Authorship Verification Machine Learning Methods For Style Change Detection In Texts

Notebook for PAN at CLEF 2023

Gianni X. Jacobo[1], Valeria Dehesa-Corona[2], Ariel D. Rojas-Reyes[2] and Helena Gómez-Adorno[3]

[1]*Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México*

[2]*Facultad de Ciencias, Universidad Nacional Autónoma de México, México*

[3]*Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, México*

Abstract

Style Change Detection refers to identifying segments within a multi-author document where the authorship may differ [1]. The PAN@CLEF 2023 Multi-Author Writing Style Analysis shared task involves addressing this task using a collection of Reddit website comments that are both cross-topic and open-set. In this paper, we explore the efficacy of various authorship verification methods in tackling this challenge and analyze their performance on the specified task.

Keywords

Style change detection, Support vector machine, Text compression, Logistic regression

1. Introduction

With the consolidation of the Internet and mobile technology, many texts have become more accessible. Alongside its diffusion, the concern about its protection has also increased. As more people access those texts, the probability of somebody who did not participate in creating them without giving correct credit authorship increases. By using Style Change detection on texts, we can get a good approach for detecting plagiarism [2]. However, it is difficult to check every existing text manually. In response, there are computer applications today that can perform that activity, but there is still room for better performance.

We implemented two methods for our PAN 2023 challenge submission. PAN is a workshop series and a networking initiative for stylometry and digital text forensics. PAN has included shared tasks on specific computational challenges related to authorship analysis, computational ethics, and determining the originality of a piece of writing [3]. In the Multi-Author Writing Style Analysis challenge proposed [3], it is required to solve the intrinsic style change detection task: for any given text, find all positions of writing style change on the paragraph-level [4]. The detection of these changes is proposed in three different complexities: Task 1 aims at identifying

CLEF 2023: Conference and Labs of the Evaluation Forum, September 5–8, 2021, Thessaloniki, Greece

ing.gianx@gmail.com (G. X. Jacobo); valedehesa@ciencias.unam.mx (V. Dehesa-Corona); damianrojas@ciencias.unam.mx (A. D. Rojas-Reyes); helena.gomez@iimas.unam.mx (H. Gómez-Adorno)

the author change for each pair of paragraphs in a document (if it exists), noticing paragraphs can cover different unrelated topics. Task 2 has the same objective, but the variation of topics between paragraphs is decreased. Finally, in Task 3, all the paragraphs in a document have the same topic.

Style change detection can be seen as one of the steps for Authorship Verification [2]. For Tasks 1 and 2, we evaluated with a method based on a text compression technique. These methods (grounded in information theory) have been tested previously and have obtained enough solid results [5]. We also used a different method based on the term-document matrix as a baseline for further comparison.

The rest of the paper is organized as follows. Section 2 shows details about the dataset provided by PAN organizers. After that, Section 3 outlines our methodology for addressing the style change detection task. This section covers the pre-processing steps, feature extraction techniques, and the machine learning algorithms employed. Section 4 presents and discusses the results obtained from our experiments. Finally, Section 5, draws conclusions based on our findings and discusses the results' implications.

2. Data

For this year 2023 and the Multi-Author Writing Style Analysis challenge, PAN organizers provided a dataset for each of the three tasks mentioned. They are divided into training and validation sets. Each dataset contains unique and non-repeated text files (4500 for training the proposed models, 900 for validating them). The text files have paragraphs; their source is some of the Reddit website user's comments.

Also, the datasets have ground truth files (hereafter dataset1, dataset2, and dataset3 for each task). The content of those files is a list of labels corresponding to the consecutive pairs of paragraphs for each text file. These labels have the following logic: if the same user wrote the pair of paragraphs, the label is a number 0. If not, the label is a number 1.

The proposed models must create, for each problem document, a JSON file containing a list of numbers one and zero. The logic of these numbers is the same as the truth ground truth files described previously.

3. Methodology

We worked on the presented tasks by treating them as Authorship Verification problems. We employed supervised learning algorithms such as Support Vector Machines and Logistic Regression [6], and utilized various text representations such as term-document matrix and cross-entropy vectors obtained through a text compression technique. Considering the labels in the ground truth files, we approached this problem as a binary classification problem for each pair of paragraphs.

