A Content Spectral-Based Analysis for Authorship Verification

Notebook for PAN at CLEF 2022

Melesio Crespo-Sanchez[1], Helena Gómez-Adorno[2], Ivan Lopez-Arevalo[1], Edwin Aldana-Bobadilla[1], Karla Salas-Jimenez[3] and Jorge Cortes-Lopez[3]

[1] *Centro de Investigación y de Estudios Avanzados del I.P.N. Unidad Tamaulipas, Victoria 87130, México.*
[2] *Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, UNAM, Ciudad de México 04510, México.*
[3] *Facultad de Ciencias, UNAM, Ciudad de México 04510, México.*

Abstract

Authorship verification aims at determining if the same author produced a given a pair of texts. This task involves analyzing the documents' essential features, such as the used vocabulary, i.e., the lexical content; the syntactic content reflected by how the author makes combinations of the different words in such vocabulary following grammar rules; and the semantic content of the documents. This work presents a content spectral-based analysis approach, using neural network techniques for the authorship verification task at PAN at CLEF 2022.

Keywords

authorship verification, content spectral-based analysis, neural networks, PAN at CLEF

1. Introduction

PAN[1] at CLEF[2] is a series of scientific events and shared tasks on digital text forensics and stylometry, such as Style Change Detection, Profiling Irony, Stereotype Spreaders on Twitter, and Authorship Verification [1]. In this paper, we present an approach for the authorship verification task at PAN 2022 [2], which description is as follows: *given a pair of texts belonging to different discourse types, the challenge is to determine if the same author wrote these or not.* From a machine learning perspective, this is a binary classification task.

Nowadays we can find different approaches to solve the authorship verification task like machine learning-based [3], distance-based [4], or deep learning-based approaches [5]. A common factor between all these techniques is the text representation in the vector space, where each vector abstracts relevant features of the text. In this space, we can find clusters of

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2021, Bologna, Italy

melesio.crespo@cinvestav.mx (M. Crespo-Sanchez); helena.gomez@iimas.unam.mx (H. Gómez-Adorno); ilopez@cinvestav.mx (I. Lopez-Arevalo); edwyn.aldana@cinvestav.mx (E. Aldana-Bobadilla); karla_dsj@ciencias.unam.mx (K. Salas-Jimenez); kokofrank@ciencias.unam.mx (J. Cortes-Lopez)
https://github.com/helenpy (H. Gómez-Adorno)
0000-0001-5688-5352 (M. Crespo-Sanchez); 0000-0002-6966-9912 (H. Gómez-Adorno); 0000-0002-7464-8438 (I. Lopez-Arevalo); 0000-0001-8315-1813 (E. Aldana-Bobadilla)

[1] https://pan.webis.de/
[2] https://clef2022.clef-initiative.eu/

vectors that may denote certain classes of objects with similar features. Text representations should consider lexical, syntactic, and semantic components [6]. Lexical component associated with the used vocabulary in the texts. Syntactic component associated with how words structure a text that may denote a writing style (relevant for authorship verification). Semantic component associated with the main idea conveyed in a text.

In this work, we tackle the authorship verification task by transforming texts into a content spectral-based representation [7], which takes into account the previously mentioned text components used by a machine learning algorithm for detecting if pairs of documents belong to the same author.

The rest of this paper is structured as follows. Section 2 synthesizes some works related to our approach. Section 3 describes the dataset used for the task and presents the training, validation, and test partitions we generated to train the models. Section 4 describes the methodology for data transformation and classification. In section 5, the experiment configuration and results are reported. Section 6 finishes the paper and gives future work direction.

2. Previous Work

Authorship verification is still an area of exploration and development of strategies that evolve between the different problems and results [8]. Deep learning approaches have been used to solve authorship verification problems in recent years. Some examples of these works are the well-known pre-trained models such as transformers [9, 10, 11]. In PAN 2020, [5] introduced a Siamese network that learns the difference between two documents with a fully connected layer using n-grams and a residual network.

Other approaches use stylometric features, and lexical, syntactic, and semantic features as input for classical machine learning algorithms [12]. Sometimes is better to perform manual feature extraction to train some classification models. In this work, we propose to explore features from different linguistic levels of the language description lexical, syntactic, and semantic and represent the documents in a combined vector space.

3. Dataset

For this shared task, the dataset to be analyzed is the one provided by the PAN[3]. The following subsections review the dataset and explain the division of the dataset in training, validation, and test partitions that we use to train and evaluate our authorship verification model.

