BERT	$2k$	5	72.22	64.29	76.98	90.48	84.13	90.48	79.76
	BERT	$5k$	5	**73.81**	61.11	77.78	**92.86**	84.13	90.48	**80.03**
	ELMo	$2k$	10	72.22	65.08	75.40	89.68	76.19	**91.27**	78.31
	ELMo	$5k$	10	72.22	**67.46**	**77.78**	88.10	76.98	89.68	78.57
	GPT-2	$2k$	20	72.22	64.29	73.02	80.16	67.46	82.54	73.28
	GPT-2	$5k$	20	69.84	65.87	69.84	84.13	73.81	85.71	74.87
	ULMFiT	$1k$	10	64.29	57.94	73.02	87.30	80.16	88.89	75.26
	ULMFiT	$2k$	10	64.29	54.76	73.81	88.89	78.57	88.10	74.74
	ULMFiT	$5k$	20	58.73	54.76	75.40	88.89	75.40	84.13	72.88
	C3G-SVM [19]	–	–	33.41	36.53	36.66	57.46	49.91	56.35	45.05
	PPM5 [22]	–	–	52.38	39.68	50.00	57.94	36.51	47.62	47.35
	DV-MA [22]	–	–	43.65	65.87	60.32	71.43	80.16	67.46	64.81

As concerns individual pre-trained language models, BERT and ELMo are better able to surpass the RNN baseline while ULMFiT and GPT-2 are not that competitive. In addition, BERT and ELMo methods need small number of training epochs while ULMFiT and GPT-2 improve with increased number of epochs.

Table 2 also shows the corresponding results from previous studies on cross-topic AA using exactly the same experimental setup. These baselines are based on character 3-grams features and a SVM classifier (C3G-SVM) [19], a compression-based method (PPM5) [22], and a method using text distortion to mask thematic information (DV-MA) [22]. As can be seen, when $C = U$ all of the examined methods surpass the best baseline in average performance and the improvement is high in all genres. It is remarkable that all models except ULMFiT achieve to surpass the baselines (in average performance) even when $C = K$.

Figure 2 presents the mean classification accuracy with respect to vocabulary size on cross-topic AA. Each sub-figure correspond to different LM. The type of the line indicates the normalization corpus, dashed line indicates the use of training texts ($C = K$) as normalization corpus, while the continues line indicates the use of test texts ($C = U$), as noted in Sect. 3 in both cases the texts are unlabeled. The shape of each point correspond to epochs of training, 1, 5, 10 and 20 for circle, triangle, square and x-mark respectively.

From the aspect of vocabulary size, in contradiction to the state of the art [22], where the best results achieved for vocabularies that consisted of less than $1k$ words (most frequent), in our set up the most appropriate value seems to be above $2k$. Despite the gap between $2k$ and $5k$ words in vocabulary size BERT and ELMo have minor difference in accuracy indicating that above $2k$ words the affect of vocabulary size is minor. GPT-2 continues to increment the accuracy and ULMFiT started to decrement for values above $1k$ words, Table 2. Experiments with values over $5k$ were prohibitive due to runtime of training, with $5k$ words the runtime was approximately 4 days for each model running on GPU.

From the aspect of training epochs, BERT and ELMo achieved their best performance in $C = K$ case with minimal training. In $C = U$ case their performance is slightly affected by the number of training epochs. This behavior raises the question of over-fitting. As mentioned in Sect. 3 the selection criterion Eq. 2, is based on the cross-entropy of each text. MHC is trained on predicting the text flow and thus the cross-entropy decreases after each epoch of training. Having in mind the cross-entropy, if we have a second look on Fig. 2, the case of over-fitting is rejected since the behavior of accuracy in relevance with the number of training epochs (indicated by the shape of point) do not have the characteristics of over-fitting (increment of training epochs decrements the accuracy).

