

Writing Style Change Detection on Multi-Author Documents

Notebook for PAN at CLEF 2021

Rhia Singh¹, Janith Weerasinghe² and Rachel Greenstadt²

¹Macaulay Honors College (Hunter CUNY), 695 Park Avenue, New York, NY 10065, United States of America

²New York University, 6 MetroTech Center, Brooklyn, NY 11201, United States of America

Abstract

This paper describes the approach we took to create a machine learning model for the PAN 2021 Style Change Detection Task. We approached this task by transforming it to an authorship verification task and applying a slightly modified version of our previous authorship verification approach. We extracted stylometric features from each paragraph in each document and used the absolute differences between the feature vectors corresponding to pairs of paragraphs as input to a Logistic Regression classifier, together with the labels indicating if the two paragraphs were written by the same author or not. We then used this model to make predictions for the three style change detection tasks. The model achieved F1 scores of 0.634 on Task 1, 0.657 on Task 2, and 0.432 on Task 3 on the final evaluations.

Keywords

Style Change Detection, Stylometry, Machine Learning, Natural Language Processing

1. Introduction

This paper presents our approach for the Style Change Detection Task [1] at PAN 2021 [2]. The objective of this task was to create a model that would be able to determine if a document is written by multiple authors (Task 1), where the writing style changes (Task 2), and which paragraphs are written by the same author (Task 3). The dataset contains English documents from forums on Stack Exchange. Each record in the dataset consists of a multi-paragraph document, which may or may not be written by the same person. The ground truth specifies whether the document is a multi-author document, where the writing style changes occur, and the author identifiers of each paragraph. The training dataset contained 11, 200 records, and the validation dataset contained 2, 400 records, with each paragraph on average containing about 264 characters and 52 tokens.


To solve this task, we applied an approach similar to the one we used in our previous authorship verification approach[3] at PAN 2020. Given two documents, our authorship verification model predicts if they were authored by the same person or by different people. To do this, we extract stylometric features from the two documents, take the absolute difference between the feature vectors, and then use this vector difference as input to a Logistic Regression classifier.

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

✉ rhia.singh@macaulay.cuny.edu (R. Singh); janith@nyu.edu (J. Weerasinghe); greenstadt@nyu.edu (R. Greenstadt)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

In order to train a verification model for the style change detection task, we used the given ground truth to generate training records by creating paragraph pairs with the corresponding label indicating if they were written by different authors. We then used this verification model to detect style changes in the given documents.

This paper is structured as follows: in Section 2 we will describe our approach, in Section 3 we will present our results of the shared task, and in Section 4 we will discuss some of the relevant previous works. Finally, in Section 5 we will discuss our conclusions and future work.

2. Approach

2.1. Overview

This section describes the approach we took to build a model for style change detection. The preprocessing and feature extraction processes were similar to those discussed by Weerasinghe *et al.* [3]. Our previous authorship verification approach performed quite well on large fanfiction documents, consisting of around 4800 tokens per document. The paragraphs in this task are significantly smaller, which makes it challenging to extract meaningful stylometric features. We attempted to address this challenge by integrating more dense features. We used a Logistic Regression classifier for our approach to predict how similar the writing style of each paragraph was to other paragraphs in the document. We used these predictions to find solutions to the three style change detection tasks. We implemented our approach on Python with NLTK [3] and Scikit Learn [9] libraries. The source code and the models we used for the final prediction are available on our GitHub page.¹

2.2. Problem Setup

The PAN 2021 style change detection task was to answer three authorship related questions given a document D consisting of multiple paragraphs $P = \{p_1, p_2, \dots, p_n\}$. Task 1 is to identify if the document is a multi-author document. Task 2 is to identify the points in the document where style changes occur. The problem setup guarantees that the authorship changes only at paragraph boundaries. Therefore the solution to task 2 is an $(n - 1)$ -length-array which indicates if there is a style change between all adjacent paragraph pairs. Task 3 is to identify the author of each paragraph. The problem setup guarantees that the maximum number of authors in a document is 4. The solution to Task 3 is an n -length-array indicating the author of each paragraph using integers 1 – 4.

