Style Change Detection with Feed-forward Neural Networks

Notebook for PAN at CLEF 2019

Chaoyuan Zuo, Yu Zhao, and Ritwik Banerjee

Department of Computer Science
{chzuo,yuzhao1,rbanerjee}@cs.stonybrook.edu
Stony Brook University, Stony Brook New York 11794, USA

Abstract The majority of previous authorship attribution studies mainly focus on a dataset of documents (or parts of documents) with labeled authorship. This scenario, however, is not applicable to documents written by more than one author. Detecting the authorship switches within multi-author documents has been shown to be a challenging task in previous PAN tasks. A simplified version of the style change task is thus organized by PAN 2019, which aims at identifying the number of authors in a given document. To this end, we present a system consisting of two modules, one for distinguishing the single-author documents from the multi-author documents and the other for determining the exact number of authors in the multi-author documents.

1 Introduction

Authorship attribution is a difficult task with a long history [10]. With the advent of the web and social media, however, the nature of authorship is changing. On one hand, collaborative writing is becoming more commonplace, while on the other, the easy availability of vast amounts of source material makes plagiarism easy to carry out but hard to detect. There have been several settings of the authorship attribution task, with the most common scenario focusing on a closed set of documents and candidate authors with the assumption that each document is written by a single author from among the candidates [4,18]. These models are not applicable if a document is written by more than one author, however. The style breach detection task at PAN 2017 was designed to bridge this gap. The task was to find the border positions where authorship changes, but the results showed it to be an extremely challenging task [20]. A simplified version was presented in the following year at PAN, where the task was to detect whether or not there was any stylistic change, *i.e.*, whether or not the document had multiple authors. The results of this task were quite promising, attaining accuracy up to 0.893. The current task [22] goes further and aims at detecting the exact number of authors in a document.

This paper reports on the PAN 2019 shared task on style change detection. A two-step pipeline is presented to solve this problem. The first step of the system aims at

distinguishing multi-author documents from the documents with a single author, and the second identifies the exact number of authors within the multi-author document. The evaluation results of this task suggest that automated detection of writing style change remains a challenging task.

2 Related Work

Earlier research in author profiling worked under the assumption that each article has exactly one author [18] where authorship attribution could largely be done on the basis of lexical, syntactic, and semantic features [2,14]. Documents may have multiple authors, however. For instance, in collaborative work, different sections may be written by different authors. The stylistic cues linking authors to text can be used to partition a document into stylistic clusters, thus providing insight into the number of authors of a document, and who those authors might be. This is known as the author diarization problem, an important component of which is to identify the number of authors within a document.

Prior research on the author diarization task at PAN-2016 [19] indicates that capturing stylometric changes is perhaps the most promising approach on author clustering within documents [9,16]. The text is split into sentences, and features including word frequency, selected part-of-speech (POS) tag counts, and average word length are calculated for each sentence. The K-means clustering algorithm is applied to generate clusters based on the distances computed with these features [16]. The style breach detection task at PAN-2017 [20] is to find the exact position of authorship changing with a document. Here, too, the three submissions extract shallow stylometric features like character n-grams, word frequency, function words and punctuation. Such features are explored at the level of sentences [6,12] as well as paragraphs [5]. The similarity between the consecutive objects is then evaluated to detect a change of authorship.

Given the evident difficulty of author diarization, PAN-2018 presented a simpler binary classification task of identifying whether or not a document was written by a single author. Going beyond typical stylometric features [7,13], several other approaches were explored. The winning submission by Zlatkova et al. [23] split each document into three segments of equal length and use several classifiers to obtain the final results. Hosseinia and Mukherjee [3] relied solely on grammatical structures and lexical features where each sentence was represented by a collection of features extracted from its parse tree. This representation was provided as input to two recurrent neural networks in the original or reversed order of the sentences of the document, respectively. Multiple similarity measures were then computed the difference between the two network representations, yielding the final binary classification. Another approach was taken by Schaetti [15], based on a character embedding layer used in a convolutional neural network.

