Style change detection using Siamese neural networks

(Notebook for PAN at CLEF 2021)

Sukanya Nath[1]

[1]*University of Neuchâtel, Switzerland, Avenue du 1er-Mars 26,2000 Neuchâtel, Switzerland*

Abstract

Style change detection focuses on the intrinsic characteristics of a document to determine the location of authorship changes. In this paper, we describe how style change detection can be interpreted as a one shot learning task, which is a type of classification task where a single example of each class is given. For example, one shot learning image classification aims to train classifiers over a dataset, where only one image per class is available. While a large number of classes can be handled in such an approach, it is very difficult to train a neural network using only one sample per class. Siamese neural networks are one of the popular approaches for one shot learning tasks because they can learn the similarity between two object representations. Siamese neural networks have been shown to be useful for learning image similarity and semantic similarity between two texts. In this paper, we propose an end-to-end Siamese neural network approach for learning the stylistic similarity between two texts, even when the size of the texts are relatively short (mean text size of 50 words).

Keywords

style change detection, siamese neural network, multi-author detection, one shot learning

1. Introduction

The goal of style change detection is to identify distinct changes of writing style in a document by focusing on its intrinsic characteristics and is closely related to intrinsic plagiarism. Intrinsic plagiarism detection examines the writing style consistency of the author over the length of the document and has the advantage of being independent of any other reference corpus. The techniques of style change detection could also be useful for fake news detection or other forensics purposes.

The PAN 2021 style change detection task presents three sub-tasks. The first sub-task is to find out if a document is authored by one or multiple authors. Secondly, if the document is authored by multiple authors, locate the style changes in the document. Thirdly, by using the knowledge of the style change locations, assign the texts uniquely to their respective authors. This paper focuses on the first two sub-tasks where we locate the style changes in the document but does not identify the number of distinct authors.

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

sukanya.nath@unine.ch (S. Nath)
https://github.com/8sukanya8 (S. Nath)
0000-0003-4210-9090 (S. Nath)

The rest of the paper is organized as follows. In Section 2, we discuss the background of the problem and the related work. Then, in Section 3, we describe the dataset and the preprocessing required for our models. Section 4 explains our method and architecture while Section 5 includes the evaluation and results. Finally, in Section 6, we draw our conclusions and outline the future work.

In this paper, we present an end-to-end Siamese neural network based approach for style change detection on short texts. The network architecture is based on Mueller [1], however, the unidirectional LSTM is replaced with bidirectional LSTMs and GRUs. Moreover, we use the GloVe embedding [2] and Cosine distance function.

2. Related work and Background

A variety of stylometric features [3, 4, 5, 6] such as lexical, syntactic or structural have been introduced to study the style change variation in a document. Stamatatos in [7] used n-gram profiles to study intrinsic plagiarism.

More recently, approaches based on deep neural networks and word embeddings [8, 2] have been shown to be effective for authorship verification and attribution. Neural network based approaches have also been introduced in the PAN style change detection tasks. In PAN 2018, Schaetti [9] used Convolutional neural networks with a character embedding layer. However, in the PAN 2018 challenge, the style change detection task focussed only on finding out if a document is authored by one or multiple authors and not on the location of the changes. In PAN 2019, Zuo [10] used a feed forward neural network to classify the documents into single author or multiple author documents and further deployed an ensemble of clustering approaches to identify the number of authors. The PAN 2019 style change detection task focussed on finding out if a document is authored by one or multiple authors and the number of authors, but did not focus on the location of the changes. In PAN 2020, the participants were asked to detect if a document is written by one or multiple authors and the location of the changes at paragraph-level. Iyer and Vosoughi [11] used a BERT [12] model to learn sentence level features in PAN 2020. The sentence level features were further assimilated into a paragraph level representation and fed into a random forest model as a classifier. In our knowledge, no complete end-to-end neural network based model has been used for locating style changes in a document.

