

Graph-Based Siamese Network for Authorship Verification

Notebook for PAN at CLEF 2022

Jorge Alfonso Martinez-Galicia¹, Daniel Embarcadero-Ruiz², Alejandro Ríos-Orduña³
and Helena Gómez-Adorno⁴

¹Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Ciudad de México, México

²Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Ciudad de México, México

³Facultad de Ciencias, Universidad Nacional Autónoma de México, Ciudad de México, México

⁴Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad de México, México

Abstract

Authorship verification is the task of determining whether or not the same author wrote two texts based on comparing the texts. The PAN@CLEF 2022 Authorship Verification challenge [1] requires solving the task on a cross-Discourse Types and open-set collection of essays, emails, text messages, and business memos. Our approach is to extract features from the text by first modeling it as a graph and then using a graph neural network to identify relevant features. We use a Siamese Network Architecture because it has shown good generalization on unseen classes in previous work related to verification tasks.

Keywords

Authorship verification, Text graphs, Graph neural networks, Siamese network

1. Introduction

The Authorship analysis research field study the characteristics that help to define an author's writing style. The features can be extracted using text samples of the authors. This research area includes different tasks such as authorship attribution, author profiling, author clustering, and plagiarism detection [5]. The authorship verification task aims at determining if the same author wrote two given texts.

To approach the authorship verification task at PAN 2022, we used a Siamese network architecture composed of two graph convolutional neural networks, pooling, and classification layers as introduced in [18]. We also evaluated the three strategies (short, med, and full) for representing texts as graphs based on the relation of the POS labels and the co-occurrence of the

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

✉ tiuh22@hotmail.com (J. A. Martinez-Galicia); danielembro@gmail.com (D. Embarcadero-Ruiz);


aro-6@ciencias.unam.mx (A. Ríos-Orduña); helena.gomez@iimas.unam.mx (H. Gómez-Adorno)

🌐 <https://helenagomez-adorno.github.io> (H. Gómez-Adorno)

🆔 0000-0003-1904-1606 (H. Gómez-Adorno)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

words. The graph representation provides structural information that can help us distinguish writing styles independently of the discourse type.

2. Related work

The authorship analysis started in the 19th century, the first tackled approach was using linguistic and eventually by statistical and computational methods [6].

One of the most important approach for authorship analysis is using feature extraction to train a classification algorithm either supervised learning or similarly measures. The extracted feature method could be determined by the computational requirements in semantic level, i.e., semantic dependencies, synonyms; syntactic level, i.e., chunks, POS tags, sentence and phrase structure; character level, i.e., character types, character n-grams, count of special characters; and lexical level, i.e., misspelled words, sentence length, word length, bag of words, vocabulary richness [7].

Discriminant analysis, support vector machines, decision trees, neural networks, and genetic algorithms, are examples of supervised classification algorithms used in authorship verification analysis [8]. If we consider the authorship verification as an open set classification problem over a lot of possible authors the option could be using the similarity between texts to predict whether both texts are written by the same author or not [9].

Some proposals in recent years to solve the authorship verification task have been neural network like Bagnall [10] that proposed a recurrent neural network architecture over characters in the PAN 2015 dataset, it's important to say that this approach achieved better performance than others over the same dataset, but one disadvantages of this method is that is computationally more expensive.

One option which can model relationships and structural information effectively is the mathematical construct graph representation. This representation can be possible using feature term as vertices and significant relations between the feature terms as edges [11].

The graph-based approach consist of identifying relevant elements in the text, i.e., words, sentences, paragraphs, etc. and model them as nodes in the graph. Then meaningful relations between these elements are considered to be edges. Normally, the elements used as nodes in the graph are words, sentences, paragraphs, documents, and concepts. To define the edges, usually syntactic, semantic relations, and statistical counts are used [11].

Co-occurrence graph, co-occurrence based on POS, semantic graph and hierarchical keyword graph are relevant graph-based representations proposed for authorship analysis [11].

Castillo et al. [12] proposed different graph-based representations to solve authorship analysis tasks.

