Three Style Similarity: sentence-embedding, auxiliary words, punctuation

Carlos A. Rodríguez-Losada[1], Daniel Castro-Castro[2]

[1]Computer Science Department, University of Oriente "Antonio Maceo", Santiago de Cuba, Cuba
[2]Information Retrieval Lab, Computer Science Department, University of La Coruña, Spain

Abstract

This paper illustrates the strategy followed by the UO-UDC team at the Style Change Detection Shared Task at PAN22. Our model builds a representation from author entries using text embeddings and term frequency vectors. We used a non-supervised approach to Task 1 and tested two similarity decisions for Task 2 and Task 3 based on semantic, punctuation marks, and auxiliary word similarity. The fitting of the model parameters relied on style change means similarity over the released train dataset. The model reached its best results at Task 1 and its worse ones at Task 2.

1. Introduction

Style change detection problem is widely known in fields such as authorship analysis, author style analysis, and multi-authored digital documents classification [1]. Its goal relies on detecting, through intrinsic stylistic features, variations in the writing style to find out if a document was written by several authors. This task is useful to detect plagiarism when no previous document of the author is given. Style change detection is also useful detecting fake news and forensic applications [2].

Since 2017, Style Change detection was included as one of the tasks in the PAN shared evaluation forum [3]. Many interesting solutions were presented along this time [4, 1]. Strøm [5] addresses this problem by proposing a stacked ensemble based on text embedding and text features to increase performance. Iyer et al. [6] used Google AI's open-source BERT [7] to tokenize and generate embeddings for sentences to train a random forest classifier. Safin et al. [8] employed NLTK's sentence tokenizer [9] and skip-thoughts model [1] to build high dimensional sentence vectors in order to represent authors.

Other authors as Singh et al. [10] extracted stylometric features from each paragraph and used it to feed a Logistic Regression Classifier helped by the Scikit Learn library [11]. Deibel et al. [12] trained a two-layered Bidirectional LSTM model using the Keras API [13] with the TensorFlow [14] backend. Castro et al. [15] presented a model that constructed an author representation based on boolean, numeric, and stylistic set features and proposed an incremental algorithm design to detect style changes.

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
carlosarl1999@gmail.com (C. A. Rodríguez-Losada); daniel.castro3@udc.es (D. Castro-Castro)
0000-0001-9102-7601 (D. Castro-Castro)

[1]https://github.com/ryankiros/skip-thoughts

Our solution to the style change problem relies on the representation of paragraphs from digital documents using text embeddings and stylistic features.
This year the Style Change Detection shared Task by PAN22 [16, 17] proposed the evaluation of three subtasks, highlighting Task 3 because it exposes the real-world style change[2]. One relevant aspect is that text from different authors in multi-authored documents presents topic similarity due to the nature of the dataset: StackExchange Q&A.
As usual, the task organizers provide the TIRA platform [18] to perform all the heavy computations by the competitor's models.
This paper is structured as follows. Section 2 provides an overview of our model, its stages that eventually lead to predictions, and the task specifications released in the competition. Section 3 shows up our developed experiments and results. Section 4 summarize our findings and present the future work.

2. Our proposal

2.1. Task's specifications

All the tasks to be solved are described by the organizers in [17]
Task 1 aims to find the position of a style change for a text written by two authors that contains a single style change (i.e., cut the text into the two authors' texts on the paragraph-level).
For **Task 2** given a text written by two or more authors, the objective is to find all positions of writing style change (i.e., assign all paragraphs of the text uniquely to some author out of the number of authors assumed for the multi-author document).
Finally, in the **Task 3** for a text written by two or more authors, the challenge is to find all positions of writing style change, where style changes now not only occur between paragraphs but at the sentence level.
Description of all three tasks is illustrated in Figure 1.