The style change detection task may also be viewed as a formulation of the one shot learning problem where a prediction must be made, given that, only a single example of a new class is available. One shot learning has been used in the computer vision domain to identify if two given images are similar [13]. In case of style change detection, each pair of texts written by the same author can be considered as a new class. In other words, the one shot learning approach may be used to identify if two texts are similar in terms of style.

In the PAN data set, since the possibility of a style change occurs at the end of a paragraph, each document can be transformed such that any two consecutive paragraphs are paired as a single data point. The label of authorship corresponds to whether they were written by the same author or by two different authors. In this way, the style change detection problem can be transformed into the one shot learning problem.

Siamese neural networks are a widely used approach for the one shot learning problem. Koch

[13] has shown the effectiveness of Convolutional Siamese neural networks for one shot image recognition. A Siamese neural network is composed of two identical twin neural networks with same weights which learn the hidden representation of two different input vectors. The output of both these twin networks is evaluated for similarity using a suitable distance measure. In other words, the two inputs have the same output label which indicates whether they belong to the same class.

Similar approaches have been made for learning textual similarity. Mueller and Thyagarajan [1] have shown that Recurrent Siamese neural networks with LSTM (Long Short-Term Memory) cells can be used for learning sentence similarity using the SICK dataset [14]. Boenninghoff [15] achieved positive results with Hierarchical Recurrent Siamese neural networks for authorship verification with a small PAN dataset. Convolutional Siamese neural networks have also been used for large scale author identification in [16]. However, the length of the texts in their dataset (1000 words in BL2K dataset and 2000 words in the FF dataset) were larger as compared to PAN datasets. The mean number of words per document in the PAN 2021 style change detection dataset is 351, while the mean number of words per paragraph is only 51. This is important because smaller texts have limited contextual information and therefore, are more challenging than larger texts.

Now, Recurrent neural networks (RNN) have internal states for processing variable length sequence data, making them useful in speech and text processing amidst other applications. However, the gradients tend to vanish for long term dependencies in RNNs making it difficult to retain information over several time steps. To solve this vanishing gradient problem [17], gated neural networks such as LSTM [18] and GRU (Gated Recurrent Units) [19] were introduced which use gates to regulate the information flow through the network.

The LSTM unit has a cell and three gates, viz. the input, the output and the forget gates. The input gate determines the amount of content to be added to the cell while the forget gate determines the amount of information to be forgotten. On the other hand, the GRU unit contains a reset gate and an update gate but no hidden state. The update gate decides the amount of information to be updated, and the reset gate allows the unit to forget the previously computed state. In case of GRUs, the entire memory content is exposed. However, in LSTMs, the output gate controls the amount of memory content exposure. Since GRUs have a simpler design as compared to LSTMs, they have fewer parameters to calculate, thereby reducing training time. However, LSTMs have higher expressive power and may perform better than GRUs when the dataset is sufficiently large. A clear winner between the two architectures was not found during empirical evaluations conducted in [19] and [20]. Changing the hyperparameters such as batch-size and number of units can cause large oscillations as shown in [21].

LSTMs (and GRUs) can be unidirectional or bidirectional. Unidirectional LSTMs preserve pre-word information but do not adequately consider the post-word information [22]. In other words, only the positive time direction is followed in such cases. Bidirectional LSTMs have been proposed to solve this problem. Two LSTM layers can be stacked together in bidirectional LSTMs, such that one layer is fed an as it is input sequence, while the other layer is fed a reversed input sequence. By training the layer in both positive and negative time directions, both the pre-word and the post-word contexts can be learned.

Table 1
Dataset statistics

Dataset statistics	Training	Validation	Test
Size	11200	2400	2400
Percentage of single author documents	25.00%	25.00%	NA
Percentage of multi author documents	75.00%	75.00%	NA
Number of authors	1-4	1-4	NA
Range of style changes	0-20	0-19	NA
Mean Document Length (in characters)	1741	1743	1737
Std. Dev. of Document Length (in characters)	825	872	856
Mean Document Length (in words)	**351**	**351**	**349**
Std. Dev. of Document Length (in words)	167	176	173
Mean number of paragraphs per document	7	7	7
Std. Dev. of number of paragraphs per document	3	3	3
Mean paragraph length (in words)	**51**	**51**	**51**
Std. Dev. of paragraph length (in words)	29	29	29
Maximum paragraph length (in words)	**1207**	**445**	**385**
Mean paragraph length (in characters)	**252**	**253**	**254**
Std. Dev. of paragraph length (in characters)	143	144	143

