Overview of the Style Change Detection Task at PAN 2020

Eva Zangerle,¹ Maximilian Mayerl,¹ Günther Specht,¹
Martin Potthast,² and Benno Stein³

¹University of Innsbruck
²Leipzig University
³Bauhaus-Universität Weimar
pan@webis.de http://pan.webis.de

Abstract The goal of style change detection is to identify text positions within a multi-author document at which the author switches. Detecting these positions is a crucial part of processing multi-author documents for purposes of authorship identification. In this year’s PAN style change detection task, we asked the participants to answer the following questions for a given document: (1) Given a document, was it written by multiple authors? (2) For each pair of consecutive paragraphs in a given document, is there a style change between these paragraphs? The task is performed and evaluated on two datasets compiled from an English Q&A platform, which differ in their topical breadth (i.e., the number of different topics that are covered in the documents contained). The paper in hand introduces style change detection as a task and its underlying dataset, surveys the participants’ submissions, and analyzes their performance.

1 Introduction

The task of style change detection aims at detecting positions of author changes within a collaboratively written text. Previous PAN editions paved the way for PAN’20 by analyzing multi-authored documents for style changes. This includes the identification and clustering of text segments by author in 2016 [25]. In 2017, participants were asked to detect whether a given document has been authored by multiple authors, and in that case, to determine the boundaries at which authorship changes [34]. The results showed that accurately determining such boundaries is still beyond current capabilities. Hence, in 2018, the task was relaxed by formulating it as a binary classification problem, where the goal was to predict whether a given document is written by a single author or multiple authors [15]. At PAN 2019, this classification task was extended to also predict the number of authors for multi-author documents [35]. In 2020, the task was steered back into its original direction: Participants were asked to detect whether a document was authored by one or multiple authors, and the positions of style changes at the paragraph-level.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2020, 22-25 September, Thessaloniki, Greece.
The remainder of this paper is structured as follows. Section 2 presents previous style change detection approaches. Section 3 introduces the style change detection task as part of PAN 2020, along with the datasets and evaluation measures employed. Section 4 summarizes the received submissions, and Section 5 analyzes and compares the achieved results and Section 6 concludes the paper.

2 Related Work

Style change detection is closely related to the fields of stylometry, plagiarism detection, and text segmentation. All of them have in common that they rely on intrinsic stylometric analyses of documents, without referring to external documents or corpora for comparison. Hence, stylistic profiles are created that are based on lexical features like character n-grams (e.g., [29, 20]), word frequencies (e.g., [11]) and average word/sentence lengths (e.g., [36]), syntactic features like part-of-speech tag frequencies/structures (e.g., [32]), and structural features such as indentation usage (e.g., [36]).

One of the earliest works on style change detection by analyzing stylometric features to detect author boundaries is by Glover and Hirst [9], which aims at identifying inconsistencies of writing style in collaborative documents. Meyer zu Eißen and Stein [21, 31, 30] were the first to investigate intrinsic plagiarism detection based on style change detection using word frequency classes. Koppel et al. [18, 19] and Akiva and Koppel [1, 2] propose an unsupervised method to decompose multi-author documents into authorial threads by applying clustering methods on lexical features. Tschuggnall et al. [33] proposed an unsupervised decomposition approach based on grammar tree representations, whereas Rexha et al. [24] use stylistic features to predict the number of authors who wrote a text. Bensalem et al. [3] rely on n-grams to identify author style changes. Gianella [8] employs Bayesian modeling to split a document by authorship. Further approaches include that of Graham et al. [10], who utilize neural networks with several stylometric features.