The aim of this manuscript is to highlight the importance of enriched vs. non-enriched co-occurrence graphs as an alternative to traditional feature representation models such as vector representation.

There are many applications where data can be represented as a graph. While deep learning effectively captures hidden patterns of Euclidean data, graph neural networks can help us to generalize the deep learning approach to data represented as graphs [13].

Bromley et al. [14] was the first one to solve the problem of signature verification using Siamese Neural Networks (SNN).

The formulation defines two separate sub-networks, each one acting on an input pattern to extract features. The key idea of their formulation is that the two sub-networks share their weight; that means that both sub-networks must extract the features exactly in the same way. They use the cosine of the angle between the two feature vectors obtained by the sub-networks to assign a distance between the compared instances. The idea is that the siamese network learn how to extract feature vectors from the instances in a way these vectors are close if the instances are similar and these vectors are far if not. SNNs are in general computationally expensive but perform better as compared to other techniques when learning similarity [15].

Recently, SNNs were proposed to solve the authorship verification task. For instance, two approaches used SNNs to solve the PAN Authorship 2020 Verification Task. The Siamese neural network approach has been successfully applied to solve several verification related tasks. We consider that this method is a natural and powerful way to take advantage of a graph-based representation of texts.

3. Authorship Verification Dataset at PAN 2022

The train dataset provided by the PAN@CLEF 2022 [3] organization consist of cross discourse types authorship verification cases using the following DTs: essays, emails, text messages and business memos. The corpus comprises texts of around 56 individuals. All individuals have similar age (18-22) and are native English speakers. The topic of text samples is not restricted while the level of formality can vary within a certain DT, the total number of predefined pairs is 12,264. Each problem is composed of two texts belonging to two different DTs.

This year, the Authorship Verification task [1] focused on a cross-DT authorship verification and open-set scenario. The test datasets are structured in the same way, but their author sets are not overlapping. We submitted one model trained using the texts provided in the dataset. To develop our model, we conducted several experiments, and for these we need to split the dataset in three parts: train, validation, and test. We trained our model on the train split using the validation split to calibrate the hyperparameters, and used the test split to get a reference score of the model. We did not use any of the samples in the test split to calibrate our model, so the score in this set tells us about the generalization ability of the model.

Our splits were done using the same fixed pairs given by the dataset. We made these splits author disjoint, that is, no text in one split has the same author of any text in a different split, this approach was difficult to achieve because the difference between authors and problems was really high, we had 12,264 problems and only 56 authors. The result of the split the dataset in an author disjoint way was unbalanced splits, we had more positive problems than negative, thus, we had to generate new problems for each split to make it balanced.

After develop and calibrate our models we deployed them on TIRA [4] for testing it.

4. Modeling Texts as Graphs

Our graphic representation finds the relationship between words and POS labels. Before obtaining our graphic representation, we perform a text preprocessing with the intention of obtaining the greatest amount of information possible for each text, this pre-processing consisted of the following steps:

- Substitute no ASCII characters to ASCII equivalent (we employed unicode package1).
- Tokenize and get the POS label.(We employed the RegexpTokenizer package)
- Normalize to Lowercase.

We decided to respect punctuation types, substitution of non-ASCII characters was done to reduce useful punctuation enhancement.

To get the POS tags, we consider the PENN-Treebank POS tags and then add two additional tags: \$PUNCT to mark all punctuation and \$OTHER to mark any other words that the NLTK model could not identify. In total, we decided to consider 38 labels. After this process we obtain a list of tuples, each token with its corresponding POS label. To illustrate this process we show the list obtained for the following text:

<nl>I am a Second year <course> student at <university>. I am interested.

```
[ ('<nl>', 'NN'), ('i', 'PRP'), ('am', 'VBP'), ('a', 'DT'), ('second', 'JJ'), ('year', 'NN'), ('<course>', 'FW'), ('student', 'NN'), ('at', 'IN'), ('<university>', 'NNP'), ('.', '$PUNCT'), ('i', 'PRP'), ('am', 'VBP'), ('interested', 'JJ'), ('.', '$PUNCT')]
```

We build a co-occurrence graph considering that two words coexist if they appear next to each other in the text.