3 Style Change Detection

This paper is the result of our participation in the 'Style Change Detection' taks as part of PAN at CLEF 2019. The task is defined as follows: *Given a document, determine whether it contains style changes or not, i.e., if it was written by a single or multiple*

authors. If it is written by more than one author, determine the number of involved collaborators. As is evident from the task definition, an individual author is implicitly associated with a particular style of writing. Earlier PAN tasks have shown that complete author diarization is a particularly difficult problem [19,20]. As such, this task sets the relatively modest goal of detecting the number of authors in each document, omitting author attribution. The task definition lends itself naturally to a two-step pipeline process: (i) binary classification to determine whether a document has a single author or multiple authors, and (ii) if a document has multiple authors (as determined by the first step), identify how many. Next, we present the details of the data used in this work and the evaluation framework.

Data: Each document in the dataset for this task is a post (or a concatenation of multiple posts) from StackExchange[1]. The training set comprises 2,546 documents, and a separate validation set of 1,272 documents is also provided. For each document, the gold-standard labels are provided in the form of (i) the number of authors, and (ii) annotations marking who authored exactly which portions of the document. Exactly half of the training documents have a single author. The remaining half have a nearly uniform distribution over the number of authors, ranging from 2 to 5 authors. This is shown in Table 1.The test set has the same size as the validation set, but has been provided without any labels, *i.e.*, no information about where within a document authorship switched, or how many authors are there for a given document.

Table 1. Distribution of the number of authors of the documents in the dataset

# authors	1	2	3	4	5
# docs in training	1273	325	313	328	307
# docs in validation	636	179	152	160	145

Evaluation: In this task, the performance of a model is evaluated by combining (i) the accuracy, which serves as a measurement for the binary classification of single against multiple authorship, with (ii) the *ordinal classification index* (OCI) [1], which measures the error of predicting the number of authors for documents with multiple authors. Thus, the final rank r that captures both is the arithmetic mean of the accuracy and the inverted OCI is given by

$$r = \frac{1}{2}\left(\text{accuracy} + (1 - \text{OCI})\right) \tag{1}$$

To evaluate the submission, the participants are asked to submit the created software for this task through a virtual machine in TIRA [11], a web platform that supports software submission for shared tasks.

4 Methodology

Given the evaluation framework, which consists of combining two independent measurement, we build a system that treats them separately by applying a two-step pipeline. Th first step separartes single-author from multi-author documents, and the second step

[1] https://stackexchange.com/

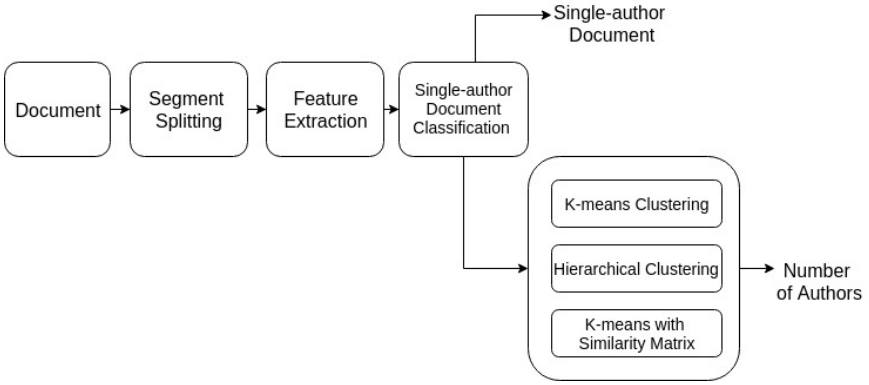

Figure 1. Overview of our system

identifies the number of authors in multi-author documents, In our approach, we divide the whole document into several segments, and cluster the segments based on writing style. The objective is to have the number of clusters be equal to the the number of authors in the document. Figure 1 illustrates this process[2].

Data preprocessing: About 1% of the documents in the dataset are in Spanish instead of English. Since it is difficult to extract features and build models for them separately within the scope of this task, we randomly assign the number of authors (ranging from 1 to 5) for these documents. For the rest, we filter out some frequent phrases that carry little or no linguistic relevance. These are typically *URL* or technical specifications like "OSX 10.11.2".