2.2. Model overview

Our model stands that a text document could be represented with three different approaches. The first one is centered on the semantic weight of a document. To do so, it was used three well-known embedding models: multi-qa-MiniLM-L6-cos-v1, all-MiniLM-L12-v2, and all-mpnet-base-v2; all of them are Bidirectional Encoder from Transformers (BERT) models for Sentence Similarity [7]. It was chosen multi-qa-MiniLM-L6-cos-v1 since it maps sentences and paragraphs to a normalized 384-dimensional dense vector space and was designed for semantic search. The second embedding model was selected since it was designed to be used as a sentence and short paragraph encoder. The last model was selected since it remains the best quality model of all sentence-transformers models[3] and it is intended to perform sentence similarity tasks[4]. All the three models were fine-tuned overall (Title, Answer) pairs from the StackExchange dataset, as reported by its developers. It was decided to choose more than one embedding model to capture

[2] https://pan.webis.de/clef22/pan22-web/style-change-detection.html
[3] https://www.sbert.net/docs/pretrained_models.html
[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

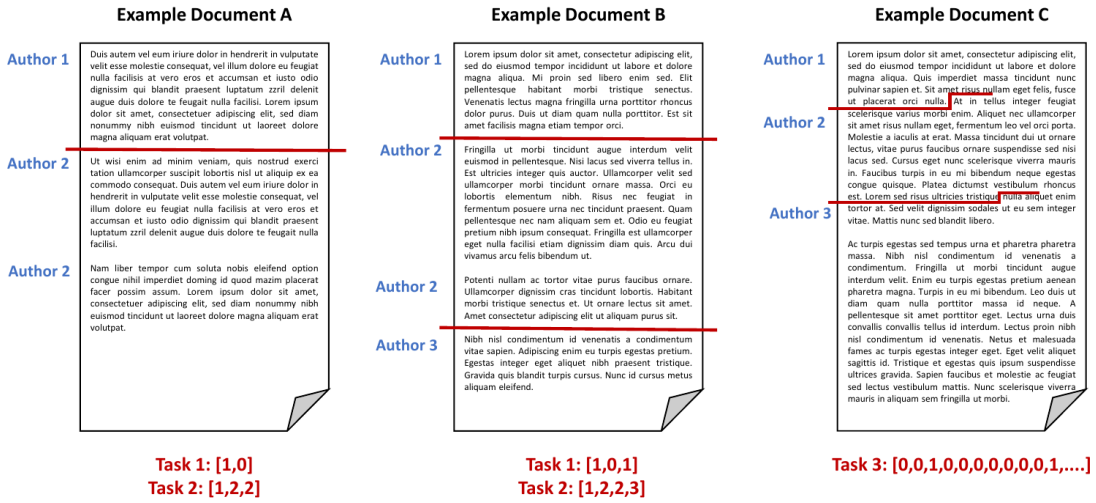

Figure 1: PAN 2022 Style change detection sub tasks (https://pan.webis.de/clef22/pan22-web/style-change-detection.html)

much diverse semantic information from different models sharing the same dataset.
The second representation of a text document is based on punctuation marks, which looks to capture the writing style of an author based on its use of punctuation marks such as emojis, text punctuation, and special characters.
The final representation our models holds is based on the use of discourse markers or auxiliary words because they lead to discrimination between some author the others. Was used in both cases, the Punctuation Representation and the Auxiliary Words Representation the Part-of-Speech (POS) and morphological features tagging module stanza from Stanford NLP-Group.[5]
The model combines these three representations to use a similarity measure between documents influenced by the different stylographic characteristics that each representation could capture.

