

# Stylometric and Neural Features Combined Deep Bayesian Classifier for Authorship Verification

Notebook for PAN at CLEF 2023

Yitao Sun<sup>1</sup>, Svetlana Afanaseva<sup>1</sup> and Kailash Patil<sup>1</sup>

<sup>1</sup>*Pindrop*

## Abstract

This paper describes the approach of a deep learning model for the PAN 2023 Cross-Discourse Type Authorship Verification Task [1]. We present a hierarchical fusion of two well-established approaches into a single end-to-end learning process: A deep metric learning framework at the top aims to align and learn from a pseudo-metric that maps a document of variables to a fixed-length feature vector. A separate extraction layer then extracts stylometric features from the document. Finally, the Bayesian probabilistic layer scores the concatenated features to predict the similarity of the documents.

## Keywords

deep learning, authorship verification, stylometric, machine learning, natural language processing, NLP, CEUR-WS, PAN

## 1. Introduction

Authorship verification (pairwise) involves determining whether two documents were authored by the same individual. Traditionally, linguists have undertaken authorship verification to ascertain the authorship of anonymous texts by examining specific linguistic features. These features encompass a range of elements, such as errors (e.g. spelling mistakes), peculiarities in the text (e.g. grammatical inconsistencies), and patterns of writing style [2].

Automated systems, particularly those based on machine learning, have heavily depended on **stylometric features** [3]. These features are derived from linguistic metrics and are commonly used to analyze text. However, one limitation of stylometric features is that their effectiveness tends to decrease when applied to texts that exhibit significant variations in topics.

On the other hand, **deep learning systems** [4] can be designed to autonomously learn neural features in a comprehensive manner. These features can be insensitive to the specific topic of the text. However, a drawback of such features is that they are generally not easily interpretable from a linguistic perspective.

In this study, we present a significant expansion of a popular and previously published **ADHOMINEM method** [4]. In our extended approach, we not only analyze the neural features

---

*CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece*


✉ [ysun@pindrop.com](mailto:ysun@pindrop.com) (Y. Sun); [safanaseva@pindrop.com](mailto:safanaseva@pindrop.com) (S. Afanaseva); [kpatil@pindrop.com](mailto:kpatil@pindrop.com) (K. Patil)

🌐 <https://www.linkedin.com/in/yitao-s-146015104/> (Y. Sun)

🆔 0009-0007-3743-9331 (Y. Sun)



© 2023 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)

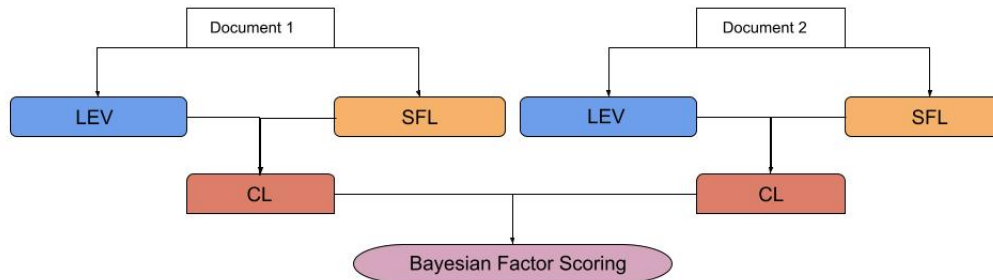
generated by ADHOMINEM using a metric perspective but also incorporate a stylometric viewpoint. This allows for a more comprehensive extraction of features from the documents.

This paper is structured as follows: we will describe our approach in Section 2, present our evaluation results in Section 3 and discusses our conclusions and future work in Section 4.

## 2. Approach

We pre-define a deep learning model architecture along with its hyper-parameters and thresholds and allow the model to autonomously learn suitable features for the provided setup. This approach is in line with most deep-learning methodologies. The success of our proposed setup heavily relies on the availability of a large collection of text samples that encompass diverse variations in writing style, enabling the model to learn effectively.