3. Dataset

The PAN 2021 SCD training dataset is based on StackExchange questions and answers which were combined into documents. The training set contained 11,200 documents while the validation set and the test set contained 2,400 documents each. Table 1 shows some statistics related to the dataset. The mean document length in characters of the training, validation and test sets were 1741, 1743 and 1737 respectively. Also, the mean document length in words of the training, validation and test sets were 351, 351 and 349 respectively. The mean number of paragraphs in a document is 7. We noted that the mean number of words per paragraph is 51 while the maximum number of words per paragraph is 1207 in our training set. Here we can observe that the length of the paragraphs which are the inputs to our models were relatively short, however there may be a few exceptions.

The number of style changes ranged from 0 to 20. The number of authors per document ranged from 1 to 4. The proportion of authors over documents were equal, i.e. 25% of the documents were written by one author, another 25% were written by two authors and so on. Therefore, the percentage of single author documents is 25% while the percentage of multi-author documents is 75%. Since the test set was released without the ground truth information, the analysis based on ground truth was not conducted.

3.1. Data Preprocessing for Siamese Neural Networks

As mentioned previously, each document is converted into a set of data points of consecutive paragraphs. It is assumed that each paragraph is written only by one author. This intermediate data representation is shown in Table 2.

In this step the text of each paragraph is converted into a numeric representation format

Table 2
Intermediate data representation of a document

author paragr. 1	author paragr. 2	paragr. 1 text	paragr. 2 text	style
1	2	I kust * realized that some …	Because he leased …	different
2	2	Because he leased …	In general, I advice …	same
2	1	In general, I advice …	My mac is the …	different
1	3	My mac is the …	You should also …	different
3	4	You should also …	You're fine for …	different
4	1	You're fine for …	My modem was …	different
1	3	My modem was …	WPA (if possible, …	different
3	3	WPA (if possible, …	As to the host…	same

*The typographical error was present in the text.

which is required for neural networks. At first, the text is transformed into lower case, then all punctuation (except the ' character) is removed. No stemming or lemmatization was performed. These sequences of words were tokenized and indexed, then converted to a sequence of integers (e.g., "because" = 01, "he"=02, etc.). The index created in this step represents the vocabulary.

Neural networks also require a fixed length input sequence but the length of the text in each paragraph is variable. Therefore, the maximum length threshold of the sequences needs to be set. Sequences having length less than the threshold are padded with zeroes, while sequences having length greater than the threshold are truncated. We have seen in Table 1 that the mean length of a paragraph is 51 words while the maximum length is 1207 words. Setting the maximum length threshold as 1207 would encourage a lot of sparse representation and increase the training time and was therefore, not considered reasonable. After some tuning, we found that the maximum length threshold of 300 words worked the best in our experiments. The final data representation is shown in Table 3. Here, y = 1 indicates different writing style, i.e. different authors, while y = 0 indicates no change in style.

Table 3
Finished data representation of a document

paragr. 1 text (x1)	paragr. 2 text (x2)	style (y)
[3, 24, 25, 26, 27…]	[5, 6, 7, 0, 0…]	1
[5, 6, 7, 0, 0…]	[8, 9, 3, 10, 0…]	0
[8, 9, 3, 10, 0…]	[2, 11, 12, 4, 0…]	1
[2, 11, 12, 4, 0…]	[13, 14, 15, 0, 0…]	1
[13, 14, 15, 0, 0…]	[16, 17, 18, 0, 0…]	1
[16, 17, 18, 0, 0…]	[2, 19, 20, 0, 0…]	1
[2, 19, 20, 0, 0…]	[21, 22, 23, 0, 0…]	1
[21, 22, 23, 0, 0…]	[28, 29, 4, 30, 0…]	0

