Author Clustering Using Compression-based Dissimilarity Scores

Notebook for PAN at CLEF 2017

Oren Halvani* and Lukas Graner

Fraunhofer Institute for Secure Information Technology SIT
Rheinstrasse 75, 64295 Darmstadt, Germany
{FirstName.LastName}@SIT.Fraunhofer.de

Abstract The *PAN 2017 Author Clustering* task examines the two application scenarios *complete author clustering* and *authorship-link ranking*. In the first scenario, one must identify the number (k) of different authors within a document collection and assign each document to exactly one of the k clusters, where each cluster corresponds to a different author. In the second scenario, one must establish authorship links between documents in a cluster and provide a list of document pairs, ranked according to a confidence score. We present a simple scheme to handle both scenarios. In order to group the documents by their authors, we use k-Medoids, where the optimal k is determined through the computation of silhouettes. To determine links between the documents in each cluster, we apply a predefined compressor as well as a dissimilarity measure. The resulting compression-based dissimilarity scores are then used to rank all document pairs. The proposed scheme does not require (text-)preprocessing, feature engineering or hyperparameter optimization, which are often necessary in author clustering and/or other related fields. However, the achieved results indicate that there is room for improvement.

1 Introduction

Author clustering (AC) is a relatively new sub-discipline in the field of authorship analysis and is offered again by PAN [10] this year as a shared task[1]. Given a collection of documents, the goal of AC is to group documents written by the same author, such that each cluster corresponds to a different author [15]. Formally, the AC problem can be defined as follows: Given a set of n documents $\mathbb{D} = \{\mathcal{D}_1, D_2, \ldots, D_n\}$ the task is to form a clustering $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_k\}$ regarding $\mathbb{D}$ such that each cluster $\mathcal{C}$ comprises documents $\{\mathcal{D}_a, \mathcal{D}_b, \mathcal{D}_c, \ldots\}$ written by the same author $\mathcal{A} \in \mathbb{A}$, where $\mathbb{A}$ denotes a set of k different authors.

The *PAN 2017 Author Clustering* task examines two application scenarios: *complete author clustering* and *authorship-link ranking*. In the first scenario, one must identify k

* Corresponding author.

[1] A shared task is an event, where researchers and practitioners aim to solve or at least make progress on open academic problems.

(the number of different authors within $\mathbb{D}$) while assigning each $\mathcal{D} \in \mathbb{D}$ exactly to one cluster $\mathcal{C} \in \mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_k\}$. In the second scenario, one must establish authorship links between the documents $\{\mathcal{D}_a, \mathcal{D}_b, \mathcal{D}_c, \ldots\}$ in each cluster $\mathcal{C}$ and provide a list of document pairs $(\mathcal{D}_a, \mathcal{D}_b)$, ranked according to a confidence score $\lambda \in [0; 1]$, where λ indicates how likely $\mathcal{D}_a$ and $\mathcal{D}_b$ are to be written by the same author.

We present a simple AC approach based on the k-Medoids algorithm and the computation of so-called silhouettes to determine the optimal k. Instead of using distances computed through well-known metrics such as *Manhattan* or *Euclid*, we decided to experiment with compression-based dissimilarity scores. To compute these scores we apply a compression-based model consisting of a predefined compressor and a dissimilarity measure designed for compressed text files. Compression-based models have been applied widely across different authorship analysis tasks including authorship attribution [5,9] or authorship verification [2,4,16], as well as in other related disciplines such as text classification [3,8,12] and have been shown to be highly effective compared to state-of-the-art approaches, not only in terms of recognition rates but also in terms of runtime. In [4, Table 4] for example, the authors have shown that their compression-based authorship verification method performed very similar to the winning approach [1] of the *PAN 2015 Author Identification* task [14], where it only required 7 seconds instead of 21 hours.

Our approach has a number of benefits. First, it does not require the explicit definition, selection and/or extraction of features as these are implicitly handled by the compression model. Second, our approach does not rely on a threshold which is often mandatory to judge whether two documents are written by the same author. Third, our approach does not involve machine learning methods and, thus, also not requires hyperparameter optimization (which is typically needed for classification/recognition). Fourth, the approach does not even need a specific preprocessing regarding the documents, which further reduces its complexity.

