Overview of the Author Identification Task at PAN-2017: Style Breach Detection and Author Clustering

Michael Tschuggnall[1], Efstathios Stamatatos[2], Ben Verhoeven[3], Walter Daelemans[3], Günther Specht[1], Benno Stein[4], and Martin Potthast[4]

[1]University of Innsbruck, Austria
[2]University of the Aegean, Greece
[3]University of Antwerp, Belgium
[4]Bauhaus-Universität Weimar, Germany

pan@webis.de http://pan.webis.de

Abstract Several authorship analysis tasks require the decomposition of a multi-authored text into its authorial components. In this regard two basic prerequisites need to be addressed: (1) style breach detection, i.e., the segmenting of a text into stylistically homogeneous parts, and (2) author clustering, i.e., the grouping of paragraph-length texts by authorship. In the current edition of PAN we focus on these two unsupervised authorship analysis tasks and provide both benchmark data and an evaluation framework to compare different approaches. We received three submissions for the style breach detection task and six submissions for the author clustering task; we analyze the submissions with different baselines while highlighting their strengths and weaknesses.

1 Introduction

An authorship analysis extracts information about the authors of given documents. There are several related supervised tasks where a set of documents with known information about its authors is available and which can be used to train a model that can extract this information from other documents. Typical examples are authorship attribution (extract the identity of authors) [47] and author profiling (extract demographics such as age and gender of the authors) [40]. The vast majority of published work focus on these two tasks. However, there are cases where authorship-related information in a set of training documents is neither available nor reliable. Examples of unsupervised tasks are intrinsic plagiarism detection (identification of plagiarized parts within a given document without a reference collection of authentic documents) [51], author clustering (grouping documents by authorship) [32, 44], and author diarization (decomposing a multi-authored document into authorial components) [2, 12, 29]. Unsupervised authorship analysis tasks are more challenging but can be applied to every authorship analysis case since they do not require any training material.

Previous editions of PAN focused on specific unsupervised tasks such as author clustering or author diarization [50]. However, it has been observed that it was very

difficult for the submitted approaches to surpass even naive baseline methods. Given the complexity of unsupervised tasks, it is essential to focus on fundamental problems and to study them separately. In the current edition of PAN, we focus on two such fundamental problems:

1. Segmentation of a multi-authored document into stylistically homogeneous parts. We call this task *style breach detection*.
2. Grouping of paragraph-length document parts by authorship. We call this task *author clustering*.

These two tasks are elementary processing steps for both author diarization and intrinsic plagiarism detection. Style breach detection could also be useful in writing style checkers, where it is required to ensure that homogeneous stylistic properties are found within a document. Moreover, author clustering of short (paragraph-length) documents could be useful in analysis of social media texts such as blog posts, comments, and reviews. For example, author clustering could help to identify different user names that correspond to the same person or user accounts that are used by multiple persons.

In this paper we present an overview of the shared tasks in style breach detection and author clustering at PAN-2017. We received three submissions for the former and six submissions for the latter task. The evaluation framework including benchmark data, evaluation measures, and baseline methods is described. In addition, we present an analysis and a survey of the submitted methods.

2 Previous Work

This section reviews related work on style breach detection and author clustering.

2.1 Style Breach Detection

The goal is to find positions within a document where the authorship changes, i.e., where the style changes. Thus, it is closely related to all fields within stylometry, especially intrinsic plagiarism detection [52]. Several approaches exist that deal with the latter, basically by creating stylistic fingerprints that include lexical features such as character n-grams [30, 48], word frequencies [21] or average word/sentence lengths [57], syntactic features such as Part-of-Speech (POS) tag frequencies/structures [54], structural features such as average paragraph lengths, or indentation usages [57]. Using these fingerprints, outliers are sought, either by applying different distance metrics on sliding windows [48] or by storing distance matrices [53, 24].

In contrast, related work targeting multi-author documents is rare. One of the first approaches that uses stylometry to automatically detect boundaries of authors of collaboratively written text was proposed by Glover and Hirst [15], with the aim to provide hints in order to produce a homogeneously written text. Graham et al. [18] utilize neural networks with several stylometric features, and Gianella [12] proposes a stochastic model on the occurrences of words to split a document by authorship. An unsupervised

decomposition of multi-author documents based on grammar features has been evaluated by Tschuggnall and Specht [56].

The diarization task at PAN-2016 [49] dealt with building author clusters within documents. The two submitted approaches use n-grams and other selected stylometric features in combination with a classifier post-processed by a Hidden Markov Model [31], as well as a sentence-based distance metric, computed from several features, that is given to a k-means algorithm in order to build clusters [46].

From a global point of view, style breach detection can also be seen as a text segmentation problem that where a document is split into segments based on the writing style. Common text segmentation approaches divide a text by different topics and/or genres [6]. Compared to an intrinsic stylometric analysis, those approaches have the advantage to be able to build dictionaries or other useful statistics for each targeted topic or genre in advance. Thereby, a wide range of methods is used, often based on the research by Hearst [20], in which the lexical cohesion of terms is analyzed. Other approaches use Bayesian models [10], Hidden Markov Models [5], vocabulary analysis in various forms such as word stem repetitions [36] or word frequency models [41]. While some of the recent papers [42, 34] compare the segmentation approaches on the same data sets, it is in general difficult to compare performances due to the heterogeneous problems and data types.

2.2 Author Clustering

Previous work on author clustering (also called author-based clustering, authorship clustering, or authorial clustering), as it is defined in this paper, is limited. Iqbal et al. [22] describe an approach based on k-means clustering which requires that the number of authors is known, and apply it to a collection of e-mail messages. Layton et al. [32] propose a method that can automatically estimate the number of clusters (authors) in a collection of documents using the iterative positive Silhouette method. The latter has been demonstrated to be useful for clustering validation purposes [33]. These techniques have been applied to literary texts (either books or book samples). Samdani et al. [44] analyze postings in a discussion forum using an online clustering method. Daks & Clark [8] use POS n-grams and spectral clustering and tested their method in a variety of corpora including newspaper articles, political speeches, and literary texts.

A shared task on author clustering documents was included in PAN-2016 [50]. The benchmark collections built for this task comprised texts in three languages (English, Dutch, and Greek) and two genres (opinion articles and reviews) taken from various sources. A total of eight submissions was received and the best-performing model was based on a successful authorship verification method using a character-level multi-headed recurrent neural network [4], closely followed by a simple approach based on word and punctuation mark frequencies [27]. Both of these methods first compute pairwise distances between texts and then form clusters by joining texts that belong to a path of small distances. In general, simple baseline methods, such as placing each text in a distinct cluster were found very competitive since in the benchmark collections the number of single-item clusters was high (>50%) or very high (>75%) [50].