2.3. Authorship Verification Model Training

The first step in our approach is to train an authorship verification model that can predict, given two paragraphs (p_i, p_j) , if they were written by the same author. We can generate $n(n + 1)/2$ unique paragraph pairs from a document with n paragraphs and then use the ground truth for Task 3 to generate the appropriate labels, 1 in cases where the two paragraphs are authored by *different authors* and 0 in cases where the same author authored the two paragraphs. The

¹https://github.com/rhiats/style_change_detection_pan2021

11,200 documents in the training set contained 77,252 paragraphs. We were able to generate 279,761 paragraph pairs to train the verification model.

As described in our earlier work [3], the authorship verification model works as follows: given the paragraph pair (p_i, p_j) , we first extract stylometric features, which gives us two vectors, x_i and x_j . Then we take the absolute difference between the two vectors and feed the result, $|x_i - x_j|$, to our classifier. We also ensure that the feature vectors are standardized by first scaling the initial feature vectors representing the paragraphs and then scaling the vector differences so that they have a zero mean and unit variance.

We used SKLearn's `LogisticRegression` classifier with `class_weight` parameter set to be balanced and used the default `lbfgs` solver running for a maximum of 1000 iterations. We used `RandomizedSearchCV` to find the best value for C (the inverse of regularization strength). The feature vectorizers, the scalers, the classifier, and parameter tuning were fitted using the training set. We used the provided validation set to measure performance.

2.3.1. Preprocessing:

We ran each paragraph in each document in the dataset through a series of pre-processing steps before feature extraction. The outputs of the preprocessing steps are stored together with the document, which is passed to the feature extraction step in our pipeline. We will use the following sentence from the training data as a running example in this section:

“There should be some setting file to edit manually I guess.”

Tokenizer: We used the NLTK's `casual_tokenize` method, which uses their `TweetTokenizer` to tokenize the documents. Our initial observations found this method better at handling punctuation marks and words better than the default `Treebank Word Tokenizer`. The tokenized version of the document is stored to be used in the next pre-processing steps and to be used in feature extraction steps.

Part of Speech (POS) Tagging: We used NLTK's `Perceptron Tagger` to perform the parts of speech tagging. The POS tags are stored together with the document, which are used in the next preprocessing steps and feature extraction. The following would be the output of our POS-tagger for the example sentence above:

```
[('there', 'EX'), ('should', 'MD'), ('be', 'VB'), ('some', 'DT'),  
 ('setting', 'VBG'), ('file', 'NN'), ('to', 'TO'), ('edit', 'VB'),  
 ('manually', 'RB'), ('i', 'PRP'), ('guess', 'VBP'), ('.', '.')] ]
```

Generating a Partial Parse Tree (POS Tag Chunking): We trained a `Maxent` (Maximum Entropy) classifier using the `CoNLL 2000 corpus`[4] to do POS tag chunking following the example provided by Bird *et al.* [5] in their NLTK book (Chapter 07). The following would be the output of our parser for the example sentence above:

```
(S  
  (NP There/EX)  
  (VP should/MD be/VB)
```

(NP some/DT setting/VBG file/NN)
 (VP to/TO edit/VB)
 manually/RB
 (NP I/PRP)
 (VP guess/VBP)
 ./.)

2.3.2. Features Used:

This section lists the features that we extract from the preprocessed data. These features are commonly used in most previous stylometry work [6] and we use a very similar pipeline as our earlier work [3]. As discussed before, one of the challenges in this task was the shorter length of the documents. A document needs to be large enough for sparse feature sets such as character and word n-grams to capture more relevant stylometric information. Since paragraphs in this task are fairly short, the amount of information that would be captured by sparse features would be minimal. Therefore, we included more dense feature sets. These are small feature sets that attempt capture style related information from the whole document such as vocabulary richness and POS-tag ratios. The intuition behind including dense features was that, since they are aggregate statistics, they would be able to capture more meaningful non-zero feature values signals, unlike token n-grams. The newly included dense features are marked with an asterisk (*). Several of our features described below are computed in terms of TF-IDF values. We used SKLearn's `TFIDFVectorizer` to compute the TF-IDF vectors for the documents. We set the `min_df` parameter to be 0.1 to ignore tokens that have a document frequency less than 10%.