2.3. Model Architecture

2.3.1. Semantic, Punctuation and Auxiliary Words Representation

As shown in Figure 2 the model encodes all paragraphs/sentences[6] within a document from the dataset to a normalized 384-dimensional dense vectors (embeddings) to build the Semantic Representation (SR) of a given author entry. Secondly, it is computed a Punctuation marks term frequency vector which holds the number of occurrences of a given punctuation mark in the author entry. The possible terms this vector could have are precomputed finding the dataset vocabulary punctuation marks. This vector gives the model the capability to quantify the punctuation writing style similarity between different authors and belongs to the Punctuation

[5]The Natural Language Processing Group at Stanford University is a team of faculty, postdocs, programmers and students who work together on algorithms that allow computers to process, generate, and understand human languages. https://nlp.stanford.edu/

[6]Task 1 and Task 2 are paragraph-oriented and Task 3 is sentence-oriented

Marks Representation (PMR) in the model. Lastly, an auxiliary word term frequency vector is calculated in the same way the PMR builds its vector. The possible terms this vector could have are any token in text tagged as a determiner, conjunction, adposition, pronoun, auxiliary, or adverb. This final vector captures stylistic features regarding the use of discourse markers in speech and belongs to the Auxiliary Words Representation (AWR).

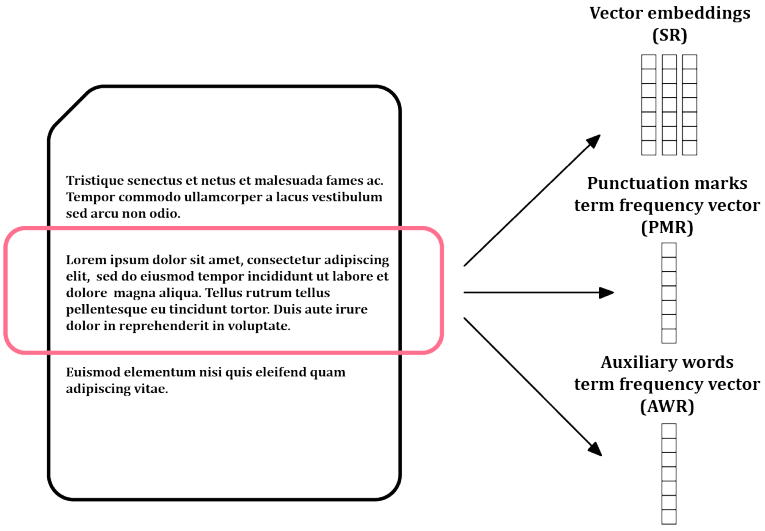

Figure 2: Author entry vector representation for Semantic, Punctuation and Auxiliary Word representation for Task 2

2.3.2. Similarities between author entries

To get a quantification of how similar two author entries are was proposed a method for each given representation.
In the SR was computed a paragraph-level similarity for Task 2. Every paragraph for Task 2 has an associated built embedding vector list corresponding to each sentence from the paragraph encoded using transformers. As the model represents an author entry as a three-embedding vector tuple, the similarity between two author entries is computed by taking the mean similarity between all embedding vectors. Hence, the model can calculate the semantic similarity by taking the mean from the highest vector embedding pair similarities (v_i, v_j) where i represents the i$^{\text{th}}$ sentence embedding vector from a document A and j represents the j$^{\text{th}}$ sentence embedding vector from a document B, such i, j not associated. This is illustrated in Figure 3.
Was calculated the cosine similarity from two vectors as usual:

$$sim(v_1, v_2) = \cos(\theta) = \frac{v_1 . v_2}{\|v_1\| \|v_2\|} \tag{1}$$

Being v_1 and v_2 vectors embeddings from two sentences.

In the graph above, the numbers on the lines connecting two sentences represent the greatest cosine similarities between the embeddings of the encodings of the sentences in paragraph A and the sentences in paragraph B. The general similarity value of paragraph A and B would