3 Style Breach Detection

As a specific type of author identification, the style breach detection task at PAN-2017 focuses on finding stylistic differences within a text document as a result of having multiple authors collaborating on it. The main goal is to identify *style breaches*, i.e., exact positions in the text where the authorship changes. Thereby no training data is available for the corresponding authors, nor can respective information be gained from potential web searches. From this perspective, this year's task attaches to a series of subtasks of previous PAN events that focused on intrinsic characteristics of text documents. Including intrinsic plagiarism detection [37] and author diarization [49], the main commonality is that the style of authors has to be extracted and quantified in some way in order to tackle those problems. In a similar way, an intrinsic analysis of the writing style is also key to approach the PAN-2017 style breach detection task, which can be summarized as follows:

> Given a document determine whether it is multi-authored
> and, if yes, find the borders where authorship switches.

In contrast to the author clustering task described in Section 4, the goal is to find only borders, and thus it is irrelevant to identify or cluster authors of segments.

The detection of style breaches, i.e., locating borders between different authors, can be seen as an application of a general text segmentation problem. Nevertheless, a significant difference to existing segmentation approaches is that while the latter usually focus on detecting switches of topics or stories [20, 42, 34], the aim of this subtask is to identify borders based on the writing style, disregarding the specific content. While segmentation algorithms may include metrics built from precomputed dictionaries comprising different topics or genres, an additional difficulty results from the fact that the content type is not known and, more importantly, coherent throughout a document.

3.1 Approaches at PAN-2017

This year, five teams registered to the style breach detection task, whereas three of them actively submitted their software to TIRA [16, 38]. In the following, a short summary of each approach is given. Moreover, the creation of two slightly different baseline approaches for comparison is explained.

- *Karaś, Śpiewak & Sobecki* [9]. The authors start by splitting the document into paragraphs, either by detecting multiple newline characters, or, if there are no newlines, by choosing a fixed number of sentences. In the following it is then assumed that style breaches may occur only on paragraph endings. To quantify the style of each paragraph, tf-idf matrices are computed using single words, word 3-grams, stop words, POS tags, and punctuation characters. By concatenating all tf-idf matrices, a paired samples Wilcoxon Signed Rank test [13] is applied—a statistical method to verify whether two given samples stem from the same distribution or not. Computing this test for all pairs of consecutive paragraphs finally yields the final prediction, where a style breach is predicted if the test suggests that the paragraphs come from a different distribution.

- *Khan* [25]. The author also segments the document into sentences within a preprocessing step. Then, the sentences are traversed using two sliding windows that share a sentence in the middle. For each window several statistics are computed, including most frequent POS tags, non-/alphanumeric characters, or words. Moreover, word statistics based on precomputed dictionaries are utilized which include common English words and several sentiment dictionaries. Using all metrics, a similarity score between the two adjacent sliding windows is calculated which is finally compared to a predefined threshold in order to decide whether or not the position before/after the overlapping sentence is predicted as style breach. In the latter case the two sliding windows are merged and considered to be written by a single author; a new second window is created, which is processed as described earlier.
- *Safin & Kuznetsova* [43]. The authors approach the style breach detection task by applying a sentence outlier detection, commonly used in intrinsic plagiarism detection algorithms [48, 55]. After splitting the document into sentences, each one is vectorized using two pretrained skip-thought models [26]. These models can be seen as word embeddings operating on sentences as the atomic units, thereby resulting in 2,400 dimensions for each sentence. The distance between each distinct pair of sentences is stored in a distance matrix (similar to, e.g., [53, 24]) by calculating the cosine distance of the corresponding vectors. Finally, an outlier detection is performed using an optimized threshold, which is compared to the average distance of each sentence. If the distance of a sentence s is larger than the threshold, the beginning of s is marked as a style breach position.
- *Baselines.* To be able to compare the performances of the submitted approaches, two simple baselines have been computed:
  1. *BASELINE-rnd* randomly places between 0 and 10 borders at arbitrary positions inside a document.
  2. As a slight variant, *BASELINE-eq* also decides on a random basis how many borders should be placed (also 0-10), but then places the borders uniformly, i.e., such that all resulting segments are of equal size with respect to tokens contained.

  Both baselines have been computed based on the average of 100 runs.

3.2 Data Set

To develop and optimize the respective algorithms, distinct training and test data sets have been provided, which are based on the Webis-TRC-12 data set [39]. The original corpus already served as origin for the PAN'16 diarization data set [49] and contains documents on 150 topics used at the TREC Web Tracks from 2009-2011 [7], which was created by hiring professional writers through crowdsourcing. Each writer was asked to search for a given topic including assignments (e.g., "Barack Obama", assignment: include information about Obama's family) and to compose a single document from the search results. All sources of the resulting document are annotated respectively, so the origin of each text fragment is known.

Table 1. Parameters for generating the data sets.

Parameter	Value/s
number of style breaches	0-8
number of collaborating authors	1-5
document length	$\geq$ 100 words
average segment length	$\geq$ 50 words
border positions	(only) at the end of sentences / paragraphs
segment length distribution	equalized / randomly

Assuming that each distinct source represents a different author in the original data set, a training and a test data set have been randomly created from these documents by varying several parameters as shown in Table 1. Beside the number of style breaches or collaborating authors, also authorship boundary types have been altered to be at paragraph or sentence levels, i.e., authors may switch only at the end of paragraphs[1] or also within paragraphs. Nevertheless, in order to not overcomplicate the task and to build more realistic data sets, the atomic units were set to be sentences, i.e., borders may not occur within sentences. With respect to the resulting segment lengths, it has been varied whether they are equalized to be of similar lengths or of random lengths within a document.

As the original corpus has been partly used and published, the test documents have been created from previously unpublished documents only. Overall, the number of documents in the training data set is 187, whereas the test data set contains 99 documents. The final statistics of the generated data sets are presented in Table 2.

3.3 Experimental Setup

Performance Measures The performance of the submitted algorithms have been measured with two common metrics used in the field of text segmentation. The *WindowDiff* metric [35], which is proposed for general text segmentation evaluation, is computed as it still is used widely for such problems. It calculates an error rate between 0 and 1 for predicting borders (whereby 0 indicates a perfect prediction), by penalizing near-misses less than other/complete misses or extra borders. Depending on the problem types and data sets used, text segmentation approaches report near-perfect windowDiff values of less than 0.01, while on the other side the error rate exceeds values of 0.6 and higher under certain circumstances [14]. As an alternative, a more recent adaption of the WindowDiff metric is the *WinPR* metric [45]. It enhances WindowDiff by computing the common information retrieval measures precision (WinP) and recall (WinR) and thus allows to give a more detailed, qualitative statement about the prediction. Internally, WinP and WinR are computed based on the calculation of true and false positives/negatives, respectively, also assigning higher values if predicted borders are closer to the real border position.

Both metrics are computed on a word-level, whereby the participants were asked to provide character positions. This means that the tokenization was delegated to the

[1] to be identified by at least two consecutive line breaks

Table 2. Data set statistics.