- **Character n-grams:** TF-IDF values for character n-grams, where $1 \leq n \leq 3$.
- **POS-Tag n-grams:** TF-IDF value of POS-Tag tri-grams.
- **Special Characters:** TF-IDF values for 31 pre-defined special characters².
- **Frequency of Function Words:** Frequencies of 851 common English words³.
- **Average number of characters per word:** The average number of characters per token.
- **Distribution of word-lengths (1-10):** The fraction of tokens of length l , where $1 \leq l \leq 10$
- **Vocabulary Richness*:** We included several vocabulary richness measures with the intuition that paragraphs written by the same author will have similar vocabulary richness measures. The first is the ratio of hapax-legomena and dis-legomena. Here, hapax-legomena is the number of words that only occur once in the document and dis-legomena is the number of words that occur twice. In addition, we included the following measures: Type-token ratio, Guiraud's R[7], Herdan's C[8, 9], Dugast's k and U[10], Maas' a^2 [11], Tuldava's LN[12], Brunet's W[13], Carroll's CTTR[14], Summer's S, Sichel's S[15], Michéa's M[16], Honoré's H[17], Herdan's V_m [18], entropy, Yule's K[19], and Simpson's D[20]. We used the implementation of these algorithms in the Python `textcomplexity` package⁴.
- **POS-Tag Chunks:** TF-IDF values for Tri-grams of POS-Tag chunks. Here, we consider the tokens at the second level of our parse tree. For example, for the sentence above, the input to our vectorizer would be ['NP', 'VP', 'NP', 'VP', 'RB', 'NP', 'VP', '.'].
- **POS chunk construction:** TF-IDF values of each noun phrase, verb phrase, and prepositional phrase expansion. This approach is similar to prior work on syntactic n-grams [21, 22]. For the sentence above, these expansions are ['NP[EX]', 'VP[MD VB]', 'NP[DT VBG NN]', 'VP[TO VB]', 'NP[PRP]', 'VP[VBP]']

²Special characters used: !"#%&'()*+,-./:;<=>@[]^_`{|}~

³Downloaded from <https://countwordsfree.com/stopwords>

⁴<https://github.com/tsproisl/textcomplexity>

- **Stop-word and POS tag hybrid tri-grams***: We replaced all the words other than the function words with their part of speech tag and computed the TF-IDF values of the tri-grams from this modified text. Similar methods of *text distortion* have been used successfully in previous studies[23, 24]. Word n-grams creates a very sparse feature set and tends to encode topic related information which could result in undesirable biases in our model. By replacing the all the words except stop-words with their POS-tag, we attempted to capture stylistic information about an author’s word ordering without making the feature set too sparse and making it topic agnostic.
- **Part-of-Speech tag ratios*** Following the work of Castro-Castro *et al.* [25] who computed the ratio of nouns and adjectives, we calculated the proportion of all parts of speech tags in the Penn Treebank POS Tag collection in an attempt to better capture the syntactic composition of the text. We hoped that these ratios would capture how different authors structure sentences. For example, an author who describes words in detail would have a high adjective-to-noun ratio when compared to an author who is not very descriptive.
- **Unique spellings***: The fraction of words that are present in the document that belong to each of the following dictionaries: commonly misspelled English words⁵, common typos when communicating online ⁶, common errors with determiners ⁷, British spelling of words ⁸, and popular online abbreviations ^{9,10}. We included these dictionaries with the aim of capturing the similarities between common typos that authors make and to identify if the authors use of British or American English.