We utilize a predecessor of our ADHOMINEM system [4] as a deep metric learning framework [5] and document-level Stylometric features extractor to assess the similarity between two text samples. The concatenated features generated by the system are then inputted into a probabilistic linear discriminant analysis (PLDA) layer [6]. This layer serves as a pairwise discriminator, conducting Bayes factor scoring within the learned metric space, thus contributing to the discriminative power of our method.<sup>1</sup>



**Figure 1:** Model structure

## 2.1. Neural extraction of linguistic embedding vectors (LEV) [5]

A text sample can be seen as a hierarchical structure composed of discrete elements arranged in a specific order. It starts with a list of sentences, where each sentence is comprised of an ordered sequence of tokens. Furthermore, each token consists of an ordered sequence of characters. The primary objective of ADHOMINEM is to transform a document into a feature vector. Specifically, its Siamese topology incorporates a hierarchical neural feature extraction process that captures the stylistic attributes of a pair of documents ( $D_1, D_2$ ), which can have varying lengths. This process results in a pair of fixed-length linguistic embedding vectors (LEVs), denoted as  $y_i$

$$y_i = \mathcal{A}_\theta(\mathcal{D}_i) \in \mathbb{R}^{D \times 1}, i \in \{1, 2\} \quad (1)$$

we denote the dimension of the linguistic embedding vectors (LEVs) as  $D$ , and  $\theta$  represents all the trainable parameters involved. This network is referred to as a Siamese network because both documents  $D_1$  and  $D_2$  undergo mapping through the exact same function.

## 2.2. Stylometric features layer (SFL)

In this section, we outline the features, which are commonly utilized in previous stylometry research [3]. We selected these features from the Writeprints feature set introduced by Weerasinghe [7]. Additionally, recognizing the importance of the syntactic structure of sentences in providing informative signals to the classifier, we included POS-Tag n-grams and partial parses (or POS-Tag chunks) as part of our feature set, following the approach of previous studies [8]. Sidorov et al. [9] introduced the use of parse trees for extracting stylometric features, specifically syntactic dependency-based n-grams of POS tags. However, we employed a slightly different method to encode parse tree features, which focuses on capturing the construction of different noun and verb phrases.

Furthermore, several features were computed based on TFIDF (Term Frequency-Inverse Document Frequency) values. We utilized NLTK's `TFIDFVectorizer` to compute the TF-IDF vectors for the documents. To exclude tokens with a document frequency below 10%, we set the `min token` parameter to 0.1.

- Character n-grams: TF-IDF values for character n-grams, where  $1 \geq n \geq 6$ .
- POS-Tag n-grams: TF-IDF value of POS-Tag tri grams.
- Frequency of Function Words: Frequencies of 179 stopwords defined in the *NLTK* corpus package.
- Vocab Richness: computed by dividing the combined count of words that appear only once (hapax-legomenon) and words that appear twice (dis-legomenon) in the document, by the total number of tokens in the document. This normalization accounts for variations in document lengths.
- POS-Tag Chunks: TF-IDF values for Tri-grams of POS-Tag chunks. Here, we consider the tokens at the second level of our parse tree.
- NP and VP construction: TF-IDF values of each noun phrase of verb phrase expansion.
- number of characters
- number of words

- Average number of characters per word
- Distribution of word-lengths (1-10)

After concatenating the above features, we use truncated singular value decomposition (SVD) to reduce the dimensions from 8708 to 10 dimensions before concatenating with LEVs.