		Train	Test
#documents		187	99
#style breaches	0	36 (19%)	20 (20%)
	1-3	81 (43%)	44 (44%)
	4-6	45 (24%)	25 (25%)
	7-8	25 (13%)	10 (10%)
#authors	1	36 (19%)	20 (20%)
	2-3	84 (45%)	44 (44%)
	4-5	67 (36%)	35 (35%)
document length (words)	< 500	13 (7%)	8 (8%)
	500-1000	42 (22%)	24 (24%)
	1000-2000	77 (41%)	50 (51%)
	2000-3000	40 (21%)	13 (13%)
	3000-4000	10 (5%)	1 (1%)
	>= 4000	5 (3%)	3 (3%)
average segment length (words)	< 100	8 (4%)	3 (3%)
	100-250	56 (30%)	28 (28%)
	250-500	61 (33%)	43 (43%)
	500-1000	48 (26%)	20 (20%)
	>= 1000	14 (7%)	5 (5%)
border position	sentence	90 (48%)	46 (46%)
	paragraph	97 (52%)	53 (54%)
segment length distribution	equalized	94 (50%)	55 (56%)
	random	93 (50%)	44 (44%)

evaluator script. For the final ranking of all participating teams, the F-score of WinPR (WinF) is used.

Workflow The participants designed and optimized their approaches with the given, publicly available training data set described earlier. The performance could be measured either locally using a provided evaluator script, or by uploading the respective software to TIRA [16, 38] and running it against the training data set. Because the test data set was not publicly available, it was necessary to use the latter option in this case. I.e., the participants submitted their final software and ran it against the test data without seeing performance results. It was manually ensured that no potentially helpful information about the data set was publicly logged during the execution. Finally, participants were allowed to submit three successful test data runs, whereby the latest submissions are used for the final ranking and for all results presented in Section 3.4.

3.4 Results

The final results of the three submitting teams are shown in Table 3. In case of WinF, the baseline equalizing the segment sizes could be exceeded by only one approach, whereas

Table 3. Style breach detection results. Participants are ranked according to their WinF score.

Rank	Participant	WinP	WinR	WinF	WindowDiff	Runtime
1	Karaś et al.	0.315	0.586	**0.323**	0.546	00:01:19
–	BASELINE-eq	0.337	**0.645**	0.289	0.647	–
2	Khan	**0.399**	0.487	0.289	**0.480**	00:02:23
3	Safin et al.	0.371	0.543	0.277	0.529	00:20:25
–	BASELINE-rnd	0.302	0.534	0.236	0.598	–

the baseline using completely random positions could be outperformed by all participants. With respect to WindowDiff, all approaches perform better than both baselines. Interestingly, besides achieving the best WinF performance, Karaś et al. also needed the shortest runtime for the prediction, whereas Safin et al. required the significantly longest runtime with over 20 minutes, probably by applying the cost-intensive neural sentence embedding technique [43].

Figure 1 depicts details about performances of all approaches including BASELINE"=eq with respect to several parameters. In case of number of style breaches (a), it can be seen that there is a significant difference between the approaches when analyzing documents with no author switches. While the overall winning approach of Karaś et al. performs poorly, Safin et al. achieve their best score for these documents. The result of the latter may be caused by the intrinsic plagiarism detection type of approach that distinguishes between documents containing suspicious sentences and plagiarism-free documents, i.e., containing style breaches or not. The other approaches assume style borders to be existent, which accounts also for the baseline in over 90% of the cases as it chooses a random number of borders between 0-10.

While the number of authors (b) seems to have no significant impact on the performances[2], the document length (c) influences the results, especially for very short and very long documents, respectively. The approach of Karaś et al. basically gets better with the length of the text, achieving best results for the majority of documents within 1,000-2,000 words. Khan achieves good results for both short and long documents, while Safin et al. scores good only in the former case. With respect to the average segment length, the performance of the winning approach of Karaś et al. decreases drastically for segment lengths of over 500 words. Nevertheless, it achieves good results for documents with shorter segments, and, remarkably, the highest score for the documents of very short segment lengths.

Finally, the impact of the border position and the segment length distribution is shown in subfigures (e) and (f) respectively. For the border position, only Karaś et al. indicate a significant improvement when style breaches appear only on paragraph ends. This reflects also their approach, which treats paragraphs as potential natural border positions, and if no paragraphs exist, creates artificial paragraphs using a fixed length of sentences. Moreover, this may also be the reason why the approach performs better for segments of similar lengths, as this scenario better matches the specified creation of artificial paragraphs.

[2] except for the distinction between one or more authors, which is already shown in subfigure (a), where *number of style breaches* = 0 corresponds to *number of authors* = 1

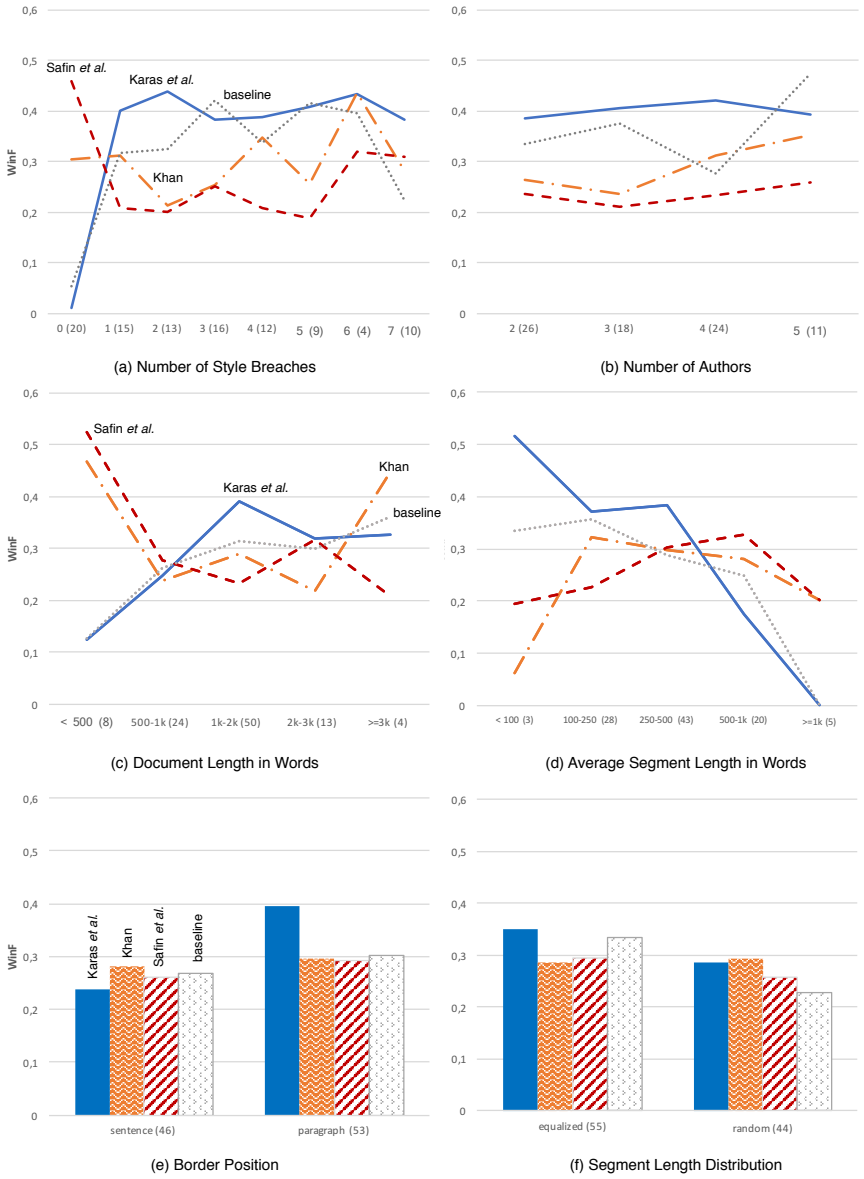

Figure 1. Style breach detection results with respect to number of style breaches, number of authors, document length, average segment length, border position and segment length distribution.