Binary classification of documents (single vs multiple authors): Here we describe how we identify whether a document is authored by one person or not. We treat all documents with multiple authors as one category, and build binary classifiers for separating them from documents with a single author. We use Keras[3] to implement this. Each documents is first tokenized and then converted into a term-document matrix where each word is encoded with its term-frequency inverse-document-frequency (TF-IDF) score. Further, we set the maximum number of words to keep as 40,000. Only the most common 40,000 words on the dataset are used, based on word frequency.

This classification is along the lines of the PAN-2018 task. Given the success of non-linear neural networks in that task (e.g., Zlatkova et al. [23]), we decide to adopt a neural network as well. We use a simple feedforward neural network as the classifier, with the term-document matrix being used as the input. The network has only one hidden layer with 128 units and a softmax output layer with two nodes for the binary classification. For activation, we use the sigmoid function since it achieved better results when compared to rectified linear units (ReLU) on validation, and for optimization we

[2] The implementation is available at https://github.com/chzuo/PAN_2019.

[3] https://keras.io/

use Adam [8]. To avoid overfitting, we use dropout [17] with the probability value set to $p = 0.5$.

Detecting the number of authors in multi-author documents: Our key idea for this step is to divide the whole document into several parts and then cluster them to groups. Since some documents are poorly structured (*e.g.), upon tokenizing using the NLTK tokenizer, some documents yield more than 200 sentences*, we divide each document into paragraph-level segments instead of sentences. Further, upon studying the test and validation sets, we notice that almost all authorship changes happen near an empty line or newline symbol, based on the observation that half of the switches happen after an empty line while 80% of the writing by new authors starts after the newline symbols. Thus, to divide a document into segments, firstly the empty line is used to separate the segments, and if we can get more than 15 segments in the document after splitting, we then use these segments for the next clustering step, otherwise, we use the newline symbol for splitting segments in the text and use that results for the next step. In general, using the newline symbol results in more segments than using an empty line. Finally, segments with less than 20 tokens are discarded.

Feature Extraction: Then the feature extraction module is conducted on each segment after splitting. Following the feature design of the winning submission of PAN-2018 [23], we use the following features:

- *Token and POS distribution features.* This includes the distribution of token length in the segment, the distribution of POS-tags in the segment, the number of sentences in the segment, the number of each special character and punctuation in the segment. The special characters included are '#', '$', '%', '&', '*', '@', parentheses and the forward and backward slashes.
- *Contracted word forms.* Writers may have their own preferences about the usage of contractions like "I'll" instead of "I will" or "I'm" instead of "I am". Thus, we maintain two lists, one containing the original words and the other containing their corresponding contracted forms. We count the total number of occurrences of the words in each list and use these two number as the feature.
- *British/American English spelling.* We use a list of spelling variations[4], and encode this feature as the number of occurrence of variant spellings in a single segment.
- *Function word frequencies.* We combine the list of function words from NLTK and the list used by Zlatkova et al. [23], and encode the frequency of each function words as additional features.
- *Readability.* We use Textstat[5] to obtain the Flesch reading ease score, SMOG grade, Flesch-Kincaid grade, Coleman-Liau index, automated readability index, Dale-Chall readability score, Linsear Write readability metric, Gunning-Fog index, and the number of difficult words in the text. We use all the above measures of readability and keep them as separate features.

Further, we also use the TF-IDF scores of the words in each segment, as are calculated for the creation of the term-document matrix earlier.

[4] https://en.wikipedia.org/wiki/Wikipedia:List_of_spelling_variants (Accessed May 2019).

[5] https://github.com/shivam5992/textstat

Segment Clustering: To cluster the segments into groups, we build an ensemble system of different algorithms on various combinations of features. It consists of three models: (i) the K-means clustering where each segment is represented as a bag-of-words vector with TF-IDF encoding, (ii) hierarchical clustering with all the rest of the features mentioned in the above section, and (iii) a feedforward neural network classifier to detect the similarity between segments and create the similarity matrix for all the segments using the output of the NN. Then we use K-means clustering with silhouette analysis to determine the number of clusters in this similarity matrix. The three models share the same weight when determining the final results of the number of authors.