We define the graph as an ordered pair $G = (V, E)$, where V is a set of vertices composed by (word, pos) tuples and E is a set of weighted edges. The edge set $E \subseteq \{(n_1, n_2, w) | n_1, n_2 \in V, n_1 \neq n_2, w \in \mathbb{R}\}$, where w is the edge weight.

In order to change the structure of the graph and also the information abstracted from the text we create a set of POS labels and denote it as REDUCE_LABELS. To construct the graph, let P be the parsed text as a list of tuples, $l(P)$ the number of elements in the list, and $P[i]$ the i -th element in the list. For each $P[i] = (\text{word}, \text{pos})$ in P , we can define:

$$M[i] = \begin{cases} (\text{word}, \text{pos}) & \text{if } \text{pos} \notin \text{REDUCE_LABELS} \\ (\text{pos}, \text{pos}) & \text{if } \text{pos} \in \text{REDUCE_LABELS} \end{cases}$$

where M is the list defined by the tuples masked as explained. For each pair of tuples $T_1, T_2 \in M$ let be $f(T_1, T_2)$ the number of times T_1 is followed by T_2 in M and let be $T = l(P) - 1 = l(M) - 1$; note that T is the total number of times a pair of tuples co-occur in M .

Now we can define the nodes and edges of our graph:

$$V = \{T | T \in M\}$$

Please note that M is a list with order and V is just the set of all tuples in M . We want to define an edge between any two nodes (tuples) that appear together in the list M :

$$E = \{(T_1, T_2, \frac{f(T_1, T_2)}{T}) | T_1, T_2 \in M \wedge f(T_1, T_2) > 0\}$$

As we said before, with this construction we identify all tuples with a specific label in the REDUCE_LABELS set as a single node. In our experiments, we evaluated graphs generated with different REDUCE_LABELS sets. From now we will denominate *short graph* to the graph generated using the set of all possible POS labels as REDUCE_LABELS, *full graph* to the graph generated using REDUCE_LABELS = \emptyset and we will denominate *med graph* to the graph generated using the following set of REDUCE_LABELS:

REDUCE_LABELS = ['JJ',	'JJR',	'JJS',	#Adjectives
	'NN',	'NNS',	'NNP', 'NNPS',	#Nouns
	'RB',	'RBR',	'RBS',	#Adverbs
	'VB',	'VBD',	'VBG',	#Verbs
	'VBN',	'VBP',	'VBZ',	#Verbs
	'CD',			#Cardinal numbers
	'FW',			#Foreign words
	'LS',			#List item marker
	'SYM',			#Symbols
	'\$OTHER',			# Others]

Continuing with our example, the med graph and the short graph are showed in (Figure 2) and (Figure 1) respectively. To make clearer the construction process (Figure 3) show the construction with empty REDUCE_LABELS set.

To input our graph to a neural network, we need to encode each node into a vector. To that end, we use a one-hot encoding representation with respect to the 38 possible POS tags used. Exists reference where a deep learning model that trains on POS embeddings instead of word embeddings performs better for a [16] authorship analysis task. Furthermore, the use of low-dimensional POS embeddings to represent nodes allows to reduce the computational cost of the model.

5. Graph-based Siamese Network (GBSN)

To approach the authorship verification task, we use a Siamese network architecture [14] including a component to transform texts as co-occurrence graphs. Our Graph-based Siamese network (Figure 4) is composed of two identical feature extraction components with shared weights, a reduction step, and a classification network.

Each feature extraction component receives a text, transforms it into a graph, and returns a vector representation of this graph; the objective is to extract relevant features that can identify the author's style from the graph representation of the texts. We can distinguish three parts in the feature extraction component: graph representation, node embedding layers, and global pooling.