2 Our approach

This section describes our approach, which is broken down into both scenarios *complete author clustering* and *authorship-link ranking*.

2.1 Task 1: Complete author clustering

Compressing distances: As mentioned in Section 1 we waive the usage of a traditional distance function and instead use a compression-based dissimilarity measure. Given this measure, we can determine the "nearness" between two documents. However, before we can use this measure we require a compressor to obtain the compressed representation of the documents. Here, we decided to use one of the most powerful available compres-

sor PPM[2] (*Prediction by Partial Matching*), which has been used excessively in various fields and domains and led to promising results. Once the documents are compressed via PPM, we apply a dissimilarity function to measure how (dis-)similar two documents are to each other. As a dissimilarity function we chose the CBC (*Compression-based Cosine*) measure, proposed by Sculley and Brodley [13], which is defined as:

$$\mathrm{CBC}(x, y) = 1 - \frac{C(x) + C(y) - C(xy)}{\sqrt{C(x)C(y)}}. \tag{1}$$

Here, x and y denote two documents, and xy their concatenation. With $C(\cdot)$ we denote the length of a compressed document, which aims to approximate its *Kolmogorov Complexity*. The resulting value is in the interval $[0; 1]$.

Clustering via k-Medoids: In the *PAN 2017 Author Clustering* task the simplification is taken that all documents are single-authored. In practice this is not very realistic as it can often occur that documents (or text fragments such as paragraphs, sentences or phrases) are authored by different authors. However, we take advantage of the fact that all documents within the PAN corpora are single-authored and chose a simple partitional clustering algorithm that generates disjoint clusters. As a clustering algorithm we decided to use k-Medoids (proposed by Kaufman and Rousseeuw [6]), which is strongly related to the well-known k-Means method. However, in k-Medoids each cluster is represented by one of the objects in the cluster (the *medoid*), while in k-Means each cluster is represented by the center of the cluster (the *mean*).

The most common realization of the k-Medoids clustering method is the PAM (*Partitioning Around Medoids*, [7]) algorithm, which we we slightly modified by using a compression-based dissimilarity measure rather than a distance function. The modified algorithm is given in Algorithm 1.

Measure the quality of the clustering. Since for each problem ρ the number of authors k is not known beforehand, a strategy is needed to measure the clustering quality, in order to determine the "optimal" k. Our strategy is based on the computation of silhouettes (proposed by Rousseeuw [11]). The idea is to perform n (= number of documents in a problem ρ) clustering iterations[3] which results in $n - 1$ clusterings $\mathbb{C}_2, \mathbb{C}_3, \mathbb{C}_n$ via k-Medoids and to pick the k for which the clustering $\mathbb{C}_k$ yields the maximum silhouette coefficient $S_{\mathbb{C}}$, defined as:

$$S_{\mathbb{C}} = \frac{1}{n_{\mathbb{C}}} \left(\sum_{\mathcal{D} \in \mathbb{C}} \mathrm{s}(\mathcal{D}) \right)$$

Here, The calculation of a silhouette value $\mathrm{s}(\mathcal{D})$ is calculated as follows:

[2] In fact we use the PPMd variant, implemented in the C# library *SharpCompress*, offered by Adam Hathcock available under https://github.com/adamhathcock/sharpcompress. As a concrete implementation we used *Michael Bone*'s port of *Dmitry Shkarin*'s PPMd Variant I Revision 1.

[3] Note that we skip the case $n = 1$, as we assume that for each problem ρ there are two or more corresponding authors.

Algorithm 1: k-Medoids, adapted to compression-based dissimilarity scores.

Input: Number of clusters: k; document collection $\mathbb{D} = \{\mathcal{D}_1, D_2, \ldots, D_n\}$; dissimilarity measure: $d(x, y)$

Output: A clustering comprising k clusters: $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$

/* 0.) Helper methods: */
/* A dissimilarity score between two documents, weighted by the sum of all dissimilarity scores between the first document and each other document within the collection: */

$$wd(\mathcal{D}_1, \mathcal{D}_2) = \frac{d(\mathcal{D}_1, \mathcal{D}_2)}{\sum_{\mathcal{D}' \in \mathbb{D}} d(\mathcal{D}_1, \mathcal{D}')};$$