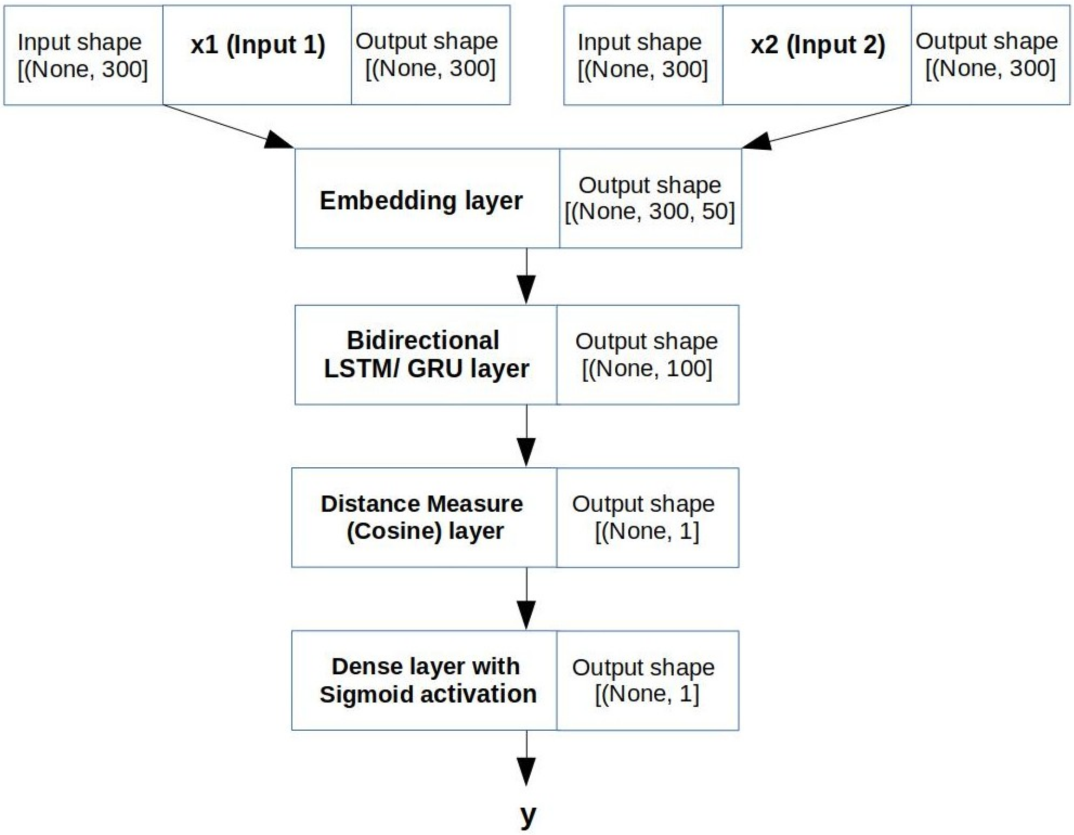

Figure 1: Siamese network architecture

4. Method and Architecture

4.1. Siamese Neural Network Architecture

In Figure 1, the Siamese neural network architecture is presented. The two paragraphs to be evaluated for similarity are converted into sequences of integers as described in Section 3.1. Here the inputs x1 and x2 are the vectors of sequences of integers representing the corresponding paragraphs 1 and 2. The maximum size of each sequence is 300 as mentioned in Section 3.1. Therefore, the shape of both the inputs are [(None, 300)].

The inputs are passed through a 50 dimensional GloVe embedding layer which embeds the inputs into their corresponding higher dimensional word vectors, resulting in an output shape of [(None, 300, 50)]. The next layer is the bidirectional LSTM (or GRU) layer containing n=50 units with the corresponding output shape as [(None,100)]. Then, a distance function computes the similarity between the outputs of the bidirectional LSTM (or GRU) layers and outputs the shape [(None,1)]. The last layer is a dense layer with sigmoid activation which outputs the label y. The binary cross entropy loss function is applied.