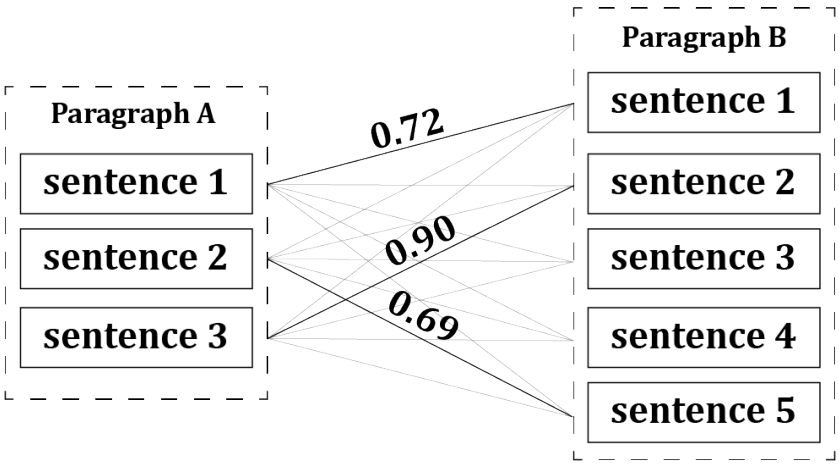

Figure 3: Semantic Representation Similarity Calculation Method.

be the average of these similarities (i.e 0.85). This approach seeks to avoid the influence in the comparison of documents (paragraphs) with different amounts of sentences and to build the similarity based on the most similar sentences.
The model computes both, the PMR similarity and AWR similarity obtaining the cosine similarity between the associated term frequency vectors belonging to each paragraph author.

2.3.3. Style change detection method

So far, the model can respond to how similar any paragraph/sentence within a document to any of the three representations are.
At **Task 1** was taken an unsupervised approach based on similarities between consecutive paragraphs. Since the goal of the task was to predict were occur the only style change in the document, was chosen the change position at the pair of a consecutive paragraphs with the lowest similarity.
It was proposed that should exist thresholds β, γ, θ such that for each representation its corresponding similarity determines whether a style change has occurred. This idea was taken for **Task 2** and **Task 3** and it is illustrated as follows:

$$SCD(D_1, D_2) = \begin{cases} 1 & \begin{cases} SR\_Similarity(D_1, D_2) < \beta \\ PMR\_Similarity(D_1, D_2) < \gamma \\ AWR\_Similarity(D_1, D_2) < \theta \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Were D_1, D_2 are paragraph/sentences entries within a document.
Was provided two methods of SCD when all representations were used, the first one considers a style change if all the criteria in 2 hold. This method was called **Strong Criterion Style Change Detection**. The second one was less sensitive to style changes and consider a style

change if the majority of criteria in 2 holds. It was labeled this method as **Soft Criterion Style Change Detection**.[7]

2.3.4. Assignment of paragraphs in Task 2

In Task 2 it is required to assign all the paragraphs of the text to only one author from the number of authors assumed for the multi-author document. To do so, a solution is given in Algorithm 1.

Algorithm 1: Assignment of paragraphs in Task 2

```
Data: document composed by paragraphs
Result: paragraphs associated with one of the possible authors
mark first paragraph as written by author 1;
for paragraph ∈ document do // this for loops starts at second paragraph
    if paragraph and previous paragraph are similar then
        mark paragraph as written by the same author as previous paragraph author;
    else
        if paragraph is similar to some of the previous marked author's last paragraph then
            if similarity occur only at one paragraph then
                mark paragraph as written by this only similar paragraph author;
            else
                mark paragraph as written by the paragraph author with the largest
                  similarity;
        else
            if number of discovered authors <= 5 then
                mark paragraph as written by a new paragraph author;
            else
                mark paragraph as written by the paragraph author with the largest
                  similarity;
```

3. Experimental results

3.1. Dataset Specification

The dataset used by our model is composed of Stack Exchange [8] entries. The dataset corpora were split into three partitions: train, validation, and test. The distribution of data is shown in Table 1.

[7] Notice Task 1 non-supervised approach doesn't involve any threshold comparison criteria and the methods can't be applied.

[8] Q&A site https://stackexchange.com/

Table 1
Number of documents per task. From https://pan.webis.de/clef22/pan22-web/style-change-detection.html

	Train dataset	Validation dataset	Test Dataset
Task 1	1400	300	300
Task 2	7000	1500	1500
Task 3	7000	1500	1500

3.2. Parameter fitting

Our model parameters were the thresholds β, γ, θ. Was found optimal values for these parameters by getting the three representations (SR, PMR, AWR) in the training dataset. Then, it was possible to determine the mean value of each similarity representation in author entries where a style change occurred and author entries where a style change did not take place. Results are summarized in Table 2, Table 3 and Table 4.