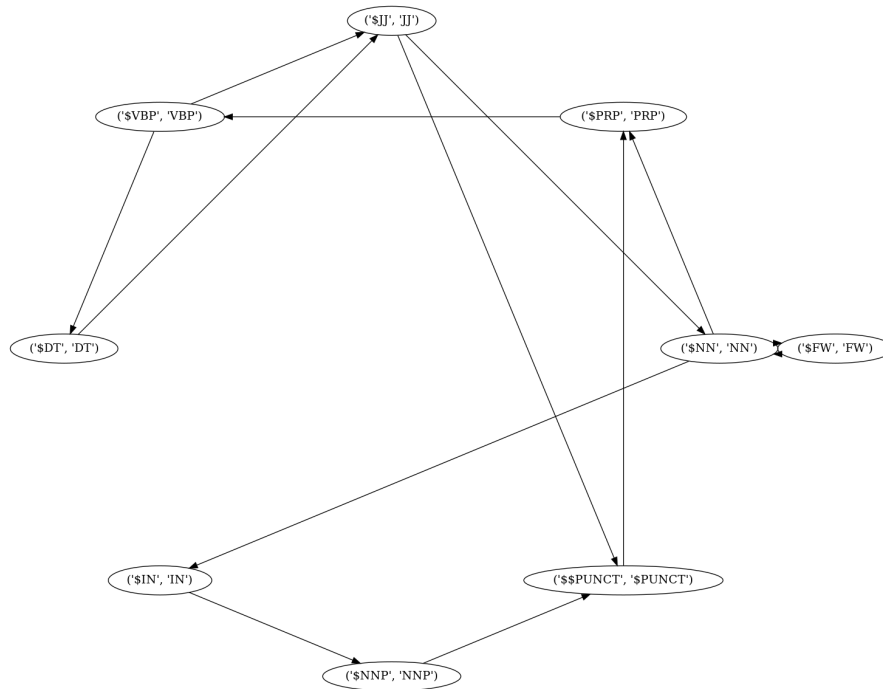


Figure 1: (a) Short graph

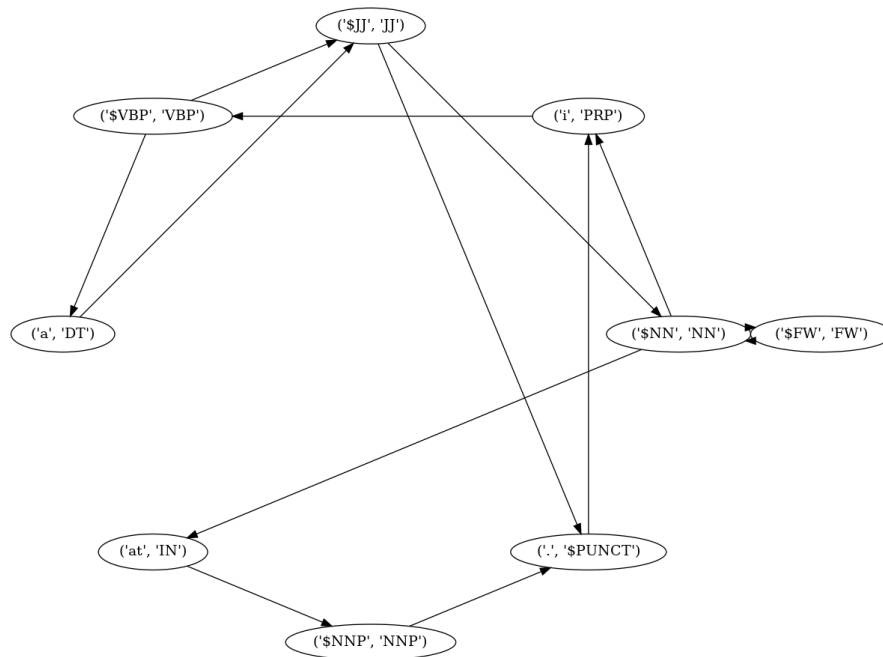


Figure 2: (b) Med graph

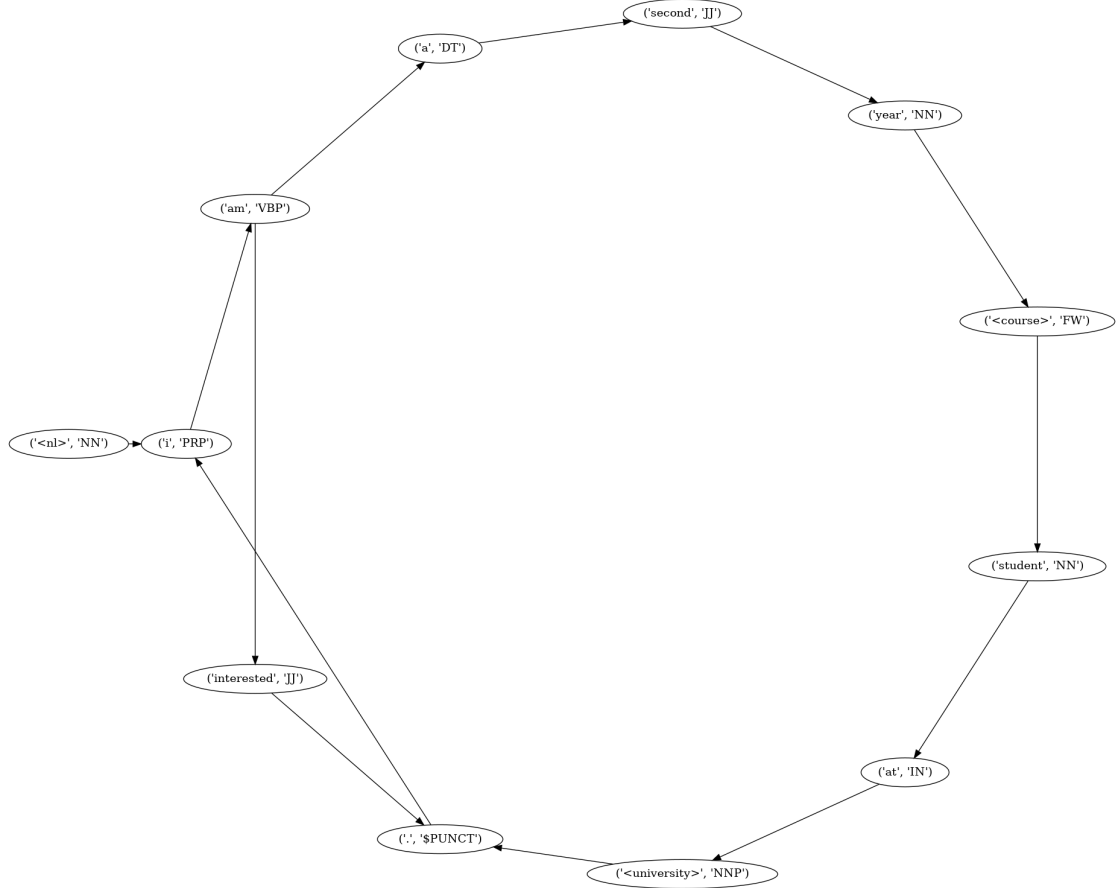


Figure 3: (c) Co-occurrence graph , equivalent to the graph generated with empty REDUCE LABELS set

In our architecture, a node embedding layer is composed of a graph convolutional layer, followed by a batch normalization layer, and a ReLU (Rectified linear activation function).

The first node embedding layer takes as input a graph with an initial feature vector in each node, each initial node vector has dimension 38 because this vector is a one-hot representation of the POS label of the node. The output of each node embedding layer is the same graph structure with new feature vectors in each node; the dimension of the vectors obtained can be defined in the same way we define the channels used in a traditional convolutional layer. Our architecture obtains vectors of dimension 64 in each convolutional layer.

We need a pooling layer to obtain a single vector representation for the whole graph.