As mentioned, the embedding had 50 dimensions while the size of the indexed vocabulary was 55,904. As such, the number of parameters in the embedding layer was 2,795,200 (50 x 55,904). The number of parameters of the bidirectional LSTM layer was 40,400 while that of the bidirectional GRU layer was 30,600.

Adding another layer of bidirectional LSTM or increasing the number of units in the layer did not yield better results but increased the running time.

The total number of parameters in the Siamese neural network with bidirectional LSTM (SNN-LSTM) and the Siamese neural network with bidirectional GRU (SNN-GRU) was found to be 2,835,602 and 2,825,802 respectively.

We compared different distance functions such as Manhattan, Cosine and Matusita. However, the Cosine distance was found to perform slightly better than the others and was the chosen distance function for our models.

5. Evaluation

The experiments were performed on an GPU enabled machine with 8 GB configuration. In Table 4, the results of our experiments are presented. The PAN evaluation strategy used the F1 score for both the sub-tasks.

The SNN-LSTM model performed the best in both the sub-tasks across the training, validation and test sets. The SNN-GRU model performed almost as well as the SNN-LSTM. Both SNN-LSTM and SNN-LSTM performed much better than the random baseline.

Table 4
Results

	Task 1			**Task 2**		
	F1 score			**F1 score**		
Model	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**
SNN-LSTM	**0.75**	**0.71**	**0.70**	**0.70**	**0.65**	**0.65**
SNN-GRU	0.73	0.69	**0.70**	0.68	0.64	0.64
Random Baseline	0.43	0.43	-	0.35	0.35	-

The training time for SNN-LSTM was found to be 1472.05 seconds, while that of SNN-GRU was found to be 1212.91 seconds.

The SNN-LSTM model performed slightly better than the SNN-GRU in our experiments. As LSTMs have higher expressive power than GRUs, owing to their more complex design and higher number of parameters, they might perform better than GRUs, given that sufficient data is available. However, due to the higher number of parameters, the training time of SNN-LSTM was somewhat higher than SNN-GRU.

6. Conclusion

This paper has shown that the style change detection task may be cast as a one-shot learning task. We have presented a simple and effective Siamese neural network based approach for learning the location and number of style changes in a document. The proposed models clearly outperformed the random baseline even when the size of the texts were rather small. An advantage of using a neural network approach is that no explicit feature selection was required.

In the future work, more datasets will be included to check the robustness of the approach presented. One of the criticisms of using neural networks is the black box nature of modelling and the lack of explainability. Visualization techniques such as heatmap representations of features may also be used in the future work for increasing explainability. Recently, a lot of progress has been made with attention based networks in text processing. Hence, approaches involving attention mechanisms and siamese networks can be explored.

References

[1] J. Mueller, A. Thyagarajan, Siamese recurrent architectures for learning sentence similarity, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 30, 2016.

[2] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[3] D. I. Holmes, The evolution of stylometry in humanities scholarship, Literary and linguistic computing 13 (1998) 111–117.

[4] B. Stein, N. Lipka, P. Prettenhofer, Intrinsic plagiarism analysis, Language Resources and Evaluation 45 (2011) 63–82.

[5] M. AlSallal, R. Iqbal, V. Palade, S. Amin, V. Chang, An integrated approach for intrinsic plagiarism detection, Future Generation Computer Systems 96 (2019) 700–712.

[6] E. Stamatatos, Plagiarism detection based on structural information, in: Proceedings of the 20th ACM international conference on Information and knowledge management, 2011, pp. 1221–1230.

[7] E. Stamatatos, Intrinsic plagiarism detection using character n-gram profiles, Threshold 2 (2009).

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, arXiv preprint arXiv:1310.4546 (2013).

[9] N. Schaetti, Character-based convolutional neural network and resnet18 for twitter authorprofiling, in: Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018), Avignon, France, 2018, pp. 10–14.

[10] C. Zuo, Y. Zhao, R. Banerjee, Style change detection with feed-forward neural networks., in: CLEF (Working Notes), 2019.