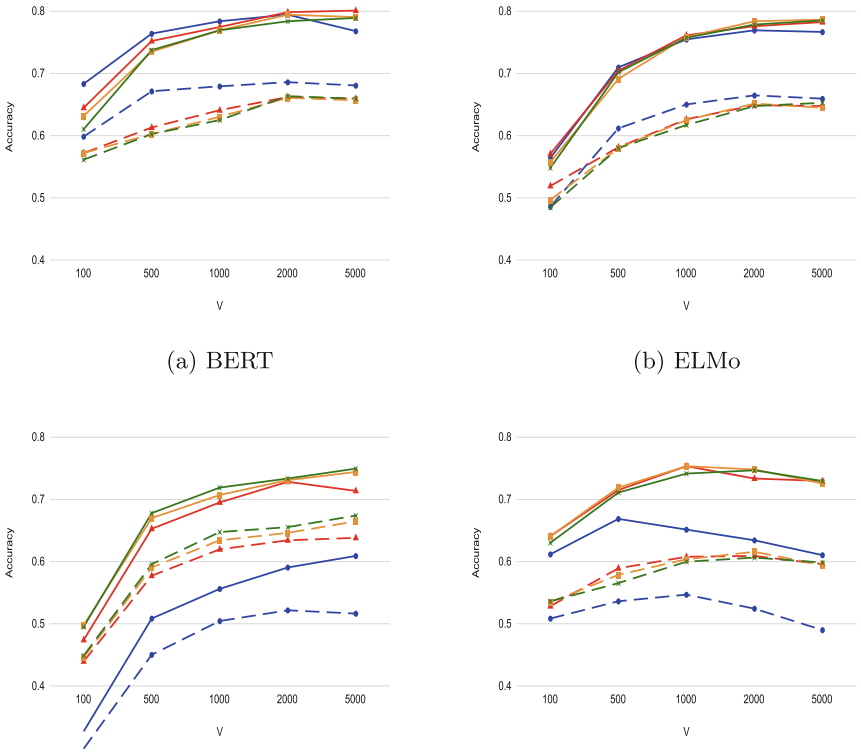
4.4 Results on Cross-Genre AA

The experiments on cross-genre performed on the same set up as in cross-topic. Table 3 presents the accuracy results on leave-one-genre-out cross-validation for each one of the six available topics and the average performance over all topics, similar to Table 2. Based on the results of Sect. 4.3 the most reasonable value of V in order to check the performance of each method is $V = 2k$. The case of $V = 5k$ is very time consuming without offering valuable gain and below $1k$ the performance is not remarkable. For the experiments on cross-genre the values of $1k$ and $2k$ were selected for V. Comparing the two cases, the results with $V = 2k$ surpass in all experiments the results with $V = 1k$ and thus we selected to present only the case of $V = 2k$ on Table 3.

Table 3. Accuracy results (%) on cross-genre AA for vocabulary size $2k$ ($V = 2k$) and each topic (Church (C), Gay Marriage (G), War in Iraq (I), Legalization of Marijuana (M), Privacy Rights (P), Gender Discrimination (S)). The reported performance of the baseline models (only available in average across all topics) is taken from the corresponding publications.

‡	LM	epochs	C	G	I	M	P	S	Avg.
$C = K$	RNN	100	58.89	68.33	71.59	60.24	50.40	62.22	61.94
	BERT	10	70.63	77.78	83.33	73.81	62.70	76.98	74.21
	ELMo	10	68.25	78.57	78.57	71.43	55.56	65.08	69.58
	GPT-2	20	52.38	67.46	61.11	57.94	50.79	53.17	57.14
	ULMFiT	20	72.22	77.78	79.37	70.63	61.11	68.25	71.56
$C = U$	RNN	100	75.32	75.95	86.11	79.52	69.37	74.21	76.75
	BERT	5	84.13	87.30	88.10	82.54	**77.78**	78.57	83.07
	ELMo	20	87.30	88.89	**88.89**	**83.33**	76.98	**81.75**	**84.52**
	GPT-2	20	69.84	76.98	74.60	67.46	61.11	72.22	70.37
	ULMFiT	10	**88.10**	**89.68**	85.71	82.54	**77.78**	79.37	83.86
	C3G-SVM [19]	–							
	PPM5 [22]	–							60.00
	DV-MA [22]	–							33.00

BERT and ELMo achieved high results as expected from their performance on cross-topic, with ELMo achieving the highest accuracy result. Unexpectedly, ULMFiT which had the worst performance in cross-topic achieved the second best performance. GPT-2 performed lower than RNN baseline in both cases of $C = K$ and $C = U$. Comparing Table 2 and Table 3 is noticeable that ELMo and BERT are more stable in performance than GPT-2 and ULMFiT. The main difference between the former and the latter is the directionality, the former two are bidirectional while the latter are unidirectional, we suspect that this is the main reason that affects the stability in performance.