To highlight the potential of the approaches, their individual best results are listed in Table 4. The upper part shows the best configurations for single-authored documents, while the lower part presents the best performances for documents containing style breaches. Again it can be seen that Karaș et al. assume style breaches to be existent and thus reaches very poor results if a document contains no breaches. On the other side,

Table 4. Best style breach detection results per approach for single-authored documents and documents containing style breaches.

Participant	#breaches	#authors	doc. len.	border	seg. distr.	WinF	WindowDiff
Karaś et al.	0	1	418	sent	eq	0.059	0.145
Khan	0	1	337	sent	rand	**1.000**	**0.000**
Safin et al.	0	1	365	par	rand	**1.000**	**0.000**
Karaś et al.	1	2	1027	sent	eq	**0.877**	**0.082**
Khan	1	2	955	par	rand	0.806	0.130
Safin et al.	3	4	692	par	rand	0.634	0.251

Khan as well as Safin et al. achieve perfect prediction rates, i.e., estimating correctly that there are no style borders[3]. In case of documents containing style breaches, Karaś et al. and Khan gain very good top results with WinF scores of over 80%.

4 Author Clustering

4.1 Task Definition

Given a collection D of short (paragraph-length) documents we approach the author clustering task following two scenarios:

- *Complete Clustering.* All documents should be assigned to clusters whereas each cluster corresponds to a distinct author. More specifically, each document $d \in D$ should be assigned to exactly one of k clusters, while k is not given.
- *Authorship-Link Ranking.* Pairs of documents by the same author (authorship-links) should be extracted. For each authorship-link $(d_i, d_j) \in D \times D$, a confidence score belonging to [0,1] should be estimated and authorship-links are ranked in decreasing order.

All documents within a clustering problem are single-authored, in the same language, and belong to the same genre; however, topic and text-length may vary. The main difference with respect to the corresponding PAN-2016 [50] task is that the documents are short including a few sentences (paragraph length). This makes the task harder since text-length is crucial when attempting to extract stylometric information.

4.2 Evaluation Datasets

The datasets used for training and evaluation were extracted from the corresponding PAN-2016 corpora that include three languages (English, Dutch, and Greek) and two genres (articles and reviews). Each PAN-2016 text was segmented into paragraphs and

[3] not shown in the Table, Khan and Safin et al. achieve perfect prediction for several of the documents containing no style breaches

Table 5. The author clustering corpus. Average clusteriness ratio (*r*), number of documents (*N*), number of authors (*k*), number of authorship links, maximum cluster size (maxC), and words per document are given.

	Language	Genre	Problems	*r*	*N*	*k*	Links	maxC	Words
Training	English	articles	10	0.3	20	5.6	57.3	9.2	52.6
	English	reviews	10	0.3	19.4	6.1	45.4	8.2	62.2
	Dutch	articles	10	0.3	20	5.3	61.6	9.8	51.8
	Dutch	reviews	10	0.4	18.2	6.5	19.7	4.0	140.6
	Greek	articles	10	0.3	20	6.0	38.0	6.7	48.2
	Greek	reviews	10	0.3	20	6.1	41.6	7.5	39.4
Test	English	articles	20	0.3	20	5.7	59.3	9.5	52.5
	English	reviews	20	0.3	20	6.4	43.5	7.9	65.3
	Dutch	articles	20	0.3	20	5.7	49.4	8.3	49.3
	Dutch	reviews	20	0.4	18.4	7.1	19.3	4.1	152.0
	Greek	articles	20	0.3	19.9	5.2	59.6	9.6	46.6
	Greek	reviews	20	0.3	20	6.0	42.2	7.6	37.1

all paragraphs with less than 100 characters and more than 500 characters were discarded. In each clustering problem, documents by the same authors were selected randomly by all original documents. This means that paragraphs of the same original document or other documents (by the same author) may be grouped. Certainly, when paragraphs come from the same original document, there is much larger thematic similarity. The only exception in this process was the Dutch reviews corpus because the texts were already short (one paragraph each). In this case, the PAN-2017 datasets were built using the PAN-2016 procedure.

Table 5 shows details about the training and test datasets. Most of the clustering problems include 20 documents (paragraphs) by an average of 6 authors. In each clustering problem there is an average of about 50 authorship links and the largest cluster contains about 8 documents. Each document has an average of about 50 words. Note that in the case of Dutch reviews these figures deviate from the norm (documents are longer and authorship links are less).

An important factor to each clustering problem is the *clusteriness ratio* $r = k/N$, where N is the size of D. When r is high, most documents belong to single-item clusters and there are few authorship links. When r is low, most documents belong to multi-item clusters and there are plenty of authorship links. Estimating r (since N is known, k should be estimated) is crucial for each clustering algorithm. In PAN-2016 three specific values r=0.5, r=0.7, and r=0.9 were used focusing on relatively high values of r [50]. In the current edition, in both training and test datasets, r ranges between 0.1 and 0.5. as can be seen in Figure 2. This means that the PAN-2017 corpus has far less single-item clusters in comparison to PAN-2016.

4.3 Performance Measures

The same evaluation measures introduced in PAN-2016 are used. As a consequence, the results are directly comparable to the ones from the corresponding PAN-2016 task.

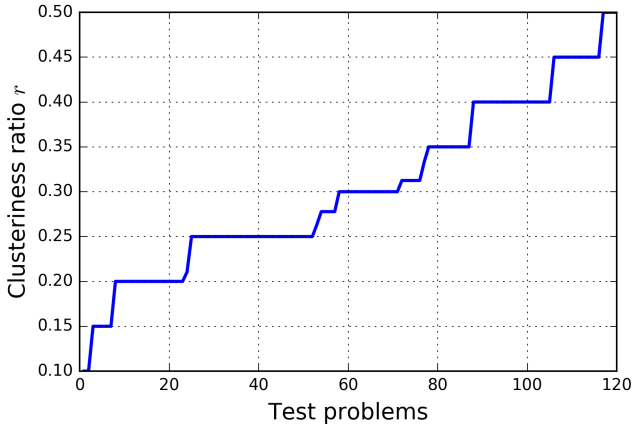

Figure 2. Distribution of clusteriness ratio r values in the test dataset problems.