/* The sum of weighted dissimilarity scores regarding a document: */

$$w(\mathcal{D}) = \sum_{\mathcal{D}' \in \mathbb{D}} wd(\mathcal{D}', \mathcal{D});$$

/* The sum of the minimum dissimilarity scores between each non-medoid **n** and a medoid **m**. Here, **N** refers to the set of all non-medoids, while **M** denotes the set of all medoids. */

$$\text{totalCost}(N, M) = \sum_{n \in N} \min_{m \in M} (d(n, m));$$

/* 1.) Initialize medoids: */

$\mathbb{D}_{\text{sorted}} \leftarrow \mathbb{D}$ sorted ascending by $w(\cdot)$;
$M \leftarrow$ first k elements of $\mathbb{D}_{\text{sorted}}$;

/* 2.) Minimize total cost by finding a more suitable medoid at each step. Repeat until cost cannot be further decreased. */

label *minimizeStep*:
foreach $m \in M$ **do**
  foreach $n \in \mathbb{D} \setminus M$ **do**
    $M' \leftarrow M \setminus \{m\} \cup \{n\}$;
    if $\text{totalCost}(\mathbb{D} \setminus M', M') < \text{totalCost}(\mathbb{D} \setminus M, M)$ **then**
      $M \leftarrow M'$;
      goto *minimizeStep*;

/* 3.) Assign non-medoids to their nearest medoids to form clusters. */

$\mathbb{C} \leftarrow \emptyset$;
foreach $m_i \in M$ **do**
  $\mathcal{C}_i \leftarrow \{\mathcal{D} \mid \mathcal{D} \in \mathbb{D} \wedge m_i = \arg\min_{m \in M}(d(\mathcal{D}, m))\}$;
  $\mathbb{C} \leftarrow \mathbb{C} \cup \{\mathcal{C}_i\}$;

return $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$;

1. Let $\mathrm{s}(\mathcal{D}) \in [-1;1]$ denote a silhouette value for a document $\mathcal{D} \in \mathbb{D}$, which was assigned to a cluster $\mathcal{C}_a$. We first compute $\mathrm{a}(\mathcal{D})$ = the average dissimilarity of $\mathcal{D}$ to all other documents in the same cluster $\mathcal{C}_a$.

2. For every other cluster $\mathcal{C} \neq \mathcal{C}_a$, we calculate the average dissimilarity $\mathrm{b}(d) = \mathrm{dist}(\mathcal{D}, \mathcal{C})$ between $\mathcal{D}$ and each document in $\mathcal{C}$. The cluster with the smallest average dissimilarity to $\mathcal{D}$ is denoted by $\mathcal{C}_b$.

3. Finally, we compute $\mathrm{s}(\mathcal{D})$ as follows: For the case that the initial cluster comprises only one document ($|\mathcal{C}_a| = 1$) or that $a = b$ holds, we set $\mathrm{s}(\mathcal{D}) = 0$. For the case that $\mathrm{a}(\mathcal{D}) < \mathrm{b}(\mathcal{D})$ we calculate $\mathrm{s}(\mathcal{D}) = 1 - \frac{\mathrm{a}(\mathcal{D})}{\mathrm{b}(\mathcal{D})}$ and otherwise $\mathrm{s}(\mathcal{D}) = \frac{\mathrm{b}(\mathcal{D})}{\mathrm{a}(\mathcal{D})} - 1$.

2.2 Task 2: Authorship-link ranking

In order to establish authorship-links within each cluster, we first modified the CBC measure in order to calculate similarity (instead of dissimilarity) scores as follows:

$$\mathrm{CBC}_{\mathrm{sim}}(x, y) = \frac{C(x) + C(y) - C(xy)}{\sqrt{C(x)C(y)}}. \tag{2}$$

Given $\mathrm{CBC}_{\mathrm{sim}}(x, y)$, we applied it on each document pair within a cluster and sorted the resulting list in a descending order. Note that the authorship-link ranking step could be also performed through an arbitrary authorship verification method. However, we tried to keep the approach as compact as possible. Therefore, we only made use of PPM to compress the documents and calculate their similarity to each other by using $\mathrm{CBC}_{\mathrm{sim}}(\cdot)$.