Table 2
Dataset 1 mean similarities over representations

	Style change	Not Style change	Difference
Semantic Representation	0.3088	0.3829	0.0740
Punctuation Marks Representation	0.2678	0.3182	**0.0503**
Auxiliary Words Representation	0.4269	0.5279	0.1010

Table 3
Dataset 2 mean similarities over representations

	Style change	Not Style change	Difference
Semantic Representation	0.3183	0.3567	**0.0384**
Punctuation Marks Representation	0.3406	0.3822	0.0415
Auxiliary Words Representation	0.4878	0.5666	0.0788

Table 4
Dataset 3 mean similarities over representations

	Style change	Not Style change	Difference
Semantic Representation	0.2539	0.2874	0.0334
Punctuation Marks Representation	0.2071	0.2368	**0.0297**
Auxiliary Words Representation	0.5047	0.5667	0.0619

Was estimated the parameters of the models β, γ, θ per task as the mean where style change occurs. Every similarity lower than the specified value should be taken as non-similar documents (i. e) a style change occurred.

3.3. Developed experiments

To evaluate the predictions over the dataset these measures were taken into count: the F1-Score (for the three tasks), the Diarization Error Rate (DER), and the Jaccard Error Rate (JER) for Task 2. [19]
Once the parameters were fixed, was possible to test our approach by tacking the released validation dataset and which representations should be considered, allowing us to test specific representations and their combinations. In Table 5, Table 6 and Table 7 the obtained experimental results are shown.

Table 5
Combining representations and similarity criteria over the task 1 in the validation dataset

Semantic Representation	Punctuation Representation	Auxiliary Representation	Task 1 F1-Score
×			0.5697
	×		0.5677
		×	0.5755
×	×		0.6007
×		×	0.5910
	×	×	0.5929
×	×	×	**0.6065**

Table 6
Experimental results from combining representations and similarity criteria over the task 2 in the validation dataset

SR	PMR	AWR	Soft Criterion	Strong Criterion	Task2 F1-Score
×					0.2801
	×				0.2815
		×			0.2131
×	×				**0.2905**
×		×			0.2798
	×	×			0.2811
×	×	×	×		0.1825
×	×	×		×	0.2896

In Table 8 it is shown the results achieved on the test dataset.

3.4. Results analysis

The representations used, prove not to be very effective when tested independently for Tasks 1 and 2. The unsupervised approach presented in Task 1 obtains the best results using all the representations. In Task 2, the results show a better F1 measurement for the Strong Criterion SCD version vs. the Soft Criterion SCD version, although the best results for the task were not

Table 7
Experimental results from combining representations and similarity criteria over the task 3 in the validation dataset

Semantic Representation	Punctuation Representation	Auxiliary Representation	Soft Criterion	Strong Criterion	Task3 F1-Score
×					**0.5329**
	×				0.4533
		×			0.4379
×	×				0.4126
×		×			0.5313
	×	×			0.4499
×	×	×	×		0.4107
×	×	×		×	0.4107

Table 8
Testing runs over test dataset

Test run	Task	SR	PMR	AWR	Soft Criterion SCD	Strong Criterion SCD	F1-Score
Run 1	Task 1	×	×	×			**0.5836**
	Task 2	×	×	×	×		0.2734
	Task 3	×	×	×	×		0.5565
Run 2	Task 1	×	×				0.5661
	Task 2	×	×	×		×	0.2958
	Task 3	×	×	×		×	0.3976

obtained by combining all the representations, but only considering the SR and the PMR. For Task 3 the best results were obtained considering independently the SR and the combination of the SR and the AWR. Moreover, in Task 3 the worst results were achieved by combining all the representations with the Strong Criterion SCD version as well as the Soft Criterion SCD version.