At PAN 2017 [34], the goal was to find the exact positions of authorship changes. This task was mostly tackled by using stylometric features to characterize sentences and paragraphs and detecting boundaries by computing similarities [14, 16], or by applying outlier detection [26]. For the binary classification task whether a document is single-author or multi-author at PAN 2018 [15], the best performing system is a stacking ensemble classifier based on lexical and syntactical features extracted via multiple sliding window approaches [37]. Alternatively, deep learning approaches such as convolutional neural networks that operate on a character input [28] and recurrent neural networks operating on parse tree features [12] have been proposed. Other participants used stylometric features to compute the similarity of sentences and paragraphs to find homogeneous text segments that correspond to an individual author [17], or as input to a binary ensemble classifier [27]. In addition, to predict the number of authors of a multi-author document, at PAN 2019, Nath [22] uses two clustering approaches based on token frequencies, whereas Zuo et al. [38] use a classification ensemble based on lexical, syntactic and word frequency features.
3 Style Change Detection Task

This section introduces the style change detection task, the dataset constructed for it, the performance measures employed, and our evaluation platform on which the task has been deployed.

3.1 Task Definition

The goal of style change detection is to segment documents into stylistically homogeneous passages, which can subsequently be utilized for authorship identification and attribution. Hence, style change detection aims at identifying text positions within a given multi-author document at which the author changes. Beforehand, it must be determined whether such a segmentation is necessary at all by checking whether the document in question is indeed a multi-author document. We there study the following two tasks:

- **Task 1.** Given a document, is the document written by multiple authors?
- **Task 2.** Given a sequence of paragraphs of a (supposedly) multi-author document, is there a style change between any of the paragraphs?

Figure 1 illustrates the possible scenarios and the expected output for the two tasks. Document A does not contain any style changes and hence, was authored by a single author; Document B contains a single style change between Paragraphs 1 and 2, and Document C contains two style changes. In order to render the task more feasible, we ensure that all documents comprised in our evaluation dataset are written in the same language (English), and that style changes occur only between paragraphs, not within them (i.e., a single paragraph is always authored by a single author and does not contain any style changes). Moreover, the documents may contain between zero and ten style changes, resulting from at most three different authors.
3.2 Dataset Construction

As with last year’s task [35], the datasets are based on data taken from StackExchange. StackExchange is a popular network of Q&A sites, covering a wide range of topics. We used a dump of the questions and answers on the various StackExchange sites\(^1\) as the basis for our datasets. As a first step, we cleaned this data as follows:

- Removal of questions and answers that contain fewer than 30 characters.
- Removal of questions and answers that were edited by a user different from the one who originally wrote them.
- Removal of the following items within all questions and answers: images, URLs, code snippets, block quotes, and bullet lists along with their contents.

After cleaning the data, we constructed two datasets differing in the number of topics covered to also enable an investigation into how well approaches are able to deal with topical diversity. For the first dataset, called \textit{dataset-narrow}, we used only questions and answers belonging to a subset of the StackExchange sites that deal with topics related to computer technology.\(^2\) For the second dataset, called \textit{dataset-wide}, we used a subset of sites covering a wide range of different topics,\(^3\) including technology, economics, literature, philosophy, and mathematics.

For every site in the subset of sites that was used for the creation of a given dataset, we grouped together all questions and answers written by the same user and split them into paragraphs, removing paragraphs with fewer than 100 characters. This yielded a list of paragraphs for every user on a particular site. In a next step, we constructed the documents making up the dataset. Each dataset contains an equal number of single-author and multi-author documents. For single-author documents, we selected a random user and drew paragraphs from the paragraph list of that user until the document had a sufficient length (randomly chosen to be between 1,000 and 3,000 words). For multi-author documents, we first randomly chose whether the document should have two or three authors. Then, we randomly constructed a structure for the document (i.e., a sequence of author changes for a set of paragraphs). Based on that, we randomly chose distinct authors from our list of users and drew paragraphs from their paragraph lists until the document had the predetermined structure and the chosen length (again, randomly chosen to be between 1,000 and 3,000 words).