(a) BERT (b) ELMo

(c) GPT-2 (d) ULMFiT

Fig. 2. Accuracy results for vocabulary sizes (V) 100, 500, $1k$, $2k$ and $5k$ for each pre-trained model. Colored symbols blue circle, red triangle, yellow square and green x-mark correspond to 1, 5, 10 and 20 training epochs, respectively. The type of the line indicates the normalization corpus, a dashed line indicates the use of training texts ($C = K$) as normalization corpus, while a solid line indicates the use of test texts ($C = U$). (Color figure online)

5 Conclusions

In this paper, we explore the usefulness of pre-trained language models in cross-domain AA. Based on Bagnall's model [1], originally proposed for authorship verification, we compare the performance when we use either the original character-level RNN trained from scratch in the small-size AA corpus or pre-trained token-based language models obtained from general-domain corpora. We demonstrate that BERT and ELMo pre-trained models achieve the best results while being the most stable approaches with respect to the results in both scenarios.

A crucial factor to enhance performance is the normalization corpus used in the MHC. In cross-domain AA, it is very important for the normalization corpus to have exactly the same properties with the documents of unknown authorship.

In our experiments, using a controlled corpus, it is possible to ensure a perfect match in both genre and topic. In practice, this is not always feasible. A future work direction is to explore how one can build an appropriate normalization corpus for a given document of unknown authorship. Other interesting extensions of this work is to study the effect of extending fine-tuning to language model layers and focus on the different layers of the language modeling representation.

References

1. Bagnall, D.: Author identification using multi-headed recurrent neural networks. In: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum (2015)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
3. Ding, S., Fung, B., Iqbal, F., Cheung, W.: Learning stylometric representations for authorship analysis. IEEE Trans. Cybern. **49**(1), 107–121 (2019)
4. Fourkioti, O., Symeonidis, S., Arampatzis, A.: Language models and fusion for authorship attribution. Inf. Process. Manag. **56**(6), 102061 (2019)
5. Ge, Z., Sun, Y., Smith, M.J.T.: Authorship attribution using a neural network language model. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI 2016, pp. 4212–4213. AAAI Press (2016)
6. Goldstein-Stewart, J., Winder, R., Sabin, R.E.: Person identification from text and speech genre samples. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 336–344. Association for Computational Linguistics (2009)
7. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 328–339 (2018)
8. Kestemont, M., Stover, J., Koppel, M., Karsdorp, F., Daelemans, W.: Authenticating the writings of Julius Caesar. Expert Syst. Appl. **63**, 86–96 (2016)
9. Kocher, M., Savoy, J.: Distributed language representation for authorship attribution. Digital Sch. Humanit. **33**(2), 425–441 (2018)
10. Madigan, D., Genkin, A., Lewis, D.D., Argamon, S., Fradkin, D., Ye, L.: Author identification on the large scale. In: Proceedings of the Meeting of the Classification Society of North America (2005)
11. Menon, R., Choi, Y.: Domain independent authorship attribution without domain adaptation. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, pp. 309–315 (2011)
12. Neal, T., Sundararajan, K., Fatima, A., Yan, Y., Xiang, Y., Woodard, D.: Surveying stylometry techniques and applications. ACM Comput. Surv. **50**(6), 1–36 (2018)
13. Peters, M., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237 (2018)

14. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
15. Rocha, A., et al.: Authorship attribution for social media forensics. IEEE Trans. Inf. Forensics Secur. **12**(1), 5–33 (2017)
16. Rosso, P., Rangel, F., Potthast, M., Stamatatos, E., Tschuggnall, M., Stein, B.: Overview of PAN'16. In: Fuhr, N., et al. (eds.) CLEF 2016. LNCS, vol. 9822, pp. 332–350. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44564-9_28
17. Sapkota, U., Bethard, S., Montes, M., Solorio, T.: Not all character n-grams are created equal: a study in authorship attribution. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93–102 (2015)
18. Sapkota, U., Solorio, T., Montes, M., Bethard, S.: Domain adaptation for authorship attribution: improved structural correspondence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2226–2235 (2016)
19. Sapkota, U., Solorio, T., Montes, M., Bethard, S., Rosso, P.: Cross-topic authorship attribution: will out-of-topic data help? In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 1228–1237 (2014)
20. Stamatatos, E.: A survey of modern authorship attribution methods. J. Am. Soc. Inform. Sci. Technol. **60**(3), 538–556 (2009)
21. Stamatatos, E.: On the robustness of authorship attribution based on character n-gram features. J. Law Policy **21**, 421–439 (2013)
22. Stamatatos, E.: Masking topic-related information to enhance authorship attribution. J. Assoc. Inf. Sci. Technol. **69**(3), 461–473 (2018)
23. Stamatatos, E., Potthast, M., Rangel, F., Rosso, P., Stein, B.: Overview of the PAN/CLEF 2015 evaluation lab. In: Mothe, J., et al. (eds.) CLEF 2015. LNCS, vol. 9283, pp. 518–538. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24027-5_49
24. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)