3 Evaluation

Since our approach does not require any type of training, there was no need to split the given training corpus into two sub-sets in order to apply hyperparameter learning on one set and the evaluation on the second set. Besides the PAN 2017 AC training corpus we also used the training corpus from PAN 2016[4]. The results regarding both corpora are listed in Tables 1-6.

3.1 PPM: Optional parametrization

As stated in this papers, our scheme does not require any type of training. However, this is only true, because we used a predefined (hard coded) parametrization regarding the PPM compressor within the involved C# library. In fact, there are two tweakable parameters (`AllocatorSize` and `ModelOrder`) that aim to improve the compression results. For `AllocatorSize`, we could not observe any influence regarding the author clustering results, irrespective of which values were used. Therefore, we waived

[4] Note that at the time this paper was written, the test corpus was not publicly released.

Table 1. PAN 2017 Author Clustering Training Dataset 2017-02-15 [English]

Problem	Language	Genre	F-Bcubed	R-Bcubed	P-Bcubed	Av-Precision
problem001	en	articles	0,47481	0,49444	0,45667	0,11564
problem002	en	articles	0,48596	0,52333	0,45357	0,065881
problem003	en	articles	0,4979	0,50606	0,49	0,041557
problem004	en	articles	0,6467	0,875	0,51288	0,25755
problem005	en	articles	0,42331	0,75	0,29487	0,047852
problem006	en	articles	0,44589	0,525	0,3875	0,095395
problem007	en	articles	0,56866	0,475	0,70833	0,16332
problem008	en	articles	0,53429	0,39444	0,82778	0,20633
problem009	en	articles	0,54495	0,48611	0,62	0,093589
problem010	en	articles	0,48862	0,34286	0,85	0,099382
problem011	en	reviews	0,63504	0,55192	0,74762	0,25705
problem012	en	reviews	0,52277	0,4	0,75429	0,137
problem013	en	reviews	0,47764	0,34619	0,77	0,033695
problem014	en	reviews	0,54136	0,57778	0,50926	0,044189
problem015	en	reviews	0,51064	0,34286	1	0,12121
problem016	en	reviews	0,7094	0,71048	0,70833	0,30976
problem017	en	reviews	0,72956	0,85333	0,63714	0,33655
problem018	en	reviews	0,60968	0,54	0,7	0,073084
problem019	en	reviews	0,52027	0,62619	0,445	0,063617
problem020	en	reviews	0,51891	0,60667	0,45333	0,023188
Average			**0,544318**	**0,546383**	**0,6163285**	**0,12929195**

to train an "optimal" value for this parameter and, instead, used the default setting of $2^{24} = 16,777,216$.
In contrast, we observed for `ModelOrder` slight variations regarding the author clustering results, during initial experiments. Hence, we applied our scheme on both training datasets (PAN 2016 and PAN 2017), in order to consider, if it make sense to discard training and, instead, to use the default parameter setting of 6 (in total there are 15 possible values, ranging from 2 to 16). As can be inferred from the results (given in Figure 1), the default parameter setting is very close to the average across all possible parameter settings. As a consequence, we decided to discard the training for this parameter and to use the default (hard-coded) setting.

3.2 Other experiments

Besides k-Medoids we also experimented with the density-based clustering method DBSCAN (*Density-Based Spatial Clustering of Applications with Noise.*), where we also used compression-based dissimilarity scores rather than distances. Our intention was to eliminate the determination of k, not only to reduce the approach's complexity, but also to save runtime as only one scan through the documents is needed. However, instead of the expected reduction it added more complexity as both density parameters ε (maximum radius of the neighborhood) and *minPts* (minimum number of points required to form a dense region) require training. In addition, it turned out that after training

Table 2. PAN 2017 Author Clustering Training Dataset 2017-02-15 [Dutch]