In our model, we use a global attention layer for the pooling, originally proposed by Li et al. [17]. As it is shown in (Figure 5), this layer takes the final output of the node feature extraction component as its input, i.e., a graph with the vector embedding in each node. To obtain the final vector, it makes a weighted sum of each node vector with a coefficient obtained by doing attention over these same vectors, the formulation is:

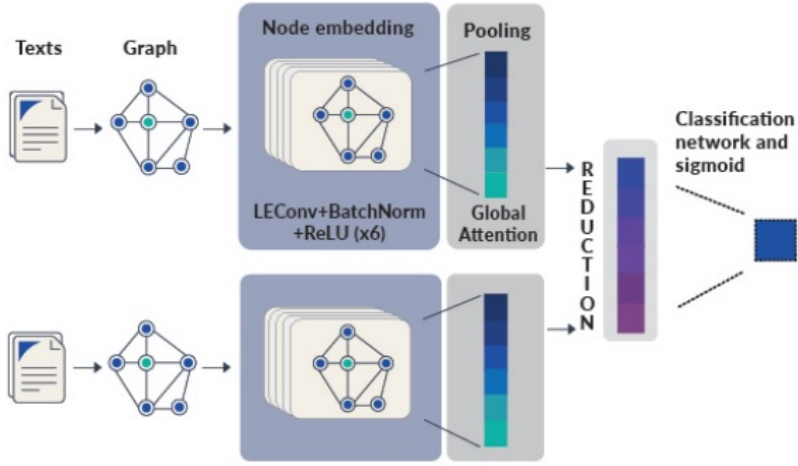


Figure 4: GBSN base architecture

$$r = \sum_{n \in V} softmax(h(x_n)) \cdot x_n$$

where V is the set of all nodes in the graphs and h is a fully connected neural network with a single scalar as output. This fully connected neural network has ReLU (Rectified linear activation function) activation, 32 neurons in each hidden layer, and L_{pool} total layers. The final output of each feature extraction component is a vector with dimension 64.

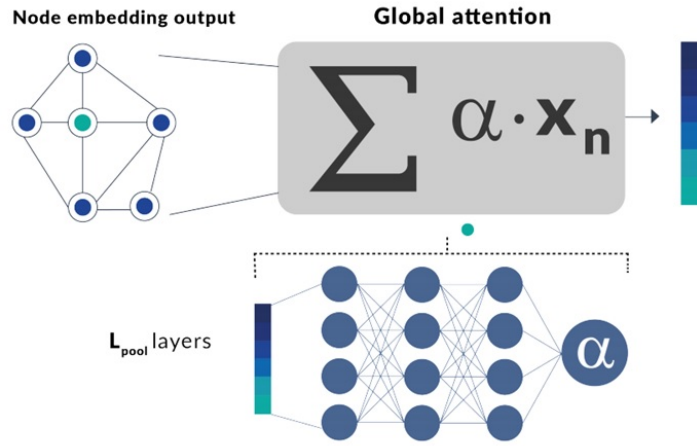


Figure 5: Global Attention Pooling layer.

For the reduction step, we simply compute the absolute value of the difference between the output of each feature extraction component for each document to be verified. The resulting

vector is passed to a final classification network. The classification network is a fully connected network with ReLU (Rectified linear activation function) activation, 64 neurons in each hidden layer, L_{class} total layers, and a final sigmoid function.

Our model returns a single value in the interval $[0, 1]$ that can be interpreted as a measure of how much the two submitted texts are alike. An output close to 1 tells us that the model finds both texts to be from the same author.

6. Results

Table 1 shows the comparative performance of the different configurations, each row shows the average of the five proposed scores in a model. The first row shows the scores of our submitted model and corresponds to a base architecture model using only the short graph component for feature extraction. The second row corresponds to a base architecture model using only the med graph component. The third row corresponds to a base architecture model using only the full graph component. Finally, our submission were scored in the test dataset of the PAN 2022 Authorship Verification task [1].