K-means clustering: For each document, all the segments are represented as a bag-of-words vector with TF-IDF encoding. We use the scikit-learn [6] tool to implement the K-means clustering for these segments. The silhouette analysis is then used to select the number (ranging from 2 to 5) of best clusters.

Hierarchical clustering: The hierarchical clustering algorithm is used to group the segments in each document. We use the extracted features for clustering except for the TF-IDF encoding. The tool we use for the implementation of hierarchical clustering is the SciPy[7]. We use the Ward's minimum variance method [21] for computing the distance between the nodes . Once the distance between nodes has been calculated, the linkage function is used for paring the objects that are close to the binary clusters (clusters consisting of two objects), and it links the newly formed clusters to each other to create bigger clusters based on the distance information until all the nodes are linked in a hierarchical tree. The distance information of each clustering step in this tree can be used as the cutoff argument to determine the number of clusters. Moreover, the inconsistent coefficient for each link in the hierarchical clustering tree can be used for determining the number of clusters as well. It is calculated by comparing the height of a link with the average height of other links at the same cluster hierarchy. The lower the coefficient, the smaller the difference between the object with those around it.

The number of clusters for each document, however, is hard to determine, as there are no methods to set a universal cut-off value for all documents. Thus, we build a feedforward neural network with one hidden layer consisting of 20 units. The input vector with the dimension as 50 is created for each document, using the distance and inconsistency value for the last 25 clustering step on a linkage matrix. If the number of segments is less than 25 in the document, we add zeros to the start of the vectors to increase its length to 50. The output target of the NN classifier is the number of authors in the document. A softmax layer with 4 output nodes is added. We use the ReLU for activation function and Adam for stochastic optimization, as well as the dropout with $p = 0.5$.

We train the network on all the documents with more than two authors in the training and validation set. During the test phase, only the documents that get categorized as multi-author documents after the first step of our process (*i.e.*, the binary classification) is sent as the input to the NN model for prediction.

[6] https://scikit-learn.org/stable/

[7] https://www.scipy.org/

Team	Accuracy	OCI	Rank	Runtime
zuo19	0.604	**0.808**	0.397	**00:25:13**
nath19	**0.847**	0.865	**0.491**	02:45:13
Random Baseline	0.500	0.876	0.312	-

Table 2. Results for the submission from all participants in style change detection task. We participated under the name *zuo19*. The best results for each metric is shown in bold. For OCI, the lower the value, the better the performance.

K-means with similarity matrix We create a dataset by splitting the documents into several parts using the provided authors switches information from the gold-standard label and pairing the obtained segments. Over 40k segment pairs are selected from the documents. For half of the pairs, the two segments are written by the same author and we treat them as one category. We build a binary NN classifier for separating them with pairs written by different authors. We use all the features mentioned above except for the TF-IDF representation. The network has 2 hidden layers with 50 units and 8 units in them. We use ReLU for activation function. For each pair of segments, we define the similarity of this pair is the probability that the two segments are written by the same author. Then for a document containing n segments, we generate a similarity matrix M of size $n * n$ where $M_{i,j}$ is the similarity of segments i, j, using the output the NN classifier. Finally, we employ the K-means clustering method with silhouette analysis for determining the best number of authors in this similarity matrix.

5 Results

The results of our system are shown in Table 2. Two teams participated in this task, and our submission outperforms the other in the OCI evaluation measure, where the lower the value, the better the performance. For the first metric - accuracy, the performance of our binary classifier reaches the value as 0.6, 20 percent increase of 0.5, which serves as the baseline for a random guess. And for the second metric - OCI for multi-authors detection, our system achieves 0.808, slightly better than the random guess. The results of our system suggest that style change detection for multi-author documents is remains a challenging task and requires significant further research to be adequately resolved.

6 Conclusion

We develope a two-step pipeline system to detect the number of authors in the given document. The purpose of the first step is to identify whether or not the document is written by more than one author. This is achieved by using a feedforward neural network as a binary classifier. The second step is an ensemble model of different clustering methods. This identifies the number of authors for the multi-author documents. This task, however, is quite challenging, and there is scope for significant improvement in this direction.