3.1. Pre-processing

We extracted consecutive pairs of paragraphs from all the text files to enhance our analysis of the training set of each dataset. These pairs were then written into a separate temporary text file, allowing for isolated examination. The corresponding labels (one or zero) were recorded in another temporary file.

For the text compression method, we did not perform any modifications on texts. However, we converted the whole text to lowercase for the term-document method and removed all the characters not included in ASCII codification. These methods are explained in the next section.

3.2. Feature Extraction

We utilized two vectorization algorithms to identify style changes: Prediction by Partial Matching (PPM) and term-document matrix.

Term-document matrix is a way to represent texts as a numeric matrix [7]. In our case, the terms (words in lowercase) were extracted from the files of each dataset. Each column of the matrix represents a term. The columns contain the number of repetitions of that word in each document (in our case, one document per row).

Prediction by Partial Matching (PPM) is a text compression technique that predicts the next symbol in a sequence based on previous symbols [8]. PPM assigns probability values to each symbol based on the context it's observed in, with the lower value assigning the symbols more likely to occur and the higher values assigning the symbols less likely to occur. This algorithm provides a vector for each pair of documents based on their difference (crossed-entropy), which we can use to create a matrix as an input for our models. Most of the code we used was developed by Potthast $et\ al$ as a baseline for the Authorship Verification challenge in PAN 2021 [9].

3.3. Machine Learning

For classifying between the same author's paragraph or not, we will use the following machine learning methods:

Support Vector Machine (SVM) is a supervised learning algorithm for classification and regression tasks. In SVM, the objective is to find an optimal hyperplane that separates data points into different classes while maximizing the margin between them. The margin is the distance between the hyperplane and the nearest data points from each class [10]. The SVM method uses a kernel function to transform the input data into a higher-dimensional space. This allows for a linear decision boundary in the transformed space, even when the data is not linearly separable in the original feature space.

Logistic Regression (LR) estimates the probability of an event or a binary outcome. The logistic regression algorithm uses the logistic function (also known as the sigmoid function) to map the linear combination of the input features and their respective weights to a value between 0 and 1. This value represents the predicted probability of the positive class.

During the training process, logistic regression adjusts the values of the input features to minimize the error between the predicted probabilities and the actual class labels in the training data [11].

Once our data was prepared, involving the output of PPM and the text converted into a term-document matrix, we trained the machine learning methods mentioned earlier. We aimed to evaluate and select the best model for each specific task. By training these models, we aimed to optimize their performance and ensure they were well-suited for the given tasks.

4. Results

Through an extensive series of experiments, we conducted an in-depth analysis to identify the most effective approach for each dataset in Task 1, Task 2, and Task 3. After these evaluations, we found that applying the PPM technique to the texts and utilizing logistic regression yielded the highest F_1 score for Task 1 and Task 2 datasets. Conversely, for Task 3, we achieved the highest F_1 score by creating a term-document matrix (TD) and training the SVM algorithm. These findings are summarized in Table 1, providing a comprehensive overview of the performance of each approach across the different tasks and datasets. The table presents the pair of vectorization methods and machine learning algorithms to show the F_1 scores obtained by each one during our validation experiments.

Table 1
F_1 scores in validation set for each task, and features extraction mode - machine learning method

Dataset	TD, SVM	TD, LR	PPM, SVM	PPM, LR
Task 1	0.3769	0.2801	0.082	**0.7896**
Task 2	0.4741	0.4695	0.3967	**0.5324**
Task 3	**0.4944**	0.4940	0.3928	0.3524

Based on these scores, we improved the summarized results of our experiments by doing the following steps: for Tasks 1 and 2, we modified the threshold for LR's output value to achieve a better F_1 score (see Table 2). This is done by modifying a parameter we named radius *r*, a value that defines a range of values where the prediction probability of LR output is assumed as true (same author between paragraphs). We noticed the value *r* performs better when is 0.1 for Task 1.