Problem	Language	Genre	F-Bcubed	R-Bcubed	P-Bcubed	Av-Precision
problem021	nl	articles	0,54377	0,55333	0,53452	0,093142
problem022	nl	articles	0,56078	0,91	0,40526	0,28237
problem023	nl	articles	0,44267	0,565	0,36389	0,020261
problem024	nl	articles	0,5565	0,39351	0,95	0,2424
problem025	nl	articles	0,65494	0,59722	0,725	0,15419
problem026	nl	articles	0,43757	0,33333	0,63667	0,079443
problem027	nl	articles	0,68961	0,56905	0,875	0,26102
problem028	nl	articles	0,68785	0,79	0,60909	0,16974
problem029	nl	articles	0,59828	0,48654	0,77667	0,16066
problem030	nl	articles	0,5784	0,41905	0,93333	0,19865
problem031	nl	reviews	0,6	0,5	0,75	0,12189
problem032	nl	reviews	0,51471	0,4375	0,625	0,03189
problem033	nl	reviews	0,46684	0,52778	0,41852	0,071347
problem034	nl	reviews	0,64865	0,75	0,57143	0,18214
problem035	nl	reviews	0,5916	0,55	0,64	0,12552
problem036	nl	reviews	0,54637	0,47222	0,64815	0,031937
problem037	nl	reviews	0,66009	0,675	0,64583	0,14429
problem038	nl	reviews	0,43555	0,61667	0,33667	0,075441
problem039	nl	reviews	0,40594	0,5	0,34167	0,013461
problem040	nl	reviews	0,49321	0,74167	0,36944	0,076857
Average			**0,5556665**	**0,5693935**	**0,607807**	**0,12683245**

DBSCAN still performed worse than k-Medoids on both training corpora PAN-2016 and PAN-2017. On average, DBSCAN achieved only 80% of k-Medoids' F-Bcubed scores. Therefore, we discarded this approach.

4 Conclusions

We proposed an experimental approach to cluster texts by their authors by using k-Medoids with compression-based dissimilarity scores. On the plus side, our approach is quite simple and entirely independent from feature engineering, threshold determination (regarding the authorship-link ranking sub-task), (text-) preprocessing as well as hyperparameter optimization. On the negative side, the proposed approach does not perform very well, which might have a number of reasons. We noticed for example (after the submission deadline of the software) that the compression-based dissimilarity measure does not fulfill even one of the required properties of a real distance-based metric, which are identity[5], symmetry[6] and triangle inequality. Especially the symmetry

[5] For example, when we compress a document x and apply $\mathrm{CBC}(x, x)$ we obtain as a dissimilarity measure the score : 0.117647. This value is somehow confusing as we might expect 0 when we are used to work with real distance metrics.

[6] For example, consider we have two different documents x and y. Computing $\mathrm{CBC}(x, y)$ returns 0.6459, while $\mathrm{CBC}(y, x)$ returns 0.6852.

Table 3. PAN 2017 Author Clustering Training Dataset 2017-02-15 [Greek]

Problem	**Language**	**Genre**	**F-Bcubed**	**R-Bcubed**	**P-Bcubed**	**Av-Precision**
problem041	gr	articles	0,42798	0,55667	0,34762	0,016841
problem042	gr	articles	0,49535	0,57083	0,4375	0,051153
problem043	gr	articles	0,52746	0,695	0,425	0,062329
problem044	gr	articles	0,47622	0,58833	0,4	0,047589
problem045	gr	articles	0,42076	0,43	0,4119	0,021957
problem046	gr	articles	0,34142	0,415	0,29	0,02013
problem047	gr	articles	0,65524	0,725	0,59773	0,16782
problem048	gr	articles	0,40556	0,485	0,34848	0,049585
problem049	gr	articles	0,44287	0,34	0,635	0,063383
problem050	gr	articles	0,47897	0,40833	0,57917	0,10167
problem051	gr	reviews	0,48119	0,5375	0,43556	0,07551
problem052	gr	reviews	0,41693	0,93333	0,26842	0,16368
problem053	gr	reviews	0,49617	0,3975	0,66	0,164
problem054	gr	reviews	0,46805	0,40889	0,54722	0,061924
problem055	gr	reviews	0,59742	0,72778	0,50667	0,23549
problem056	gr	reviews	0,53797	0,635	0,46667	0,075371
problem057	gr	reviews	0,49493	0,79167	0,36	0,097521
problem058	gr	reviews	0,67832	0,6125	0,76	0,19889
problem059	gr	reviews	0,80721	0,93333	0,71111	0,59345
problem060	gr	reviews	0,62222	0,7	0,56	0,13118
Average			**0,513612**	**0,594583**	**0,4874025**	**0,11997365**

Table 4. PAN 2016 Author Clustering Training Dataset 2016-02-17 [English]