3.1. Data Review

The PAN 2022 authorship verification shared task organizers provided a training dataset containing $12,264$ instances of problems. Each problem corresponds to a pair of authors' texts, the discourse types, and a label that identifies if the same author wrote the texts. Below, we show an example of an instance's structure:

[3] https://pan.webis.de/data.html

```
{"id": "instance id", "discourse_type": ["essay", "email"], "pair": ["Text 1...", "Text 2..."]}
```

We identified a total of 1046 unique documents. The unique discourse type classes identified for these texts were *email, essay, memo,* and *text_message*. The distribution of discourse-type classes for these texts is shown in Table 1.

Table 1
Distribution of texts per discourse type.

Discourse type	Total texts
email	507
essay	93
memo	56
text_message	390

We identified a total of 56 unique authors in the training dataset, with a balanced number of instances, 6132 positive (documents written by the same author) and 6132 negative (documents written by different authors).

3.2. Dataset Partitions

After analyzing the training dataset provided by the organizers, we divided it into three partitions: 60% for training, 10% for validation, and 30% for testing, to train and evaluate our neural network approach. With the aim of training and evaluating our models on different authors, the partitioning needs to be author disjoint, *i.e.*, the intersection of authors on the training, validation, and testing sets should be empty.

To accomplish this, we ordered the authors by the number of texts written to include authors with the highest number of texts in the training partition. Then we deleted pairs with an author in different partitions (about 5514 instances). Also, to solve the problem of small partitions, we removed texts with less than 500 tokens. Table 2 shows the distribution of the obtained partitions. Nevertheless, these partitions are unbalanced, which could cause a bias towards one of the two classes.

Table 2
Initial partition distribution.

Partition	Total instances	Positive instances	Negative instances
Train	5889	5469	420
Validation	287	274	13
Test	411	389	22

Additionally, we added new instances of document pairs written by the same author (positive) and different authors (negative) to balance the partitions. For this, we applied the next process: Let A and B be the subsets of unique documents by each author from the corpus. Positive (P) and negative (N) instances were obtained via cartesian product, $P = A \times A$ and $N = A \times B$, respectively. Then, we randomly selected positive and negative instances from P and N sets, respectively, to balance the training, validation, and test partitions. Table 3 shows the distribution of the final partitions with the total number of instances, how many of these are positive, and how many are negative cases.

Table 3
Final partition distribution.

Partition	Total instances	Positive instances	Negative instances
Train	15,732	7866	7866
Validation	754	377	377
Test	1070	535	535

4. Methodology

Several techniques can be useful in the machine learning domain to solve the authorship verification problem. Among this repertoire of techniques, artificial neural networks are some of the most popular [13]. In this work, we used a *multilayer perceptron network* as a classification algorithm to determine if each pair of texts belong to the same author or not.

An adequate data transformation is required to train a machine learning model, i.e., obtain a vector space representation of the objects in the training set. Such a representation must hold features that allow the machine learning algorithm to perform a category separation in the vector space. Each category must include elements that share common characteristics, denoting the problem's classes of interest. Given the data type in this problem (text) and its nature, we assume that the representation of the different texts in the dataset must abstract elements such as vocabulary, writing style, and main idea in the content. These elements fall on the text's lexical, syntactic, and semantic components.

For this reason, we opted to use the spectral text representation proposed in [7] as a data transformation method. To obtain a representation of each document in the dataset, we identified a set of unique texts in the PAN dataset. From now on, we refer to this set as D. Figure 1 illustrates the used transformation method, composed of four main stages. Each stage is described below.

4.1. Text Pre-Processing Stage

The pre-processing techniques applied to D depend on the layer we work on (lexical, syntactical, or semantic layer). Unlike what is mentioned in [7], the only common pre-processing tasks applied to D for all layers are converting all texts to lowercase and tokenizing them for all layers. We did not remove any stopwords or punctuation symbols to preserve as much information as possible. We also preserved labels with named entities in the form of <entity> (e.g. streets, people, organizations, etc.) in the tokenization process. The syntactic layer applies a part-of-speech tagging (POS tagging) process. From this stage, we obtained three new versions of D: D_{lex}, D_{syn}, and D_{sem} respectively, one for each text component.