2.3.3. Classifier Training:

We computed the features for all the paragraphs in each document. We also standardize features by removing the mean and scaling to unit variance. Then, we took the absolute vector difference between the feature vectors corresponding to each paragraph pair. We then applied a secondary scaling step to ensure that the vector differences are standardized as well. More formally, for a pair of paragraphs (p_i, p_j) , we represent their scaled feature vectors as x_i and x_j . Then we compute the vector difference as $x = |x_i - x_j|$. Then the input to our classifier will be the scaled version of x , together with the label indicating if the paragraph pair is written by different authors based on the ground truth we obtained from Task 3.

We used SKLearn’s `LogisticRegression` classifier. We found the best value for the C parameter using a randomized parameter search using the `RandomizedSearchCV` implementation.

2.4. Making Style Change Predictions

As discussed in Section 2.3, we now have an authorship verification model that can predict if a given pair of paragraphs are written by the same author. The output of the verification model can also be considered as a measure of the difference of the writing style between the two paragraphs (Since the model was trained to output 1 when the paragraphs were written by *different authors*, a classifier score closer to 1 indicates a highly different writing style.)

Now, we will explain how we used this model to find solutions for the three style change detection tasks. For a document D , which contains $P = \{p_1, p_2, \dots, p_n\}$ paragraphs we first ran the authorship verification model on all adjacent paragraph pairs $(p_i, p_{i+1}) \quad \forall i \in [1, n - 1]$.

⁵<https://www.mentalfloss.com/article/629813/100-commonly-misspelled-words-english>

⁶<https://www.lexico.com/grammar/common-misspellings>

⁷<https://www.ef.edu/english-resources/english-grammar/determiners/>

⁸<https://www.lexico.com/grammar/british-and-spelling>

⁹<https://preply.com/en/blog/2020/05/07/the-most-used-internet-abbreviations-for-texting-and-tweeting>

¹⁰<https://englishstudyhere.com/abbreviations-contractions/50-common-internet-abbreviations/>

For Task 1, we needed to determine if a document was authored by multiple authors. We calculated the average of the classifier scores corresponding to the adjacent paragraph pairs and considered an average greater than 0.5 to be a multi-author document because that meant that the writing style of each paragraph greatly differed.

For Task 2, we needed to predict where the writing style changes. This can be easily obtained using the above classifier scores: if the score is greater than 0.5, we consider that there is a change in writing style between the two adjacent paragraphs. One issue with this approach is that, our model will likely mis-classify a multi-author document that has few style changes where a majority of paragraphs were written by a single author and a smaller number of paragraphs written by a second author. We plan to explore this issue in our future work. One possible solution would be to take the average of predictions between all paragraphs pairs instead of just the adjacent paragraphs.

For Task 3, we needed to predict the author of each paragraph. In this year’s competition, the maximum number of possible authors in a document was 4. We approached this task by grouping paragraphs that had a similar writing style as predicted by our model. We ran our authorship verification model on all the possible paragraph pairs ($n(n + 1)/2$ predictions). We used the classifier scores for these predictions to create an $n \times n$ ‘distance’ matrix. Then we applied hierarchical clustering to group the paragraphs with a low stylometric ‘distance’ between them. More specifically, we used the Scipy’s `linkage` to compute the hierarchical clustering order and then used the `fcluster` method to form the flat clusters. For each document we grouped all the paragraphs pairs that have a classifier score less than 0.5 into the same cluster. We achieved this by running the `fcluster` method with `distance` set as the criterion. Sometimes this results in more than 4 clusters. In such cases, we re-attributed authorship using by setting the `fcluster` criterion to be `maxclust` parameter, where we set the maximum number of clusters to 4.

3. Results

Table 1

Results from our local evaluations, early submissions, and the final evaluations

Description	Task 1	Task 2	Task 3
Early submission	0.622	0.640	0.326
Local validation set	0.649	0.644	0.428
Final evaluation	0.634	0.657	0.432

Table 1 shows the performance of our model under different settings. We submitted an early version of our model during the early submission phase. This version of the model did not include the new vocabulary richness measures, stop-word and POS tag hybrid tri-grams features, POS tag ratios, and unique spellings feature sets. This version of the model was also set to consider a maximum for 5 authors when predicting for Task 3 which caused our model to have a lower performance. Once all the new features are incorporated, we evaluated the performance of our approach locally using the provided validation and deployed these models to the TIRA evaluation system [26] provided by the PAN 2021 organizers where the models were evaluated on an unseen dataset. The performance gain between the early submission and the final submission in Task 1 and 2 are due to the inclusion of the new features.