Table 1

Performance of the Graph-based Siamese Network (GBSN) with single feature extraction components in our test split dataset

	Dataset 15,732 problems
Short graph component	60.09
Med graph component	56.36
full graph component	59.98

7. Conclusions

In this paper, we presented our approach for the authorship verification task at PAN 2022 [1], which is based on a Graph-based Siamese Network. In first experiments, we tried different convolutional and pooling graph layers configurations, but we observed in the loss function graph that the model got overfitted really fast. After some experiments we noticed that a simple convolutional and pooling graph layer delivered a better performance, but we know there are modifications we can make either to the dataset or to the configuration of the layers in the model to get a better performance in the future work. We made a slightly modification to the dataset since we had a short amount of problems to use in the design stages, the model was training using 15,732 problems which is really small for this kind of tasks and this is relevant because we know our architecture could work better with more training pairs.

Acknowledgments

This work has been carried out with the support of CONACyT projects CB A1-S-27780, DGAPA-UNAM PAPIIT numbers TA400121 and TA101722. The authors thank CONACyT for the computing resources provided through the Deep Learning Platform for Language Technologies of the INAOE Supercomputing Laboratory, also thanks Engr. Roman Osorio for supporting the student administration of the project.

References

- [1] M. Kestemont, E. Stamatatos, E. Manjavacas, J. Bevendorff, M. Potthast, and B. Stein, Overview of the Cross-Domain Authorship Verification Task at PAN 2021. in: [2].
- [2] M. Kestemont, E. Manjavacas, I. Markov, J. Bevendorff, M. Wiegmann, E. Stamatatos, B. Stein, and M. Potthast, Overview of the Cross-Domain Authorship Verification Task at PAN 2021.
- [3] Research England Expanding Excellence in England (E3) grant awarded to the Aston Institute for Forensic Linguistics.
- [4] 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.
- [5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature Verification using a "Siamese" Time Delay Neural Network, *International Journal of Pattern Recognition and Artificial Intelligence* 7 (1993) 669–688. doi:10.1142/S0218001493000339.
- [6] Mekala, S.; Bulusu, V.V. A Survey On Authorship Attribution Approaches. *Int. J. Comput. Eng. Res. (IJCER)* 2018, 8, 8.
- [7] Stamatatos, E. A Survey of Modern Authorship Attribution Methods. *J. Am. Soc. Inf. Sci. Technol.* 2009, 60, 538–556.
- [8] Stamatatos, E.; Daelemans, W.; Verhoeven, B.; Potthast, M.; Stein, B.; Juola, P.; Sanchez-Perez, M.A.; Barrón-Cedeño, A. Overview of the Author Identification Task at PAN 2014. *CLEF 2014*, 1180, 877–897.
- [9] Koppel, M.; Winter, Y. Determining If Two Documents Are Written by the Same Author. *J. Assoc. Inf. Sci. Technol.* 2014, 65, 178–187.
- [10] Bagnall, D. Author Identification Using Multi-Headed Recurrent Neural Networks. arXiv 2015, arXiv:1506.04891.
- [11] Sonawane, S.S.; Kulkarni, P.A. Graph Based Representation and Analysis of Text Document: A Survey of Techniques. *Int. J. Comput. Appl.* 2014, 96, 1–8.
- [12] Castillo, E.; Cervantes, O.; Vilariño, D. Text Analysis Using Different Graph-Based Representations. *Comput. Sist.* 2017, 21, 581–599.
- [13] Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* 2020, 32, 4–24.
- [14] Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature Verification Using a "Siamese" Time Delay Neural Network. *Int. J. Pattern Recognit. Artif. Intell.* 1993, 7, 669–688.
- [15] Nandy, A.; Halder, S.; Banerjee, S.; Mitra, S. A Survey on Applications of Siamese Neural

Networks in Computer Vision. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; pp. 1–5.

- [16] Jafariakinabad, F.; Tarnpradab, S.; Hua, K.A. Syntactic Recurrent Neural Network for Authorship Attribution. arXiv 2019, arXiv:190.09723.
- [17] Li,Y.;Tarlow,D.; Brockschmidt,M.; Zemel, R. Gated Graph Sequence Neural Networks.arXiv 2015, arXiv:1511.05493.
- [18] Embarcadero, Ruiz D., Gómez, Adorno H., Embarcadero, Ruiz A., and Sierra, G. (2022) Graph based Siamese network for authorship verification, *Mathematics*, 10(2), 277.