References

1. Cardoso, J.S., Sousa, R.: Measuring the performance of ordinal classification. International Journal of Pattern Recognition and Artificial Intelligence 25(08), 1173–1195 (2011)
2. Feng, S., Banerjee, R., Choi, Y.: Characterizing Stylistic Elements in Syntactic Structure. In: Proceedings of the 2012 Joint Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning. pp. 1522–1533. Association for Computational Linguistics (2012)
3. Hosseinia, M., Mukherjee, A.: A Parallel Hierarchical Attention Network for Style Change Detection — Notebook for PAN at CLEF 2018. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, 10-14 September, Avignon, France. CEUR-WS.org (2018)
4. Iqbal, F., Binsalleeh, H., Fung, B.C., Debbabi, M.: Mining writeprints from anonymous e-mails for forensic investigation. digital investigation 7(1-2), 56–64 (2010)
5. Karaś, D., Śpiewak, M., Sobecki, P.: OPI-JSA at CLEF 2017: Author Clustering and Style Breach Detection—Notebook for PAN at CLEF 2017. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland. CEUR-WS.org (Sep 2017)
6. Khan, J.: Style Breach Detection: An Unsupervised Detection Model—Notebook for PAN at CLEF 2017. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland. CEUR-WS.org (Sep 2017)
7. Khan, J.A.: A model for style breach detection at a glance: Notebook for PAN at CLEF 2018. In: Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. (2018)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Kuznetsov, M.P., Motrenko, A., Kuznetsova, R., Strijov, V.V.: Methods for intrinsic plagiarism detection and author diarization. In: Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016. pp. 912–919 (2016)
10. Mendenhall, T.C.: The characteristic curves of composition. Science 9(214), 237–249 (1887)
11. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF. Springer (2019)
12. Safin, K., Kuznetsova, R.: Style Breach Detection with Neural Sentence Embeddings—Notebook for PAN at CLEF 2017. In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland. CEUR-WS.org (Sep 2017)
13. Safin, K., Ogaltsov, A.: Detecting a change of style using text statistics: Notebook for PAN at CLEF 2018. In: Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. CEUR-WS.org (2018)
14. Sapkota, U., Bethard, S., Montes, M., Solorio, T.: Not All Character N-grams Are Created Equal: A Study in Authorship Attribution. In: Proceedings of the 2015 Conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 93–102 (2015)
15. Schaetti, N.: Character-based Convolutional Neural Network and ResNet18 for Twitter Author Profiling — Notebook for PAN at CLEF 2018. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, 10-14 September, Avignon, France. CEUR-WS.org (2018)

16. Sittar, A., Iqbal, H., Nawab, R.: Author Diarization Using Cluster-Distance Approach—Notebook for PAN at CLEF 2016. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (eds.) CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal. CEUR-WS.org (Sep 2016)
17. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)
18. Stamatatos, E.: A Survey of Modern Authorship Attribution Methods. Journal of the American Society for Information Science and Technology 60(3), 538–556 (2009)
19. Stamatatos, E., Tschnuggnall, 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/Balog, Krisztian [edit.]; et al. pp. 691–715 (2016)
20. Tschuggnall, M., Stamatatos, E., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Overview of the author identification task at pan-2017: style breach detection and author clustering. In: Working Notes Papers of the CLEF 2017 Evaluation Labs/Cappellato, Linda [edit.]; et al. pp. 1–22 (2017)
21. Ward Jr, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American statistical association 58(301), 236–244 (1963)
22. Zangerle, E., Tschuggnall, M., Specht, G., Potthast, M., Stein, B.: Overview of the Style Change Detection Task at PAN 2019. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019)
23. Zlatkova, D., Kopev, D., Mitov, K., Atanasov, A., Hardalov, M., Koychev, I., Nakov, P.: An Ensemble-Rich Multi-Aspect Approach for Robust Style Change Detection — Notebook for PAN at CLEF 2018. In: Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L. (eds.) CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, 10-14 September, Avignon, France. CEUR-WS.org (2018)