4. Related Work

A wide variety of approaches have been used to solve style change detection problems. Over the past several years, the PAN workshop series has invited multiple models to the style change detection tasks.

Several approaches submitted to previous PAN style change detection tasks [27, 28, 29] follow a pattern of extracting stylometric features from given texts, computing the stylometric similarity using a distance function, and finding clusters of similar text segments. Castro-Castro *et al.* used a comparison criterion to decide if two feature values across two paragraphs are similar or not and then similarity between the paragraphs are determined by the number of similar features between them. The paragraphs are clustered using B_0 -maximal clustering. Nath [30] computed features for text segments, and different distance measures are used to compute the similarity between text segments which is then passed to Threshold Based and Window Merge clustering algorithms. Zuo *et al.* [31]’s approach included stylometric feature extraction, followed by a feed forward neural network to detect if the document is a multi-author document and then used an ensemble of clustering algorithms including hierarchical clustering and k-means clustering.

Our approach follows a similar pattern to these earlier work. The difference of our approach from previous approaches is that we apply a Logistic Regression classifier after computing the vector difference. Our intuition here was that we can treat the output of the classifier essentially as a “stylometric distance measure”, since the classifier assigns higher weights to more important features,

Apart from stylometric features, word embedding features such as BERT [32], and sentence embeddings [33] have also been used in previous style change detection tasks.

5. Discussion and Conclusion

This paper presented the approach we took in designing a machine learning model for style change detection. Our approach involved extracting stylometric features from each paragraph in a given document, taking the absolute difference of the feature vectors of each paragraph in a given document and using the resulting matrix as input to a machine learning model. This approach allows us to use features that were used in authorship verification problems in a style change detection task. As future work, we would like to optimize our model. We believe that using more dense features and increasing the training set size could address the challenge of handling small text samples and improve performance. We attempted to increase the size of the training set by merging previous PAN style change detection datasets, but did not find a substantial increase in performance. We would like to further investigate why this was the case. Another future step is to perform a feature analysis of our model to see which features are strongly influencing model performance.

6. Acknowledgements

We thank PAN2021 organizers for organizing the shared task and helping us through the submission process. Our work was supported by the National Science Foundation under grant 1931005 and the McNulty Foundation.

References

- [1] 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.
- [2] 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.