### 2.3. Bayes factor scoring [10]

Text samples exhibit significant variations, making it valuable to employ statistical hypothesis tests to quantify the outputs or scores generated by our algorithm. These tests aid in determining whether to accept or reject a decision. ADHOMINEM has the potential to incorporate a framework for conducting statistical hypothesis testing. Specifically, we focus on the authorship verification (AV) problem, where we are presented with the linguistic embedding vectors (LEVs) and Stylometric features layer (SFL) of two documents. We concatenate them into combined layers (CLs) and then make a decision based on one of two hypotheses:

- $\mathcal{H}_s$ : The two documents were written by the same person,
- $\mathcal{H}_d$ : The two documents were written by two different persons.

$$\underbrace{\mathbf{y}}_{\text{combined layers}} = \underbrace{\mathbf{x}}_{\text{author's writing style}} + \underbrace{\boldsymbol{\epsilon}}_{\text{noise term}} \quad (2)$$

The combined layer  $\mathbf{CL}_s$   $\mathbf{y}$  is decomposed into a latent writing style vector  $\mathbf{x}$  and a noise term  $\boldsymbol{\epsilon}$  are in Eq. (2). The probability density functions for  $\mathbf{x}$  and  $\boldsymbol{\epsilon}$  are as shown in Eq. (3):

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \mathbf{B}^{-1}) \\ p(\boldsymbol{\epsilon}) &= \mathcal{N}(\boldsymbol{\epsilon} | \mathbf{0}, \mathbf{W}^{-1}) \end{aligned} \quad (3)$$

$$p(\mathbf{y} | \mathbf{x}) = \mathcal{N}(\mathbf{y} | \mathbf{x}, \mathbf{W}^{-1}) \quad (4)$$

**Same-author pair probability:** A single latent vector  $\mathbf{x}_0$  representing the author's writing style is generated from the prior  $p(\mathbf{x})$  and both  $\mathbf{CL}_s \mathbf{y}_i, i \in \{1, 2\}$  are generated from  $p(\mathbf{y} | \mathbf{x}_0)$  in Eq. (4). The joint probability density function is then given by:

$$p(\mathbf{y}_1, \mathbf{y}_2 | \mathcal{H}_s) = \frac{p(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}_0, \mathcal{H}_s) p(\mathbf{x}_0 | \mathcal{H}_s)}{p(\mathbf{x}_0 | \mathbf{y}_1, \mathbf{y}_2, \mathcal{H}_s)} = \frac{p(\mathbf{y}_1 | \mathbf{x}_0) p(\mathbf{y}_2 | \mathbf{x}_0) p(\mathbf{x}_0)}{p(\mathbf{x}_0 | \mathbf{y}_1, \mathbf{y}_2)} \quad (5)$$

**Different-authors pair probability:** Two latent vectors  $\mathbf{x}_i, i \in \{1, 2\}$  representing the distinct writing characteristics of two different authors are generated independently from the prior distribution  $p(\mathbf{x})$ . The corresponding linguistic embedding vectors  $\mathbf{y}_i$  are generated from the conditional distribution  $p(\mathbf{y} | \mathbf{x}_i)$ . The joint probability density function can then be expressed as follows:

$$p(\mathbf{y}_1, \mathbf{y}_2 | \mathcal{H}_d) = p(\mathbf{y}_1 | \mathcal{H}_d) p(\mathbf{y}_2 | \mathcal{H}_d) = \frac{p(\mathbf{y}_1 | \mathbf{x}_1) p(\mathbf{x}_1)}{p(\mathbf{x}_1 | \mathbf{y}_1)}, \frac{p(\mathbf{y}_2 | \mathbf{x}_2) p(\mathbf{x}_2)}{p(\mathbf{x}_2 | \mathbf{y}_2)} \quad (6)$$

**Verification process:** The probabilistic model described consists of two distinct phases: a training phase and a verification phase. During the training phase, the parameters of the Gaussian distributions in Eq. (3)-(4) are learned. These distributions capture the characteristics of the latent vectors and linguistic embedding vectors. In the verification phase, the model is utilized to determine whether the two text samples originate from the same author based on the learned parameters as shown in Eq. (7).