The resulting documents were then split into training, validation, and test sets with approximately 50% of the documents being assigned to the training set, and 25% each being assigned to the validation and test sets. The procedure we used for splitting ensured that every subset contains approximately the same number of single-author and multi-author documents. Finally, we filtered all documents based on their language. As

\(^{1}\)See https://archive.org/details/stackexchange
we want our datasets to consist only of English documents, we removed all documents where at least one paragraph was identified as being written in a language other than English. For this, we used the Python library langdetect. A summary of the parameters for both datasets is given in Table 1. Table 2 shows an overview of the created datasets, including the number of contained documents as well as the average document lengths, partitioned by the number of authors.

For development, participants are provided with the documents and ground truth information. For each training and validation document, we provided the number of authors, the StackExchange site the texts were gathered from, the order of the authors within the document, the positions of the style changes, and whether the document was indeed multi-authored.

### 3.3 Performance Measures

To evaluate and compare the submitted approaches, we report both, the achieved performance for the individual subtasks, and their combination as a staged task. Furthermore,
we evaluate the approaches on both datasets individually. Submissions are evaluated using the $F_\alpha$-Measure for each document, where $\alpha = 1$ equally weighs the harmonic mean between precision and recall. For Task 1, we compute the average $F_1$ measure across all documents, and for Task 2, we use the micro-averaged $F_1$ measure across all documents. The submissions for the two datasets are evaluated independently and the resulting $F_1$ measures for the two tasks are averaged across datasets.

### 3.4 Evaluation Framework

To ensure the reproducibility of the submitted solutions, participants were asked to deploy their software on our TIRA platform [23]. Each participant was assigned a virtual machine on TIRA, where the software had to be setup with the only constraint of being executable via a POSIX command. The web frontend of TIRA allows for configuring pieces of software that are deployed within an participant’s virtual machine, and to remotely execute them via an appropriate command. This enabled participants both, to test their software on the freely available training and validation datasets, as well as to self-evaluate their software on the test dataset, which is not freely accessible. TIRA prevents direct access by participants by moving the virtual machine into a secure sandbox before enabling the a deployed software to process a test dataset. This way, TIRA enables blind evaluation, thus foreclosing optimization against the test data. Runs resulting from processing the training, validation, or test data can be evaluated using the aforementioned evaluation measure at the click of a button.

### 4 Survey of Submissions

For this year’s edition of the style change detection task, we received three submissions. However, only two participating teams submitted a working notes paper. In the following, we describe the approaches used in those submissions.

#### 4.1 Mixed Style Feature Representation and B-maximal Clustering

The approach developed by Castro-Castro et al. [6] makes use of a variant of $B_0$-maximal clustering to solve the style change detection task. First, they formulate a representation for a paragraph as a set of 185 stylometric features, consisting of character-based, lexical, and syntactic features, but excluding features which explicitly capture the semantics of the given text. The features are divided into three different categories: boolean features, features consisting of a single floating-point number, and features consisting of vectors of numbers. For each of these categories, a comparison criterion is defined which expresses whether a given feature of two different paragraphs is “similar” or not. Then, the similarity between two paragraphs is defined to be the number of similar features between them.

Based on this, $B_0$-maximal clustering is performed to group the paragraphs in a document into clusters, where every cluster is regarded to be one author. This clustering approach assigns all paragraphs with a similarity larger than a defined threshold to the same cluster. Since this makes it possible for a paragraph to be assigned to multiple
Table 3. Overall results for the style change detection task, ranked by average $F_1$.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Task1 $F_1$</th>
<th>Task2 $F_1$</th>
<th>Avg. $F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iyer and Vosoughi</td>
<td>0.6401</td>
<td>0.8567</td>
<td>0.7484</td>
</tr>
<tr>
<td>Castro-Castro et al.</td>
<td>0.5399</td>
<td>0.7579</td>
<td>0.6489</td>
</tr>
<tr>
<td>Nath</td>
<td>0.5204</td>
<td>0.7526</td>
<td>0.6365</td>
</tr>
<tr>
<td>Baseline (random)</td>
<td>0.5007</td>
<td>0.5001</td>
<td>0.5004</td>
</tr>
</tbody>
</table>

clusters, and hence to multiple authors, a basic approach for deciding to which cluster a paragraph will be assigned is proposed. In such a case, from all possible candidate clusters, the one which contains the paragraph that occurs earliest in the document is chosen. Thus, all the paragraphs in a document are assigned to authors. From this, the tasks posed in this year’s style change detection task are solved as follows: For the first task, it is simply checked whether the clustering has produced more than one cluster. For the second task, the positions in the document are identified, where consecutive paragraphs were assigned to different clusters.