- [3] J. Weerasinghe, R. Greenstadt, Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification—Notebook for PAN at CLEF 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéal (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: <http://ceur-ws.org/Vol-2696/>.
- [4] E. F. Tjong Kim Sang, S. Buchholz, Introduction to the conll-2000 shared task: Chunking, in: C. Cardie, W. Daelemans, C. Nedellec, E. Tjong Kim Sang (Eds.), Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal, 2000, pp. 127–132.
- [5] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, " O'Reilly Media, Inc.", 2009.
- [6] E. Stamatatos, A survey of modern authorship attribution methods, Journal of the American Society for information Science and Technology 60 (2009) 538–556.
- [7] P. Guiraud, Les caractères statistiques du vocabulaire: essai de méthodologie, Presses universitaires de France, 1954.
- [8] G. Herdan, Type-token mathematics, volume 4, Mouton, 1960.
- [9] G. Herdan, Quantitative linguistics, london: Buttersworths, See Also (1964).
- [10] D. Dugast, Vocabulaire et stylistique, volume 8, Slatkine, 1979.
- [11] H.-D. Mass, Über den zusammenhang zwischen wortschatzumfang und länge eines textes [relationship between vocabulary and text length, Zeitschrift für Literaturwissenschaft und Linguistik 2 (1972) 73.
- [12] J. Tuldava, Quantitative relations between the size of the text and lexical richness, Journal of Linguistic Calculus (1977) 28–35.
- [13] É. Brunet, Le vocabulaire de Jean Giraudoux, structure et évolution, volume 1, Slatkine, 1978.
- [14] J. B. Carroll, Language and thought, Reading Improvement 2 (1964) 80.
- [15] H. S. Sichel, On a distribution law for word frequencies, Journal of the American Statistical Association 70 (1975) 542–547.
- [16] R. Michéa, Répétition et variété dans l'emploi des mots, Bulletin de la Société de Linguistique de Paris (1969) 1–24.
- [17] A. Honoré, Some simple measures of richness of vocabulary, Association for literary and linguistic computing bulletin 7 (1979) 172–177.
- [18] G. Herdan, A new derivation and interpretation of yule's 'characteristic'k, Zeitschrift für angewandte Mathematik und Physik ZAMP 6 (1955) 332–339.
- [19] C. U. Yule, The statistical study of literary vocabulary, Cambridge University Press, 2014.
- [20] E. H. Simpson, Measurement of diversity, nature 163 (1949) 688–688.
- [21] G. Hirst, O. Feiguina, Bigrams of syntactic labels for authorship discrimination of short texts, Literary and Linguistic Computing 22 (2007) 405–417.
- [22] K. Luyckx, W. Daelemans, Shallow text analysis and machine learning for authorship attribution, LOT Occasional Series 4 (2005) 149–160.
- [23] S. Bergsma, M. Post, D. Yarowsky, Stylometric analysis of scientific articles, in: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2012, pp. 327–337.
- [24] E. Stamatatos, Authorship attribution using text distortion, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 2017, pp. 1138–1149.
- [25] D. Castro-Castro, C. Rodríguez-Losada, R. Muñoz, Mixed Style Feature Representation and B0-maximal Clustering for Style Change Detection—Notebook for PAN at CLEF 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéal (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-

- WS.org, 2020. URL: <http://ceur-ws.org/Vol-2696/>.
- [26] 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.
 - [27] M. Tschuggnall, E. Stamatatos, B. Verhoeven, W. Daelemans, G. Specht, B. Stein, M. Potthast, Overview of the Author Identification Task at PAN 2017: Style Breach Detection and Author Clustering, in: L. Cappellato, N. Ferro, L. Goeuriot, T. Mandl (Eds.), *Working Notes Papers of the CLEF 2017 Evaluation Labs*, volume 1866 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017. URL: <http://ceur-ws.org/Vol-1866/>.
 - [28] 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. URL: <http://ceur-ws.org/Vol-2380/>.
 - [29] E. Zangerle, M. Mayerl, G. Specht, M. Potthast, B. Stein, Overview of the Style Change Detection Task at PAN 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéal (Eds.), *CLEF 2020 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2020. URL: <http://ceur-ws.org/Vol-2696/>.
 - [30] S. Nath, Style Change Detection by Threshold Based and Window Merge Clustering Methods, in: L. Cappellato, N. Ferro, D. Losada, H. Müller (Eds.), *CLEF 2019 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2019. URL: <http://ceur-ws.org/Vol-2380/>.
 - [31] C. Zuo, Y. Zhao, R. Banerjee, Style Change Detection with Feed-forward Neural Networks, in: L. Cappellato, N. Ferro, D. Losada, H. Müller (Eds.), *CLEF 2019 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2019. URL: <http://ceur-ws.org/Vol-2380/>.
 - [32] A. Iyer, S. Vosoughi, Style Change Detection Using BERT—Notebook for PAN at CLEF 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéal (Eds.), *CLEF 2020 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2020. URL: <http://ceur-ws.org/Vol-2696/>.
 - [33] K. Safin, R. Kuznetsova, Style Breach Detection with Neural Sentence Embeddings—Notebook for PAN at CLEF 2017, in: L. Cappellato, N. Ferro, L. Goeuriot, T. Mandl (Eds.), *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*, 11-14 September, Dublin, Ireland, CEUR-WS.org, 2017. URL: <http://ceur-ws.org/Vol-1866/>.