In more detail, for the complete clustering scenario, Bcubed Recall, Bcubed Precision, and Bcubed F-score are calculated. These are among the best extrinsic clustering evaluation measures and were found to satisfy several important formal constraints including cluster homogeneity, cluster completeness, etc. [3] With respect to the authorship-link ranking scenario, established measures are used to estimate the ability of systems to rank high correct results. These are Mean Average Precision (MAP), R-precision, and P@10.

4.4 Baselines

To understand the complexity of the tasks and the effectiveness of participating systems we used a set of baseline approaches and applied them to the evaluation datasets. The baseline methods range from naive to strong and will allow to estimate weaknesses and strengths of participant approaches. More specifically, the following baseline methods were used:

- *BASELINE-Random.* Given a set of documents, the method randomly chooses the number of authors and randomly assigns each document to one of the authors. It extracts all authorship links from the produced clusters and assigns a random score to each one of them. The average performance of this method over 30 repetitions is reported. This naive approach can only serve as an indication of the lowest performance.
- *BASELINE-Singleton.* This method sets $k = N$, that is all documents are by different authors. It forms singleton clusters and no authorship links. As a result, it is used only for the complete clustering scenario. This simple method was found very effective in PAN-2016 datasets and its performance increases with r [50]. Since the range or r is lower in PAN-2017 datasets, its performance should be negatively affected.

– *BASELINE-Cosine.* Each document is represented by the normalized frequencies of all words occurring at least 3 times in the given collection of documents. Then, for each pair of documents the cosine similarity is calculated and it is used as an authorship-link score. This simple method is only used in the authorship-link ranking scenario and it was found hard-to-beat in PAN-2016 evaluation edition [50].
– *BASELINE-PAN16.* This is the top-performing method submitted to the corresponding PAN-2016 task. It is based on a character-level recurrent neural network and it is a modification of an effective authorship verification approach [4]. There was no attempt to modify this method to be more suitable for the PAN-2017 corpus. Given that it follows a highly conservative approach to form multi-item clusters (suitable for the PAN-2016 corpus) its Bcubed recall is expected to be very low in the current corpus.

4.5 Survey of Submissions

We received six submissions from research teams in Cuba [11], Germany [19], the Netherlands [1], Mexico [17], Poland [23], and Switzerland [28]. All participants also submitted a notebook paper describing their approach.

In general, all submissions follow a *bottom-up* paradigm where first the pairwise similarity between any pair of documents is estimated and then this information is used to form clusters. Gómez-Adorno et al. use hierarchical agglomerative clustering [17] while García et al. use β-compact graph-based clustering. Kocher & Savoy apply some merging criteria in the pairwise similarities [28]. Alberts [1] proposes a modification of a similar method submitted to PAN-2016 [27]. Halvani & Graner use the k-medoids clustering algorithm and Karaś et al. are based on a variation of locality-sensitive hashing [23].

To calculate the pairwise (dis)similarity between documents in a given collection Alberts and Kocher & Savoy propose simple formulas that compare two probability distributions. García et al. use Dice index, Gómez-Adorno et al. use cosine similarity while Halvani & Graner are based on the Compression-based Cosine measure.

A crucial issue is how to estimate the number of clusters k in a given collection of documents. A common choice is the use of Silhouette coefficient to indicate the most suitable number of clusters [19, 23] while Gómez-Adorno et al. use the Calinski-Harabasz index [17]. Another idea is the use of graph-based clustering methods that can be automatically adopted to a clustering problem [1, 11].

As concerns the stylometric measures used to quantify the personal style of authors, most of the submissions are based on low-level character or lexical features such as word and character n-grams. García et al. was the only submission experimenting with higher-level features requiring NLP tools such as lemmatizers and POS taggers, only for the English datasets. Some submissions used a single type of features (e.g., character n-grams [1, 28]) while others used a pool of different feature types and attempted to select the most suitable type (or combination of types) for each language and genre [17, 11]. A feature-agnostic compression-based approach is proposed by Halvani & Graner [19].

Table 6. Overall evaluation results in author clustering (mean values for all clustering problems). Participants are ranked according to Bcubed F-score.

Participant	Complete clustering			Authorship-link ranking			Runtime
	B^3 F	B^3 rec.	B^3 prec.	MAP	RP	P@10	
Gómez-Adorno et al.	**0.573**	**0.639**	0.607	**0.456**	**0.417**	**0.618**	00:02:06
García et al.	0.565	0.518	0.692	0.381	0.376	0.535	00:15:49
Kocher & Savoy	0.552	0.517	0.677	0.396	0.369	0.509	00:00:42
Halvani & Graner	0.549	0.589	0.569	0.139	0.251	0.263	00:12:25
Alberts	0.528	0.599	0.550	0.042	0.089	0.284	00:01:46
BASELINE-PAN16	0.487	0.331	0.987	0.443	0.390	0.583	50:17:49
Karaś et al.	0.466	0.580	0.439	0.125	0.218	0.252	00:00:26
BASELINE-Singleton	0.456	0.304	**1.000**	–	–	–	–
BASELINE-Random	0.452	0.339	0.731	0.024	0.051	0.209	–
BASELINE-Cosine	–	–	–	0.308	0.294	0.348	–

4.6 Evaluation Results

All participant methods were submitted to the TIRA experimentation platform where the participants were able to run their software on both training and test datasets [16, 38]. PAN organizers provided feedback in case a run produced errors or unexpected output. The participants were given the opportunity to perform at most two runs on the test dataset and have been informed about the evaluation results. However, the last run was always considered for the final evaluation.

Table 6 shows the overall evaluation results for both complete clustering and authorship-link ranking on the entire test dataset. The elapsed runtime of each submission is also reported. As can be seen, the method of Gómez-Adorno et al. [17] achieves the best results in both scenarios. Actually, this is the top-performing method taking into account all but one evaluation measures, that is BCubed precision. By definition, BASELINE-singleton achieves perfect Bcubed precision since it provides single-item clusters exclusively. Moreover, BASELINE-PAN16 attempts to optimize precision by following a very conservative strategy when multi-item clusters are considered. Within the PAN-2017 submissions, the approaches of García et al. [11] and Kocher & Savoy [28] are the best ones in terms of Bcubed precision. However, the winning approach of Gómez-Adorno et al. [17] is the only one that achieves both Bcubed recall and precision higher than 0.6. As concerns efficiency, almost all submitted approaches are quite fast. The approaches of García et al. [11] and Halvani & Graner are relatively slower than the rest of submissions. However, both of them are much faster than the very demanding approach of BASELINE-PAN16.

Table 7 provides a more detailed view of performance (Bcubed F-score) in each evaluation dataset separately for the complete clustering scenario. All submitted methods are better than BASELINE-Singleton and BASELINE-Random in overall results. Actually, the performance of these two baseline methods is quite similar, in contrast to the results of PAN-2016 [50]. Moreover, all but one submission were better than BASELINE-PAN16. These observations can be explained by the low values of clusteriness ratio (r) used in PAN-2017 datasets. This means that single-item clusters are not

Table 7. Evaluation results (BCubed F-score) per language and genre for the complete clustering scenario. Participants are ranked according to overall Bcubed F-score.