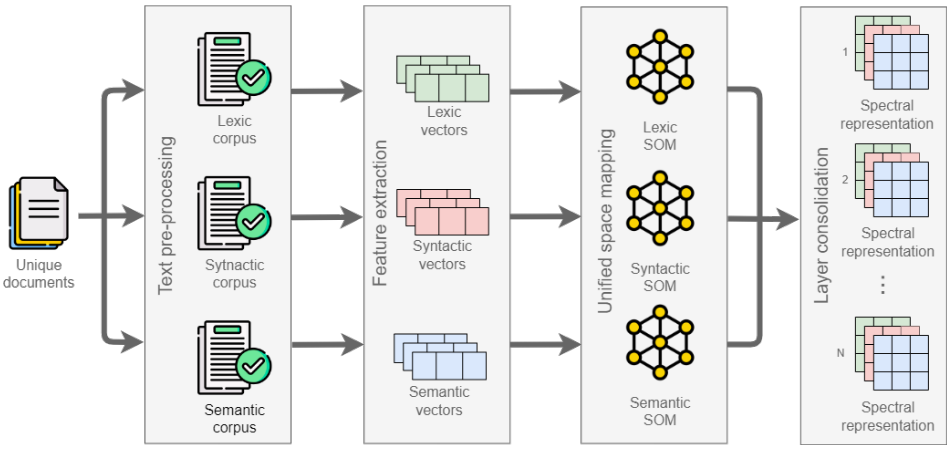

Figure 1: Spectral text representation method.

4.2. Feature Extraction Stage

Given the pre-processed sets D_{lex}, D_{syn}, and D_{sem}, at this stage, the aim is to extract feature vectors, one per text document for each layer. That said, we obtain three feature vectors for the same text ($\vec{x}_{lex}$, $\vec{x}_{syn}$, and $\vec{x}_{sem}$) corresponding to each of the text components. Each type of vector must contain features related to each component in question. For this, the extraction of each of these types of vectors is as follows:

- **Lexical layer:** For each text in D_{lex}, we obtain a vector $\vec{x} = [I(w_0), I(w_1), \ldots, I(w_j)]$, where each column in it corresponds to a word in the vocabulary. The value assigned to each element of $\vec{x}$ is given by Shannon information content of an outcome w_j [14], as described in Equation 1:

$$I(w_j) = -log_2(p(w_j)) \tag{1}$$

where $I(wj)$ is the amount of information that w_j gives to the text, and w_j is a word in the vocabulary. With this approach, those infrequent words are emphazied according to the amount of information they contribute to the text. In this sense, we consider all words, including punctuation marks and stop words, as part of the vocabulary. We obtain the set of feature vectors X_{lex} from this process.

- **Syntactic layer:** To extract syntactic features we use the well-known *Doc2Vec* algorithm [15]. Although this algorithm is commonly used to extract semantic content vectors, in the pre-processing stage, a POS tagging process was applied to D_{syn} to obtain POS tag sequences from the original text. In this way, the Doc2Vec algorithm is expected to capture syntactic information about the content rather than semantics that can denote a writing style for a given author. Then, we obtain the set of vectors X_{syn} from D_{syn}.

- **Semantic layer:** In this layer, we want to obtain feature vectors from D_{sem} that capture semantic information. Given this, we resort to the *Doc2Vec* algorithm once again to extract the corresponding feature vectors set X_{sem} for this text component.

4.3. Unified Space Mapping Stage

The sets of extracted feature vectors X_{lex}, X_{syn}, and X_{sem}, can have a different number of dimensions. At this stage, we make use of the *Self-Organizing Maps* (SOM) [16] to transfer vectors with a different number of dimensions to a space with the same dimensions where their similarity is preserved. This type of neural network is well known for mimicking the distribution of its training feature vectors. A SOM has a single output layer known as lattice, typically denoting a two-dimensional matrix. Each neuron in this lattice has a vector of weights associated with the same number of dimensions as its training vectors. For this purpose, we train a SOM model for each set of vectors X_{lex}, X_{syn}, and X_{sem}. We use the trained SOM models to obtain projections of the X vectors by applying the activation function described in Equation 2 between each feature vector $\vec{x}$ and each of the weights vectors $\vec{w}$ associated with every neuron of the SOM output lattice.

$$f(\vec{x}, \vec{w}) = \frac{1}{(\sum_{i=0}^{n} |x_i - w_i|^2)^{\frac{1}{2}}} \tag{2}$$

where $\vec{x}$ is a feature vector of a given text in its corresponding component layer, $\vec{w}$ is the vector of weights for a given neuron in the SOM lattice, and n is the number of dimensions of $\vec{x}$.

By applying the above activation function, we generate a feature matrix on the SOM lattice that denotes a specter of the text content in a unified space, one specter for each of the three text components per document.

4.4. Layer Consolidation

This final stage of the text transformation consists only of taking the three spectra of each text to consolidate them into a single three-layer text representation containing lexical, syntactical, and semantic features about the content.