[11] A. Iyer, S. Vosoughi, Style change detection using bert, in: CLEF, 2020.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, ArXiv preprint arXiv:1810.04805 (2018).

[13] G. Koch, R. Zemel, R. Salakhutdinov, Siamese neural networks for one-shot image recognition, in: ICML deep learning workshop, volume 2, Lille, 2015.

[14] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, R. Zamparelli, et al., A sick cure for the evaluation of compositional distributional semantic models., in: Lrec, Reykjavik, 2014, pp. 216–223.

[15] B. Boenninghoff, R. M. Nickel, S. Zeiler, D. Kolossa, Similarity learning for authorship

verification in social media, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2457–2461.

[16] C. Saedi, M. Dras, Siamese networks for large-scale author identification, ArXiv preprint arXiv:1912.10616 (2019).

[17] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (1994) 157–166.

[18] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[19] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, ArXiv preprint arXiv:1412.3555 (2014).

[20] R. Jozefowicz, W. Zaremba, I. Sutskever, An empirical exploration of recurrent network architectures, in: International conference on machine learning, PMLR, 2015, pp. 2342–2350.

[21] W. Yin, K. Kann, M. Yu, H. Schütze, Comparative study of cnn and rnn for natural language processing, arXiv preprint arXiv:1702.01923 (2017).

[22] M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, IEEE transactions on Signal Processing 45 (1997) 2673–2681.

[23] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection, in: 12th International Conference of the CLEF Association (CLEF 2021), Springer, 2021.

[24] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.

[25] E. Zangerle, M. Mayerl, , M. Potthast, B. Stein, Overview of the Style Change Detection Task at PAN 2021, in: CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[26] B. Boenninghoff, S. Hessler, D. Kolossa, R. M. Nickel, Explainable authorship verification in social media via attention-based similarity learning, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 36–45.

[27] J. Savoy, Machine Learning Methods for Stylometry: Authorship Attribution and Author Profiling, Springer International Publishing, 2020.

[28] M. Kocher, J. Savoy, Distance measures in author profiling, Information Processing & Management 53 (2017) 1103–1119.

[29] J. W. Pennebaker, The secret life of pronouns: What our words say about us, Bloomsbury Press, New York, 2011.

[30] M. Potthast, P. Rosso, E. Stamatatos, B. Stein, A decade of shared tasks in digital text forensics at pan, in: European Conference on Information Retrieval, Springer, 2019, pp. 291–300.

[31] J. Savoy, Comparative evaluation of term selection functions for authorship attribution, Digital Scholarship in the Humanities 30 (2013) 246–261.

[32] J. Savoy, Analysis of the style and the rhetoric of the 2016 us presidential primaries, Digital

Scholarship in the Humanities 33 (2017) 143–159.

[33] F. Sebastiani, Machine learning in automated text categorization, ACM computing surveys (CSUR) 34 (2002) 1–47.

[34] J. Savoy, Is Starnone really the author behind Ferrante?, Digital Scholarship in the Humanities 33 (2018) 902–918.

[35] E. Zangerle, M. Tschuggnall, G. Specht, M. Potthast, B. Stein, Overview of the Style Change Detection Task at PAN 2019, in: L. Cappellato, N. Ferro, D. Losada, H. Müller (Eds.), CLEF 2019 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2019.

[36] W. Daelemans, M. Kestemont, E. Manjavavancas, M. Potthast, F. Rangel, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Tschuggnall, M. Wiegmann, E. Zangerle, Overview of PAN 2019: Author Profiling, Celebrity Profiling, Cross-domain Authorship Attribution and Style Change Detection, in: F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D. Losada, G. Heinatz, L. Cappellato, N. Ferro (Eds.), Proceedings of the Tenth International Conference of the CLEF Association (CLEF 2019), Springer, 2019.

[37] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF, Springer, 2019.

[38] E. Zangerle, M. Mayerl, G. Specht, M. Potthast, B. Stein, Overview of the style change detection task at pan 2020, CLEF, 2020.