Participant	Overall	English articles	English reviews	Dutch articles	Dutch reviews	Greek articles	Greek reviews
Gómez-Adorno et al.	**0.573**	**0.618**	0.565	**0.679**	0.474	0.544	**0.552**
García et al.	0.565	0.567	**0.578**	0.614	**0.603**	0.489	0.513
Kocher & Savoy	0.552	0.607	0.570	0.586	0.535	0.511	0.506
Halvani & Graner	0.549	0.534	0.528	0.606	0.519	**0.566**	0.533
Alberts	0.528	0.523	0.487	0.56	0.536	0.524	0.536
BASELINE-PAN16	0.487	0.477	0.483	0.485	0.570	0.426	0.485
Karaś et al.	0.466	0.508	0.428	0.461	0.474	0.498	0.464
BASELINE-singleton	0.458	0.436	0.475	0.438	0.543	0.403	0.455
BASELINE-random	0.452	0.441	0.462	0.437	0.508	0.415	0.450

the majority in PAN-2017 datasets and approaches that attempt to optimize precision over recall are not equally effective as in PAN-2016. Note that in the case of Dutch reviews where r is higher, BASELINE-Singleton and BASELINE-PAN16 are improved.

The approaches of Gómez-Adorno et al. [17], Kocher & Savoy [28], and Halvani & Graner seem to be more effective on articles rather than reviews, while the method of García et al. [11] is not affected significantly by genre. Moreover, the methods of García et al. and Kocher & Savoy seem to be better able to handle English and Dutch texts rather than Greek.

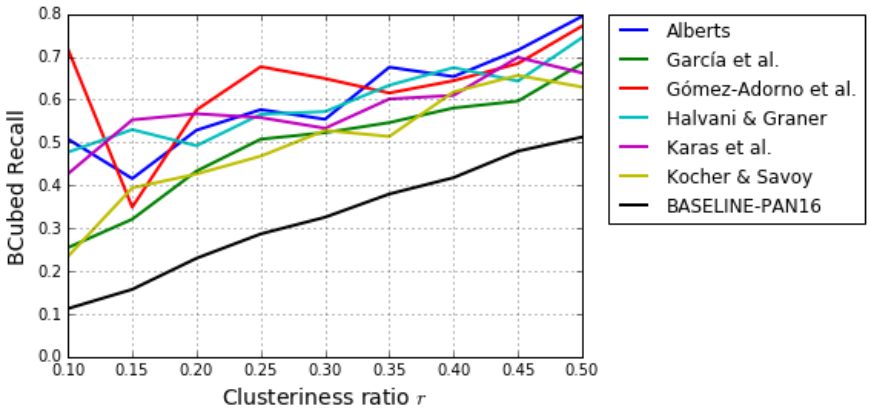

Figure 3. BCubed recall for varying clusteriness ratio values in the test dataset problems.

Figures 3 and 4 show Bcubed recall and precision for varying values of the clustering ratio r in the entire test dataset. As can be seen, the tendency for Bcubed recall is to improve, while Bcubed precision is decreased as r increases. BASELINE-PAN16 suffers in recall that increases almost linearly with r while it maintains almost perfect precision. The approach of Gómez-Adorno et al. achieves the most balanced recall and precision scores especially for relatively low r values. The rest of submissions either

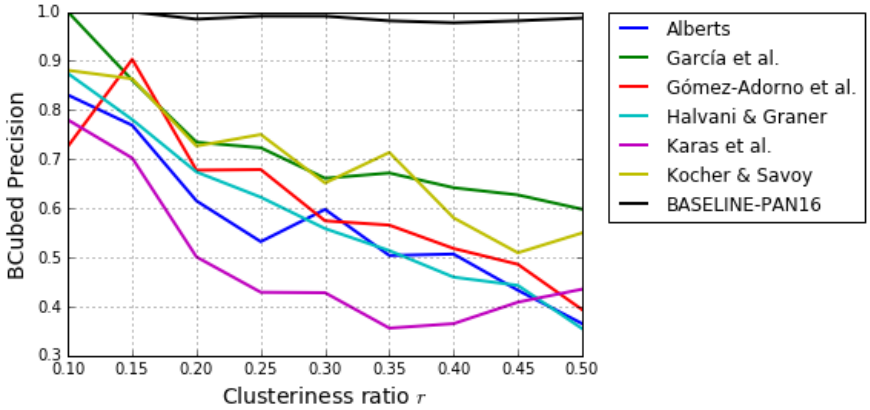

Figure 4. BCubed precision for varying clusteriness ratio values in the test dataset problems.

Table 8. Evaluation results (MAP) per language and genre for the authorship-link ranking scenario. Participants are ranked according to overall MAP.

Participant	Overall	English articles	English reviews	Dutch articles	Dutch reviews	Greek articles	Greek reviews
Gómez-Adorno et al.	**0.455**	0.551	**0.491**	**0.534**	0.311	0.482	**0.422**
BASELINE-PAN16	0.443	**0.554**	0.463	0.532	0.303	**0.505**	0.302
Kocher & Savoy	0.395	0.470	0.386	0.440	0.307	0.445	0.384
García et al.	0.380	0.376	0.421	0.432	**0.318**	0.426	0.366
BASELINE-cosine	0.308	0.388	0.274	0.315	0.211	0.386	0.273
Halvani & Graner	0.139	0.117	0.129	0.152	0.097	0.192	0.145
Karaś et al.	0.125	0.133	0.105	0.138	0.079	0.176	0.148
Alberts	0.042	0.043	0.048	0.049	0.046	0.035	0.029
BASELINE-random	0.024	0.027	0.022	0.023	0.021	0.026	0.023

favor recall (Alberts [1], Halvani & Graner [19], Karaś [23]) or precision (García et al. [11], Kocher & Savoy [28]).

Table 8 shows the evaluation results (MAP) per language and genre for the authorship-link ranking scenario. Here, BASELINE-PAN16 is quite competitive and only the method of Gómez-Adorno et al. [17] is able to surpass it. Moreover, BASELINE-Cosine is quite strong and outperforms half of submissions. Recall that the winning approach of Gómez-Adorno et al. is also based on cosine similarity using a richer set of features and a log-entropy weighting scheme. In general, almost all submissions achieve their worst results in the Dutch reviews dataset. Recall from Table 5 that this dataset has distinct characteristics. Despite the fact that it contains longer texts with respect to the rest of datasets, Dutch reviews form the most difficult case. It seems that the method of Gómez-Adorno et al. [17], Kocher & Savoy [28], and Karaś et al. [23] are better in handling articles than reviews. The same is true for BASELINE-Cosine, indicating that thematic information is more useful in articles. In the authorship-link

scenario, the language factor seems not to be crucial since the evaluation results are balanced over the three examined languages.