4.2 Style Change Detection Using BERT

The approach of Iyer and Vosoughi [13] is based on using Google’s BERT language model [7] as a feature extractor, and random forests as a classifier. First, the documents contained in the dataset are split into sentences, and every sentence is fed to BERT, taking the outputs of the last four BERT layers to represent a given sentence. Since the size of the feature matrix produced by this depends on the number of tokens in a sentence, the values along the length dimension are summed to obtain a feature matrix of a fixed length. After this, representations are formulated for consecutive pairs of paragraphs (to solve the second task), and the whole document (to solve the first task), based on the representations of sentences, by summing (paragraphs) or averaging (whole documents) the feature values of the sentences that make up the paragraph or document. These feature representations are then used to train random forest models for both tasks.

5 Evaluation Results

The results for the participants submissions as well as a random baseline are given in Table 3. The table shows the $F_1$ score for both tasks, as well as the overall average scores. Both participants’ submissions significantly outperformed the baseline with respect to individual and overall score. The best-performing submission is the one by Iyer and Vosoughi, which achieved the best scores in both tasks as well as in the overall average score. The approach developed by Castro-Castro et al. performs significantly better than the random baseline, but also significantly worse than the approach by Iyer and Vosoughi, forming a middle ground. The approach of Nath performs only slightly worse than that of Castro-Castro et al.

5The participant did not submit their working notes and was hence omitted from further analysis.
We further analyzed the performance of both approaches with regard to the specific properties of the documents in our datasets. First, compared both approaches with respect to single-author versus multi-author documents. The results for this analysis are shown in Figures 2a and b. The approach by Iyer and Vosoughi reaches an $F_1$ score of almost 1.0 for Task 2 on single-author documents. This suggests that it may be beneficial for them to reduce their approach to one model predicting style changes between paragraphs, and then calculating predictions for Task 1 based on the output of that model (i.e., predicting a document to be multi-author if and only if there was at least one style change predicted between the paragraphs of that document). Another point to note is that the approach by Castro-Castro et al. performed best for Task 1 on multi-author documents. This suggests that their model is especially well-suited for detecting documents that have been written by more than one author. Moreover, we analyzed how the performance for both submitted approaches changes depending on the number of authors. The results for this analysis are shown in Figure 2c, confirming that the approach of Castro-Castro et al. performs better for multi-author documents, regardless of whether the number of authors is two or three. It is interesting that Castro-Castro et al.’s approach improves for multi-author documents, whereas that of Iyer and Vosoughi performs best for single-author documents, exerting a sharp drop in performance when a document is written by multiple authors.

Finally, we analyzed the performance of the participants’ approaches dependent on topic diversity (see Figure 2d). In most cases, we found a significant difference in performance between both datasets. The exception to this is the approach by Castro-Castro et al. on Task 1, where the performance on the narrow and wide datasets are almost identical. In all other cases, the performance differs significantly, with performance on the narrow dataset being higher than on the wide dataset, implying that dealing with documents of a diverse topical variety renders the task more difficult.
6 Conclusion

In the 2020 edition of the PAN style change detection task, we asked participants to answer the following questions for a given document: (1) Given a document, was it written by multiple authors? (2) For each pair of consecutive paragraphs in a given document, is there a style change between these paragraphs? Three participants submitted their systems and two participants submitted a working notes paper. The two approaches differed fundamentally, the best-performing system relying on semantic features (i.e., BERT embeddings), while the second-best approach focused on syntactic ones. Future challenges include finding the exact position of authorship changes beyond the paragraph level, and assigning paragraphs to individual authors.

Bibliography