$$\begin{aligned}
\text{score}(\mathbf{y}_1, \mathbf{y}_2) &= \log p(\mathbf{y}_1, \mathbf{y}_2 | \mathcal{H}_s) - \log p(\mathbf{y}_1, \mathbf{y}_2 | \mathcal{H}_d) \\
&= \log p(\mathbf{x}_0) - \log p(\mathbf{x}_1) - \log p(\mathbf{x}_2) \\
&\quad + \log p(\mathbf{y}_1 | \mathbf{x}_0) + \log p(\mathbf{y}_2 | \mathbf{x}_0) - \log p(\mathbf{y}_1 | \mathbf{x}_1) - \log p(\mathbf{y}_2 | \mathbf{x}_2) \\
&\quad - \log p(\mathbf{x}_0 | \mathbf{y}_1, \mathbf{y}_2) + \log p(\mathbf{x}_1 | \mathbf{y}_1) + \log p(\mathbf{x}_2 | \mathbf{y}_2)
\end{aligned} \tag{7}$$

A higher value for  $\text{score}(\mathbf{y}_1, \mathbf{y}_2)$  indicates higher similarity and vice versa.

### 3. Training Details

We implemented our training algorithm in Python. We conducted our preprocessing in our customized regular expression function and then use spaCy *en\_core\_web\_lg* to do sentence boundary detection and tokenization. Given that the stylometric part of the model is set and described, we fine-tuned our deep Bayesian model to achieve higher performance. However, none of the fine-tuning trials' performance exceeds the default hyper-parameters model. Details are as follows:

- character embedding dimension: 10
- character representation dimension: 30
- dimension of word embeddings: 300
- dimension of sentence embedding: 50
- dimension of document embedding: 50
- LEV dimension: 60
- dimension reduction for BFS: 40
- maximum number of characters per words: 15
- maximum number of words per sentence: 210
- maximum number of sentences per document: 30
- hop length for sliding windowing: 26
- dropout for attention layer: 0.9
- dropout for BFS layer: 0.8
- dropout for 1D-CNN: 0.8
- variational dropout for BiLSTM layer: 0.9
- dropout for final DML layer: 0.8
- learning rate end: 0.0002
- learning rate start: 0.0006

For the final submitted model in Tira [11], we used the entire training dataset with the above hyper-parameters setting and combined stylometric layers outputs to train the deep Bayesian model. We took epoch number 8, 24, and 35 for our final three submissions.

## 4. Evaluation

The following table presents the experimental results conducted on the competition dataset. The dataset was divided into train and test sets for evaluation purposes. In our analysis, we compared the performance metrics provided by the PAN competition with two baseline models, our predecessor the deep metric model (DML, a model that directly learns from LEV [5]), and the uncertainty adaptation layer model (UAL, which models the noise behavior [12]), and the Bayes factor scoring model (BFS) with/without Stylometric features layer(SFL).

**Table 1**  
Test Results of PAN 2023 Training Dataset

<i>Model</i>	<i>AUC</i>	<i>C@1</i>	<i>f<sub>0</sub>5<sub>u</sub></i>	<i>F1</i>	<i>brier</i>	<i>overall</i>
Naive, Distance-based	0.493	0.497	0.553	0.664	0.741	0.589
Method-based text compression	0.504	0.033	0.048	0.621	0.75	0.391
DML without SFL	0.503	0.523	0.492	0.357	0.603	0.495
UAL without SFL	0.499	0.52	0.477	0.336	0.593	0.485
BFS without SFL	0.47	0.502	0.474	0.37	0.597	0.483
DML with SFL	0.523	0.499	0.605	0.522	0.73	0.576
UAL with SFL	0.568	0.492	0.584	0.467	0.747	0.571
BFS with SFL	<b>0.658</b>	<b>0.662</b>	<b>0.739</b>	<b>0.735</b>	<b>0.762</b>	<b>0.711</b>

The experimental results indicate that the incorporation of the Stylometric Features Layer (SFL) significantly improves the performance of the ADHOMINEM model. Among the various configurations of the ADHOMINEM model, the Bayes factor scoring (BFS) model consistently outperforms the others across all evaluated metrics.

These findings suggest that the integration of the SFL enhances the ability of the ADHOMINEM model to capture relevant stylometric characteristics, leading to improved authorship verification results. The superiority of the BFS model further highlights the effectiveness of the Bayesian factor scoring approach in selecting discriminative features for distinguishing between different authors.