The work was focused on testing the Strong Criterion SCD vs. the Soft Criterion SCD and that is the reason why it was included these approaches in competition test runs. The results from the test dataset over the validation dataset were consistent with the validation tests, exposing a low-variance and high-bias model. These methods tend to identify more authors than expected, so it is important to evaluate strategies that reduce the significance of features that are frequent at the document level and focus on those features that characterize the paragraph over the document.

For test run 1 of Task 2, the Soft Criterion SCD version combining all representations reported a DER measure of **0.4035** and a JER measure of **0.5771**.

4. Conclusion and future work

Style Change Detection is a very interesting and challenging investigation field in Natural Language Processing. This work shows the results obtained by the UO-UDC team in the Style Change Detection Shared Task at the PAN22 competition. This paper presented all approaches taken by each of the proposed tasks and the results obtained by each strategy. We believe that adding some other writing style representations to the model could improve the system's effectiveness in expressing style changes, (e.g) could be added a phonetic representation from words or characters n-grams, and study for each paragraph the elements that define it self respect frequent elements in the document.

Acknowledgments

This work was supported by projects PLEC2021-007662 (MCIN/AEI/10.13039/501100011033, Ministerio de Ciencia e Innovación, Agencia Estatal de Investigación, Plan de Recuperación, Transformación y Resiliencia, Unión Europea-Next Generation EU) and RTI2018-093336-B-C22 (Ministerio de Ciencia e Innovación, Agencia Estatal de Investigación). The second author also thanks the financial support supplied by the Consellería de Cultura, Educación e Universidade (GPC ED431B 2022/33).

References

[1] J. Bevendorff, B. Chulvi, G. L. D. L. Peña Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, et al., Overview of pan 2021: Authorship verification, profiling hate speech spreaders on twitter, and style change detection, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2021, pp. 419–431.

[2] S. Nath, Style change detection using siamese neural networks, in: CLEF, 2021.

[3] 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: Working Notes Papers of the CLEF 2017 Evaluation Labs/Cappellato, Linda [edit.]; et al., 2017, pp. 1–22.

[4] 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. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/.

[5] E. Strøm, Multi-label style change detection by solving a binary classification problem, in: CLEF, 2021.

[6] A. Iyer, S. Vosoughi, Style change detection using bert., in: CLEF (Working Notes), 2020.

[7] 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).

[8] K. Safin, R. Kuznetsova, Style breach detection with neural sentence embeddings: Notebook for pan at clef 2017, in: CEUR Workshop Proceedings, 2017.

[9] E. Loper, S. Bird, Nltk: The natural language toolkit, arXiv preprint cs/0205028 (2002).
[10] R. Singh, J. Weerasinghe, R. Greenstadt, Writing Style Change Detection on Multi-Author Documents—Notebook for PAN at CLEF 2021, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021. URL: http://ceur-ws.org/Vol-2936/paper-190.pdf.
[11] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.
[12] R. Deibel, D. Löfflad, Style Change Detection on Real-World Data using an LSTM-powered Attribution Algorithm—Notebook for PAN at CLEF 2021, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021. URL: http://ceur-ws.org/Vol-2936/paper-163.pdf.
[13] R. Khan, M. Gubanov, Weblens: Towards web-scale data integration, training the models, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 5727–5729.
[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th USENIX symposium on operating systems design and implementation (OSDI 16), 2016, pp. 265–283.
[15] 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. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/.
[16] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle, Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection, in: M. D. E. F. S. C. M. G. P. A. H. M. P. G. F. N. F. Alberto Barron-Cedeno, Giovanni Da San Martino (Ed.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), volume 13390 of *Lecture Notes in Computer Science*, Springer, 2022.
[17] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Overview of the Style Change Detection Task at PAN 2022, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR Workshop Proceedings, 2022.
[18] 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.
[19] S. E. Tranter, D. A. Reynolds, An overview of automatic speaker diarization systems, IEEE Transactions on audio, speech, and language processing 14 (2006) 1557–1565.