4.5. Authorship Verification

Our proposal to address the authorship verification problem is illustrated in Figure 2. It consists of transforming each text in the dataset to its corresponding spectral representation. For each pair of texts in each problem instance, a subtraction is performed between the layers of each spectrum. Note that the subtraction operation is not commutative, and thus, the sign of the dissimilarity among documents can change depending on the order of them ($A - B$ or $B - A$). Our proposal showed better results when the sign or direction of such dissimilarity was considered. Once calculated $A - B$, the resulting matrices are flattened to form a single vector of features that feeds a multilayer perceptron neural network which determines whether the content in these texts shares lexical, syntactic, and semantic features in common that may denote a similarity in their authorship.

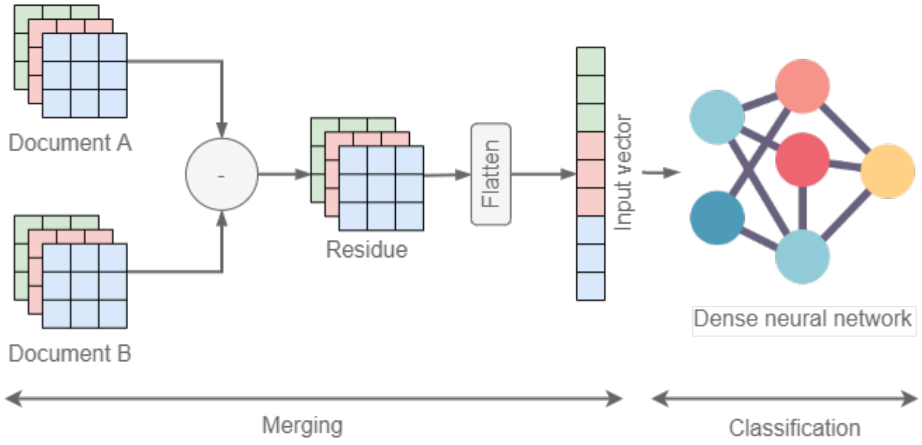

Figure 2: Authorship verification proposal.

5. Experiments and Results

Following the methodology described in Section 4 we transformed all unique documents into matrices of size 20×20 by training SOM models for each layer. The SOM models were trained with a learning rate of 0.01 and 1000 epochs each. We used the *Doc2Vec* algorithm to obtain vectors of size 300 using a window of 5 words for the syntactic and semantic layer.

All the implementations were made by using the Keras[4] and Tensorflow tools[5]. Also, we used the POS tagger algorithm implemented in NLTK[6], and the Doc2Vec implementation of Gensim[7].

After the transformation of texts was done, we used the merging strategy shown in Figure 2 to train a multilayer perceptron neural network with the following architecture:

- A first dense layer with 600 neurons.
- A ReLU activation function.
- An L2 kernel regularizer.
- A first drop out layer with value of 0.4.
- A second dense layer with 300 neurons.
- A ReLU activation function.
- An L2 kernel regularizer.
- A second drop out layer with value of 0.4.
- A third dense layer with 1 neuron.
- A Sigmoid activation function.

Given that this task is a binary classification problem, we used the *binary cross entropy* as a loss metric to guide the algorithm during the training process. This model was trained with a

[4] keras.io
[5] www.tensorflow.org
[6] www.nltk.org
[7] radimrehurek.com/gensim/models/doc2vec.html

total of 100 epochs using the Adam optimizer. We combined 1, 2, and 3 spectral layers from texts to train this model for completeness. We used the test partition to validate the training results with the metrics established in the shared task. Table 4 summarizes the results of these experiments, where column *auc* is the conventional area-under-the-ROC-curve [17], *c@1* is a variant of the conventional F1-score [18], *f_05_u* is a measure that emphasizes correctly deciding same-author cases [19], *F1* score is the harmonic mean of precision and recall [20], *brier* is the complement of the Brier score [21], and *overall* is the average of all these metrics.

Table 4
Results of the classification model using different SOM layers and its combinations on our testing set.

SOM layers	auc	c@1	f_05_u	F1	brier	overall
all	**0.592**	**0.593**	**0.596**	**0.611**	**0.756**	**0.630**
lexical-syntactical	0.596	0.584	0.584	0.581	0.756	0.620
syntactical-semantic	0.512	0.509	0.471	0.420	0.744	0.531
lexical-semantic	0.597	0.593	0.593	0.598	0.755	0.627
lexical	0.594	0.565	0.556	0.523	0.754	0.598
syntactical	0.491	0.500	0.556	0.667	0.750	0.593
semantic	0.493	0.503	0.557	0.668	0.750	0.594

From the previous table, we can highlight that the best results on average are obtained when using the three layers of the text representation. This suggests that using all the information provided by the different layers together allows the classifier to better discern between the positive and negative cases in the problem. Using the lexical and semantic spectra exhibits the best results for the cases of two-layer combinations. This indicates that the vocabulary and the main idea support the decision-making more than using the syntax of the content, suggesting that the topic discussed in the text is important as well as the used vocabulary. Finally, when using a single layer, the best results on average are obtained by using only the lexical specter. This may lead to the fact that the vocabulary used by the different authors in the dataset is crucial to determining the separability of the problem. Given the obtained results, we selected the model trained with all the layers of the text representation to deploy it on the TIRA platform for use with the task test corpus [22].