## 5. Conclusions

We have introduced a novel approach to authorship verification (AV) that combines neural feature extraction and stylometric features with statistical modeling. The observed performance improvements affirm the value of the proposed enhancements in the ADHOMINEM model, emphasizing the significance of the feature selection technique and the utilization of stylometric features for the authorship verification task.

In AV, there are numerous factors that introduce variabilities, such as topic, genre, text length and text types, which can negatively impact the performance of the system. However, we believe that there is significant potential for further improvements by incorporating compensation techniques to address these aspects in future challenges.

## Acknowledgments

We thank *PAN2023* [13] organizers for arranging this task and helping us through the submission process. We also thank the reviewers for their helpful comments and feedbacks. Our work was supported by Pindrop.

## References

- [1] E. Stamatatos, K. Kredens, P. Pezik, A. Heini, J. Bevendorff, M. Potthast, B. Stein, Overview of the Authorship Verification Task at PAN 2023, in: *CLEF 2023 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2023.
- [2] S. Ehrhardt, 7. Authorship attribution analysis, De Gruyter Mouton, Berlin, Boston, 2018, pp. 169–200. URL: <https://doi.org/10.1515/9781614514664-010>. doi:doi:10.1515/9781614514664-010.
- [3] E. Stamatatos, A survey of modern authorship attribution methods, *Journal of the American Society for Information Science and Technology* (2009).
- [4] B. T. Boenninghoff, S. Hessler, D. Kolossa, R. M. Nickel, Explainable authorship verification in social media via attention-based similarity learning, *CoRR abs/1910.08144* (2019). URL: <http://arxiv.org/abs/1910.08144>. arXiv:1910.08144.
- [5] B. Boenninghoff, R. M. Nickel, S. Zeiler, D. Kolossa, Similarity learning for authorship verification in social media, in: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019. URL: <https://doi.org/10.1109%2Ficassp.2019.8683405>. doi:10.1109/icassp.2019.8683405.
- [6] S. Cumani, N. Brümmer, L. Burget, P. Laface, O. Plchot, V. Vasilakakis, Pairwise discriminative speaker verification in the i-vector space, *IEEE Transactions on Audio, Speech, and Language Processing* 21 (2013) 1217–1227. doi:10.1109/TASL.2013.2245655.
- [7] J. Weerasinghe, R. Greenstadt, Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification—Notebook for PAN at CLEF 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), *CLEF 2020 Labs and Workshops, Notebook Papers*, CEUR-WS.org, 2020.
- [8] M. Kestemont, E. Manjavacas, I. Markov, J. Bevendorff, M. Wiegmann, E. Stamatatos, M. Potthast, B. Stein, Overview of the cross-domain authorship verification task at pan 2020, in: *Conference and Labs of the Evaluation Forum*, 2020.
- [9] G. Sidorov, F. Castillo, E. Stamatatos, A. Gelbukh, L. Chanona-Hernández, Syntactic n-grams as machine learning features for natural language processing, *Expert Systems with Applications: An International Journal* 41 (2014) 853–860. doi:10.1016/j.eswa.2013.08.015.
- [10] B. T. Boenninghoff, J. Rupp, R. M. Nickel, D. Kolossa, Deep bayes factor scoring for authorship verification, *CoRR abs/2008.10105* (2020). URL: <https://arxiv.org/abs/2008.10105>. arXiv:2008.10105.
- [11] 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. Goeriot, 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.

- [12] B. T. Boenninghoff, D. Kolossa, R. M. Nickel, Self-calibrating neural-probabilistic model for authorship verification under covariate shift, CoRR abs/2106.11196 (2021). URL: <https://arxiv.org/abs/2106.11196>. arXiv:2106.11196.
- [13] J. Bevendorff, I. Borrego-Obrador, M. China-Ríos, M. Franco-Salvador, M. Fröbe, A. Heini, K. Kredens, M. Mayerl, P. Pezik, 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. Tsirikla, 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.