Problem	**Language**	**Genre**	**F-Bcubed**	**R-Bcubed**	**P-Bcubed**	**Av-Precision**
problem001	en	articles	0,30386	0,82133	0,18641	0,009715
problem002	en	articles	0,42318	0,64302	0,31537	0,019366
problem003	en	articles	0,28383	0,96	0,16653	0,0090597
problem004	en	reviews	0,20851	0,76667	0,12067	0,0028351
problem005	en	reviews	0,17956	0,94583	0,099198	0,0039141
problem006	en	reviews	0,30265	0,67875	0,19474	0,010734
Average			**0,283598333**	**0,8026**	**0,180486333**	**0,00927065**

Table 5. PAN 2016 Author Clustering Training Dataset 2016-02-17 [Dutch]

Problem	**Language**	**Genre**	**F-Bcubed**	**R-Bcubed**	**P-Bcubed**	**Av-Precision**
problem007	nl	articles	0,36444	0,90643	0,22807	0,0014286
problem008	nl	articles	0,60859	0,62765	0,59064	0,042869
problem009	nl	articles	0,3755	0,80117	0,24522	0,013505
problem010	nl	reviews	0,3779	0,64833	0,26667	0,0087443
problem011	nl	reviews	0,25545	0,72	0,15527	0,0017606
problem012	nl	reviews	0,30055	0,91	0,18	0
Average			**0,380405**	**0,76893**	**0,277645**	**0,011384583**

Table 6. PAN 2016 Author Clustering Training Dataset 2016-02-17 [Greek]

Problem	Language	Genre	F-Bcubed	R-Bcubed	P-Bcubed	Av-Precision
problem013	gr	articles	0,29726	0,7	0,1887	0,017252
problem014	gr	articles	0,26916	0,88	0,15888	0,024111
problem015	gr	articles	0,21535	0,93939	0,12162	0,010769
problem016	gr	reviews	0,22652	0,93939	0,12879	0,0012533
problem017	gr	reviews	0,41019	0,86818	0,26853	0,033574
problem018	gr	reviews	0,34012	0,92727	0,20825	0,022682
Average			**0,2931**	**0,875705**	**0,179128333**	**0,01827355**

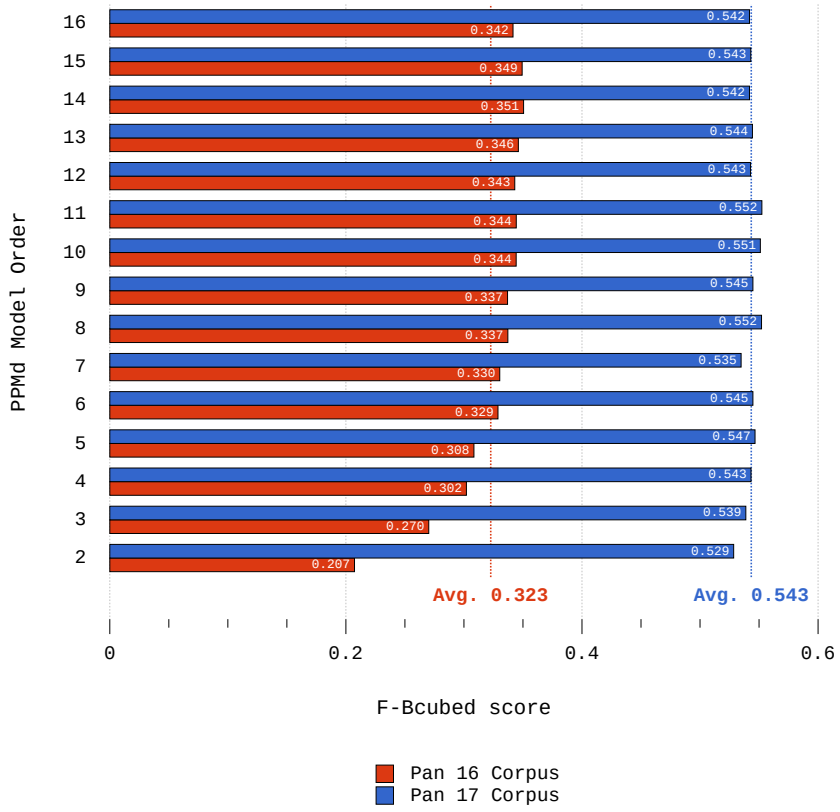

Figure 1. Author clustering results for the 15 different *ModelOrder* parameter settings.

property leads to an unexpected behavior, due to the implication that the order of the compressed documents matters when applying the compression-based dissimilarity on them. As future work we therefore need to examine for which cases compression-based models are applicable. Currently, we believe that they are well suited for establishing

authorship-link rankings, but for clustering alternative strategies might be more promising (and reliable).