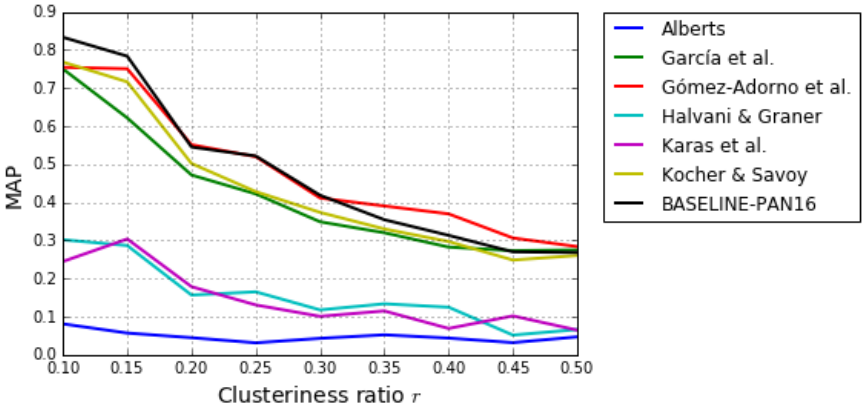

Figure 5. MAP for varying clusteriness ratio values in the test dataset problems.

Figure 5 demonstrates how MAP values are affected by the clusteriness ratio; there are two groups of methods: a group that contains strong methods that can compete with BASELINE-PAN16, and another weak group with low results (also lower from BASELINE-Cosine). Clearly, the method of Gómez-Adorno et al. and BASELINE-PAN16 are better than the methods of Kocher & Savoy and García et al. practically in the whole range of r. The winning approach of Gómez-Adorno et al. is better than BASELINE-PAN16 for relatively high values of r. In addition, the approach of García et al. surpasses the method of Kocher & Savoy only for high values of r. Recall that when r increases, there are less true authorship-links.

5 Discussion

The author identification task at PAN-2017 focused on unsupervised author analysis by decomposing text documents into their authorial components. To study different aspects in detail, two separate subtasks were addressed: (1) style breach detection, aiming to segment a document by stylistic characteristics, and (2) author clustering, aiming to group paragraph-length texts by authorship as well as assigning confidence scores between documents written by the same author. For both tasks, comprehensive data sets have been provided, which allowed participants to train their approaches on the respective training part prior to evaluating them against the inaccessible test part. Although both subtasks seem not that different to approach, e.g., by computing similar stylometric fingerprints, results indicate that intrinsically segmenting a text into distinct authorial components is hard to be tackled. On the other hand, the gap for building clusters of already segmented texts could be narrowed, in large parts due to the outcomes of similar studies conducted in previous PAN events.

For the style breach detection subtask, three approaches have been submitted, utilizing common stylometric features and word dictionaries in combination with different

distance metrics, or by applying a neural network similar to the word embeddings technique. Although all approaches achieved a better performance than the simple random baseline, only one of them could exceed a slightly enhanced baseline, which is also based on random guesses. Interestingly, this winning approach considers only the ends of preformatted paragraphs as possible segment borders, and, if no paragraphs exist, creates artificial paragraphs of predefined, fixed lengths. This fact underlines that there is still room for significant improvements, e.g., by dynamically adjusting the borders. Moreover, another approach basically used an intrinsic plagiarism detection method, which aims at outlier detection over the whole document, marking them as borders. Clearly, tackling the style breach detection task with this method is not optimal since the intrinsic plagiarism detection algorithms assume a main author to be existent. Finally, non of the approaches utilized standard machine learning techniques such as support vector machines. Considering the findings of recent research using such techniques on textual data, it can be assumed that—if optimized and used accordingly—the performance of style breach detection algorithms can be improved significantly.

For the author clustering task, in comparison to the evaluation results of the corresponding task at PAN-2016, the submitted methods achieved lower Bcubed F-score. However, this should not be explained by the fact that text-length in PAN-2017 datasets is much lower. A more crucial factor is the much lower range of the clusteriness ratio r which limits the number of single-item clusters and significantly increases the number of true authorship-links. As a result, the performance of naive baseline methods, like BASELINE-Singleton, is not so competitive as in the corresponding task at PAN-2016. Moreover, MAP scores are considerably increased in comparison to PAN-2016. Given that the MAP scores of BASELINE-PAN16 are also improved with respect to its performance on PAN-2016 datasets, this can be largely explained by the low values of clusteriness ratio again. The success of the top-performing submission shows that very good results can be obtained by using well-known clustering methods and similarity functions given that a suitable feature set and feature weighting scheme is selected for each dataset [17].

Bibliography

[1] Alberts, H.: Author clustering with the aid of a simple distance measure. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Working Notes. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (2017)

[2] Aldebei, K., He, X., Jia, W., Yang, J.: Unsupervised multi-author document decomposition based on hidden markov model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, Volume 1: Long Papers (2016)

[3] Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval 12(4), 461–486 (2009)

[4] Bagnall, D.: Authorship Clustering Using Multi-headed Recurrent Neural Networks. In: CLEF 2016 Working Notes. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (2016)

[5] Blei, D.M., Moreno, P.J.: Topic segmentation with an aspect hidden markov model. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and

Development in Information Retrieval. pp. 343–348. SIGIR '01, ACM, New York, NY, USA (2001), http://doi.acm.org/10.1145/383952.384021
[6] Choi, F.Y.: Advances in Domain Independent Linear Text Segmentation. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference. pp. 26–33. Association for Computational Linguistics (2000)
[7] Clarke, C.L., Craswell, N., Soboroff, I., Voorhees, E.M.: Overview of the TREC 2009 web track. Tech. rep., DTIC Document (2009)
[8] Daks, A., Clark, A.: Unsupervised authorial clustering based on syntactic structure. In: Proceedings of the ACL 2016 Student Research Workshop. pp. 114–118. Association for Computational Linguistics (2016)
[9] Daniel Karaś, M.S., Sobecki, P.: OPI-JSA at CLEF 2017: Author Clustering and Style Breach Detection. In: Working Notes Papers of the CLEF 2017 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2017)
[10] Eisenstein, J., Barzilay, R.: Bayesian Unsupervised Topic Segmentation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 334–343. EMNLP '08 (2008)
[11] García, Y., Castro, D., Lavielle, V., noz, R.M.: Discovering Author Groups Using a β-compact Graph-based Clustering. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Working Notes. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (2017)
[12] Giannella, C.: An improved algorithm for unsupervised decomposition of a multi-author document. JASIST 67(2), 400–411 (2016)
[13] Gibbons, J.D., Chakraborti, S.: Nonparametric Statistical Inference, pp. 977–979. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
[14] Glavaš, G., Nanni, F., Ponzetto, S.P.: Unsupervised text segmentation using semantic relatedness graphs. Association for Computational Linguistics (2016)
[15] Glover, A., Hirst, G.: Detecting stylistic inconsistencies in collaborative writing. In: The New Writing Environment, pp. 147–168. Springer (1996)
[16] Gollub, T., Stein, B., Burrows, S.: Ousting Ivory Tower Research: Towards a Web Framework for Providing Experiments as a Service. In: Hersh, B., Callan, J., Maarek, Y., Sanderson, M. (eds.) 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12). pp. 1125–1126. ACM (Aug 2012)
[17] Gómez-Adorno, H., Aleman, Y., no, D.V., Sanchez-Perez, M.A., Pinto, D., Sidorov, G.: Author Clustering using Hierarchical Clustering Analysis. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Working Notes. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (2017)
[18] Graham, N., Hirst, G., Marthi, B.: Segmenting documents by stylistic character. Natural Language Engineering 11(04), 397–415 (2005)
[19] Halvani, O., Graner, L.: Author Clustering based on Compression-based Dissimilarity Scores. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Working Notes. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (2017)
[20] Hearst, M.A.: Texttiling: Segmenting text into multi-paragraph subtopic passages. Computational linguistics 23(1), 33–64 (1997)
[21] Holmes, D.I.: The evolution of stylometry in humanities scholarship. Literary and Linguistic Computing 13(3), 111–117 (1998)
[22] Iqbal, F., Binsalleeh, H., Fung, B.C.M., Debbabi, M.: Mining writeprints from anonymous e-mails for forensic investigation. Digital Investigation 7(1-2), 56–64 (2010)
[23] Karaś, D., Śpiewak, M., Sobecki, P.: OPI-JSA at CLEF 2017: Author Clustering and Style Breach Detection. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Working Notes. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (2017)