Table 2
Text Compression F_1 scores

	$r = 0$	$r = 0.1$
Task 1	-	0.8080
Task 2	0.5814	0.4572

For Task 3, our original proposed term-document matrix (terms composed of uni-grams, bi-grams, and tri-grams, and 5000 of them as maximum) was changed until the best scores

were achieved. The matrix was finally determined by n-grams of only one *(1,1)* single word (*analyzer*), and the columns were created considering a minimum word occurrence in texts of 1 (*min-df*). Essentially, a traditional bag of words. Some experiments are presented in Table 3,

Table 3
Support Vector Classifier F_1 scores

	$n-gram=(1,1), min-df=1, analyzer=word$	$n-gram=(1,3), min-df=2, analyzer=char$
Task 1	-	0.3769
Task 2	0.4740	-
Task 3	0.4940	0.4928

Finally, we could evaluate our final models in the test set on the TIRA platform [12]: the scores are shown in Table 4. These results are very similar to the obtained scores in validation set. It demonstrates that our results were congruent to the used logic in our feature extraction methods and classification algorithms, based on the challenge description that explains that the nature of training, validation and test sets is the same.

Table 4
F_1 scores of the predictions in validation and test sets

Task	Feature extraction	Machine learning method	Validation set $f1$ score	Test set $f1$ score
Task 1	PPM	LR	0.8086	0.7930
Task 2	PPM	LR	0.5819	0.5907
Task 3	TD	SVM	0.4969	0.4978

5. Conclusion

This paper looks into the effectiveness of a text compression technique and term-document matrix transform methods for Style Change Detection tasks. The best scores for Task 1 and Task 2 were obtained using Text Compression, with f_1 scores of 0.8080 and 0.5819, respectively. For Task 3, the best f_1 score was 0.4978, obtained using Support Vector Classifier.

These results may be explained as follows: PPM is a robust method that could separate the paragraphs because the topics between them had enough stylistic features, mainly by the topic changes. But the technique lost its effectiveness when the topic was the same for the whole text of the document (Task 3). In Task 3, the term-document matrix was insufficient to get an F1 score higher than 0.50. It's possible that adding more representative features to this matrix can improve this input for the same or another IA classification method.

With these scores, we can compare and continue working to fix, improve and change many features used in our methodology for future writing style change detection challenges.

Acknowledgments

This work has been carried out with the support of DGAPA-UNAM PAPIIT project number TA101722. The authors also thank CONACYT for the computing resources provided through the Deep Learning Platform for Language Technologies of the INAOE Supercomputing Laboratory. We also want to thank Eng. Roman Osorio for supporting the student administration of the project.

References

[1] E. Zangerle, M. Mayerl, G. Specht, M. Potthast, B. Stein, Overview of the style change detection task at pan 2020., CLEF (Working Notes) 93 (2020).

[2] Z. Zhang, Z. Han, L. Kong, X. Miao, Z. Peng, J. Zeng, H. Cao, J. Zhang, Z. Xiao, X. Peng, Style change detection based on writing style similarity, Training 11 (1970) 17–051.

[3] J. Bevendorff, I. Borrego-Obrador, M. Chinea-Ríos, M. Franco-Salvador, M. Fröbe, A. Heini, K. Kredens, M. Mayerl, P. Pęzik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, A. G. Stefanos Vrochidis, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science, Springer, 2023.

[4] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Overview of the Multi-Author Writing Style Analysis Task at PAN 2023, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, CEUR-WS, 2023.

[5] W. J. Teahan, D. J. Harper, Using compression-based language models for text categorization, Language modeling for information retrieval (2003) 141–165.

[6] C. N. Kamath, S. S. Bukhari, A. Dengel, Comparative study between traditional machine learning and deep learning approaches for text classification, in: Proceedings of the ACM Symposium on Document Engineering 2018, 2018, pp. 1–11.

[7] M. Anandarajan, C. Hill, T. Nolan, Term-document representation, Practical Text Analytics: Maximizing the Value of Text Data, Springer, 2019, pp. 61–73.

[8] A. Moffat, Implementing the ppm data compression scheme, IEEE Transactions on Communications 38 (1990) 1917–1921.

[9] M. Potthast, S. Braun, T. Buz, F. Duffhauss, F. Friedrich, J. M. Gülzow, J. Köhler, W. Lötzsch, F. Müller, M. E. Müller, et al., Who wrote the web? revisiting influential author identification research applicable to information retrieval, in: Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38, Springer, 2016, pp. 393–407.

[10] R. Gholami, N. Fakhari, Chapter 27 - support vector machine: Principles, parameters, and

applications, in: P. Samui, S. Sekhar, V. E. Balas (Eds.), Handbook of Neural Computation, Academic Press, 2017, pp. 515–535.
[11] M. Kuhn, K. Johnson, Applied Predictive Modeling, Springer, 2018.
[12] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241.