Acknowledgments This work was supported by the German Federal Ministry of Education and Research (BMBF) in the funded project EWV (award number: **13N13500**).

References

1. Bagnall, D.: Author Identification Using Multi-headed Recurrent Neural Networks. In: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015. [14], http://ceur-ws.org/Vol-1391/150-CR.pdf
2. Cerra, D., Datcu, M., Reinartz, P.: Authorship Analysis Based on Data Compression. Pattern Recognition Letters 42, 79 – 84 (2014), http://www.sciencedirect.com/science/article/pii/S0167865514000336
3. Coutinho, D.P., Figueiredo, M.A.T.: Text Classification Using Compression-Based Dissimilarity Measures. IJPRAI 29(5) (2015), http://dx.doi.org/10.1142/S0218001415530043
4. Halvani, O., Winter, C., Graner, L.: Authorship Verification based on Compression-Models. ArXiv e-prints (Jun 2017)
5. Jr., W.O., Justino, E., Oliveira, L.: Comparing Compression Models for Authorship Attribution. Forensic Science International 228(1âĂŞ3), 100–104 (2013), http://www.sciencedirect.com/science/article/pii/S0379073813000923
6. Kaufman, L., Rousseeuw, P.J.: Clustering by Means of Medoids. Statistical Data Analysis Based on the L1-Norm and Related Methods pp. 405–416 (1987)
7. Kaufman, L., Rousseeuw, P.J.: Partitioning around Medoids (Program PAM). Finding Groups in Data: An Introduction to Cluster Analysis pp. 68–125 (1990)
8. Marton, Y., Wu, N., Hellerstein, L.: On Compression-Based Text Classification. In: Advances in Information Retrieval, 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005, Proceedings. pp. 300–314 (2005), http://dx.doi.org/10.1007/978-3-540-31865-1_22
9. Nagaprasad, S., Reddy, P.V., Babu, A.V.: Authorship Attribution based on Data Compression for Telugu Text. International Journal of Computer Applications 110(1), 1–5 (2015)
10. Potthast, M., Rangel, F., Tschuggnall, M., Stamatatos, E., Rosso, P., Stein, B.: Overview of PAN'17: Author Identification, Author Profiling, and Author Obfuscation. In: Jones, G., Lawless, S., Gonzalo, J., Kelly, L., Goeuriot, L., Mandl, T., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 8th International Conference of the CLEF Initiative (CLEF 17). Springer, Berlin Heidelberg New York (Sep 2017)
11. Rousseeuw, P.: Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. J. Comput. Appl. Math. 20(1), 53–65 (Nov 1987), http://dx.doi.org/10.1016/0377-0427(87)90125-7
12. Saikrishna, V., Dowe, D.L., Ray, S.: Statistical Compression-based Models for Text Classification. In: 2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS). pp. 1–6 (Dec 2016)
13. Sculley, D., Brodley, C.E.: Compression and Machine Learning: A New Perspective on Feature Space Vectors. In: 2006 Data Compression Conference (DCC 2006), 28-30 March 2006, Snowbird, UT, USA. pp. 332–332. IEEE Computer Society (2006), http://dx.doi.org/10.1109/DCC.2006.13

14. Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., Stein, B.: Overview of the Author Identification Task at PAN 2015. In: CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers. CEUR, CEUR, Toulouse, France (2015/09/10 2015)
15. Tschuggnall, M., Stamatatos, E., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes Papers of the CLEF 2017 Evaluation Labs
16. Veenman, C.J., Li, Z.: Authorship Verification with Compression Features. In: Forner, P., Navigli, R., Tufis, D., Ferro, N. (eds.) Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23–26, 2013. CEUR Workshop Proceedings, vol. 1179. CEUR-WS.org (2013), http://ceur-ws.org/Vol-1179/CLEF2013wn-PAN-VeenmanEt2013.pdf