[24] Kestemont, M., Luyckx, K., Daelemans, W.: Intrinsic Plagiarism Detection Using Character Trigram Distance Scores. In: Notebook Papers of the 5th Evaluation Lab on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN). Amsterdam, The Netherlands (September 2011)
[25] Khan, J.A.: Style Breach Detection: An Unsupervised Detection Model. In: Working Notes Papers of the CLEF 2017 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2017)
[26] Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in neural information processing systems. pp. 3294–3302 (2015)
[27] Kocher, M.: UniNE at CLEF 2016: Author Clustering. In: CLEF 2016 Working Notes. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (2016)
[28] Kocher, M., Savoy, J.: UniNE at CLEF 2017: Author Clustering. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Working Notes. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (2017)
[29] Koppel, M., Akiva, N., Dershowitz, I., Dershowitz, N.: Unsupervised decomposition of a document into authorial components. In: Lin, D., Matsumoto, Y., Mihalcea, R. (eds.) Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics. pp. 1356–1364 (2011)
[30] Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. Journal of the American Society for Information Science and Technology 60(1), 9–26 (2009)
[31] Kuznetsov, M., Motrenko, A., Kuznetsova, R., Strijov, V.: Methods for Intrinsic Plagiarism Detection and Author Diarization. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016)
[32] Layton, R., Watters, P., Dazeley, R.: Automated unsupervised authorship analysis using evidence accumulation clustering. Natural Language Engineering 19, 95–120 (2013)
[33] Layton, R., Watters, P., Dazeley, R.: Evaluating authorship distance methods using the positive silhouette coefficient. Natural Language Engineering 19, 517–535 (2013)
[34] Misra, H., Yvon, F., Jose, J.M., Cappe, O.: Text segmentation via topic modeling: an analytical study. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 1553–1556. ACM (2009)
[35] Pevzner, L., Hearst, M.A.: A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics 28(1), 19–36 (2002)
[36] Ponte, J.M., Croft, W.B.: Text Segmentation by Topic. In: Research and Advanced Technology for Digital Libraries, pp. 113–125. Springer (1997)
[37] Potthast, M., Eiselt, A., Barrón-Cedeño, A., Stein, B., Rosso, P.: Overview of the 3rd International Competition on Plagiarism Detection. In: Notebook Papers of the 5th Evaluation Lab on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN). Amsterdam, The Netherlands (September 2011)
[38] Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014)
[39] Potthast, M., Hagen, M., Völske, M., Stein, B.: Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. In: Fung, P., Poesio, M. (eds.) Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 13). pp. 1212–1221. Association for Computational Linguistics (Aug 2013), http://www.aclweb.org/anthology/P13-1119

[40] Rangel Pardo, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., Stein, B.: Overview of the 4th Author Profiling Task at PAN 2016: Cross-Genre Evaluations. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016)
[41] Reynar, J.C.: Statistical Models for Topic Segmentation. In: Proc. of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. pp. 357–364 (1999)
[42] Riedl, M., Biemann, C.: Topictiling: a text segmentation algorithm based on lda. In: Proceedings of ACL 2012 Student Research Workshop. pp. 37–42. Association for Computational Linguistics (2012)
[43] Safin, K., Kuznetsova, R.: Style Breach Detection with Neural Sentence Embeddings. In: Working Notes Papers of the CLEF 2017 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2017)
[44] Samdani, R., Chang, K.W., Roth, D.: A discriminative latent variable model for online clustering. In: Proceedings of The 31st International Conference on Machine Learning. pp. 1–9 (2014)
[45] Scaiano, M., Inkpen, D.: Getting more from segmentation evaluation. In: Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies. pp. 362–366. Association for Computational Linguistics (2012)
[46] Sittar, A., Iqbal, R., Nawab, A.: Author Diarization Using Cluster-Distance Approach. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016)
[47] Stamatatos, E.: A Survey of Modern Authorship Attribution Methods. Journal of the American Society for Information Science and Technology 60, 538–556 (2009)
[48] Stamatatos, E.: Intrinsic Plagiarism Detection Using Character n-gram Profiles. In: Notebook Papers of the 5th Evaluation Lab on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN). Amsterdam, The Netherlands (September 2011)
[49] Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Clustering by Authorship Within and Across Documents. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016), http://ceur-ws.org/Vol-1609/
[50] Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Clustering by Authorship Within and Across Documents. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016)
[51] Stein, B., Lipka, N., Prettenhofer, P.: Intrinsic plagiarism analysis. Language Resources and Evaluation 45(1), 63–82 (Mar 2011)
[52] Stein, B., Lipka, N., Prettenhofer, P.: Intrinsic plagiarism analysis. Language Resources and Evaluation 45(1), 63–82 (2011)
[53] Tschuggnall, M., Specht, G.: Plag-Inn: Intrinsic Plagiarism Detection Using Grammar Trees. In: Proceedings of the 18th International Conference on Applications of Natural Language to Information Systems (NLDB). pp. 284–289. Springer, Groningen, The Netherlands (June 2012)
[54] Tschuggnall, M., Specht, G.: Countering Plagiarism by Exposing Irregularities in Authors' Grammar. In: Proceedings of the European Intelligence and Security Informatics Conference (EISIC). pp. 15–22. IEEE, Uppsala, Sweden (August 2013)
[55] Tschuggnall, M., Specht, G.: Using Grammar-Profiles to Intrinsically Expose Plagiarism in Text Documents. In: Proc. of the 18th Conf. of Natural Language Processing and Information Systems (NLDB). pp. 297–302 (2013)

[56] Tschuggnall, M., Specht, G.: Automatic decomposition of multi-author documents using grammar analysis. In: Proceedings of the 26th GI-Workshop on Grundlagen von Datenbanken. CEUR-WS, Bozen, Italy (October 2014)
[57] Zheng, R., Li, J., Chen, H., Huang, Z.: A framework for authorship identification of online messages: Writing-style features and classification techniques. Journal of the American Society for Information Science and Technology 57(3), 378–393 (2006)