6. Conclusions

In this paper, we addressed the authorship verification problem by using a content spectral-based representation to train a multilayer perceptron neural network to classify those cases in which the same author produced pairs of texts. Our classification model achieved better results when using all text content spectra. However, even using this information, the highest average score is 0.630. We think we can improve this performance if additional information, such as the type of discourse of the texts, is included as a feature into the model. This variable was not used in our experiments. However, in future work, using such information could help to ensure a better separability of classes.

Acknowledgments

This work has been carried out with the support of CONACyT projects CB A1-S-27780, DGAPA-UNAM PAPIIT numbers TA400121 and TA101722, and CONACYT 709733 scholarship. The authors thank CONACYT for the computing resources provided through the Deep Learning Platform for Language Technologies of the INAOE Supercomputing Laboratory. We thank Eng. Roman Osorio for his help in managing the project's students.

References

[1] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pęzik, M. Potthast, et al., Overview of pan 2022: Authorship verification, profiling irony and stereotype spreaders, style change detection, and trigger detection, in: European Conference on Information Retrieval, Springer, 2022, pp. 331–338.

[2] E. Stamatatos, M. Kestemont, K. Kredens, P. Pezik, A. Heini, J. Bevendorff, M. Potthast, B. Stein, Overview of the Authorship Verification Task at PAN 2022, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022.

[3] J. Weerasinghe, R. Singh, R. Greenstadt, Feature vector difference based authorship verification for open-world settings., in: CLEF (Working Notes), 2021, pp. 2201–2207.

[4] M. Pinzhakova, T. Yagel, J. Rabinovits, Feature similarity-based regression models for authorship verification., in: CLEF (Working Notes), 2021, pp. 2108–2117.

[5] E. Araujo-Pino, H. Gómez-Adorno, G. F. Pineda, Siamese network applied to authorship verification., in: CLEF (Working Notes), 2020.

[6] G. Verma, B. V. Srinivasan, A lexical, syntactic, and semantic perspective for understanding style in text, arXiv preprint arXiv:1909.08349 (2019).

[7] M. Crespo-Sanchez, I. Lopez-Arevalo, E. Aldana-Bobadilla, A. Molina-Villegas, A content spectral-based text representation, Journal of Intelligent & Fuzzy Systems (2022) 1–12.

[8] P. Juola, Authorship attribution, volume 3, Now Publishers Inc, 2008.

[9] Z. Peng, L. Kong, Z. Zhang, Z. Han, X. Sun, Encoding text information by pre-trained model for authorship verification., in: CLEF (Working Notes), 2021, pp. 2103–2107.

[10] X. Miao, H. Qi, Z. Zhang, G. Cao, R. Lin, W. Lin, Dual neural network classification based on bert feature extraction for authorship verification., in: CLEF (Working Notes), 2021, pp. 2069–2072.

[11] R. Futrzynski, Author classification as pre-training for pairwise authorship verification., in: CLEF (Working Notes), 2021, pp. 1945–1952.

[12] A. Menta, A. Garcia-Serrano, Authorship verification with neural networks via stylometric feature concatenation., in: CLEF, 2021.

[13] J. Tyo, B. Dhingra, Z. C. Lipton, Siamese bert for authorship verification., in: CLEF (Working Notes), 2021, pp. 2169–2177.

[14] D. J. MacKay, D. J. Mac Kay, Information theory, inference and learning algorithms, Cambridge university press, 2003.

[15] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International conference on machine learning, PMLR, 2014, pp. 1188–1196.

[16] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78 (1990) 1464–1480.

[17] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern recognition 30 (1997) 1145–1159.

[18] A. Peñas, A. Rodrigo, A simple measure to assess non-response (2011).

[19] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Generalizing unmasking for short texts, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 654–659.

[20] B. Wang, C. Li, V. Pavlu, J. Aslam, A pipeline for optimizing f1-measure in multi-label text classification, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 913–918.

[21] G. Blattenberger, F. Lad, Separating the brier score into calibration and refinement components: A graphical exposition, The American Statistician 39 (1985) 26–32.

[22] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.