Explainable Authorship Identification in Cultural Heritage Applications

MATTIA SETZU, Università di Pisa, Pisa, Italy
SILVIA CORBARA, Scuola Normale Superiore, Pisa, Italy and Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy
ANNA MONREALE, Università di Pisa, Pisa, Italy
ALEJANDRO MOREO and FABRIZIO SEBASTIANI, Consiglio Nazionale delle Ricerche, Pisa, Italy

While a substantial amount of work has recently been devoted to improving the accuracy of computational Authorship Identification (AId) systems for textual data, little to no attention has been paid to endowing AId systems with the ability to explain the reasons behind their predictions. This substantially hinders the practical application of AId methods, since the predictions returned by such systems are hardly useful unless they are supported by suitable explanations. In this article, we explore the applicability of existing general-purpose eXplainable Artificial Intelligence (XAI) techniques to AId, with a focus on explanations addressed to scholars working in cultural heritage. In particular, we assess the relative merits of three different types of XAI techniques (feature ranking, probing, factual and counterfactual selection) on three different AId tasks (authorship attribution, authorship verification and same-authorship verification) by running experiments on real AId textual data. Our analysis shows that, while these techniques make important first steps towards XAI, more work remains to be done to provide tools that can be profitably integrated into the workflows of scholars.

CCS Concepts: • **Computing methodologies** → **Information extraction**; • **Applied computing** → **Evidence collection, storage and analysis**; • **Human-centered computing** → **HCI theory, concepts and models**;

Additional Key Words and Phrases: Explainable artificial intelligence, cultural heritage, authorship identification

ACM Reference format:
Mattia Setzu, Silvia Corbara, Anna Monreale, Alejandro Moreo, and Fabrizio Sebastiani. 2024. Explainable Authorship Identification in Cultural Heritage Applications. *ACM J. Comput. Cult. Herit.* 17, 3, Article 44 (June 2024), 23 pages.
https://doi.org/10.1145/3654675

The work by A. Moreo and F. Sebastiani has been supported by the SoBigData++ project (Grant 871042) and by the AI4Media project (Grant 951911), both funded by the European Commission (Grant 871042) under the H2020 Programme, by the SoBigData.it project. Furthermore, this work was supported by the National Recovery and Resilience Plan—Prot. IR0000013—Avviso n. 3264 del 28/12/2021, and by project PNRR—M4C2—Investimento 1.3, Partenariato Esteso PE00000013—"FAIR—Future Artificial Intelligence Research"—Spoke 1 "Human-centered AI" project, funded by the Italian Ministry of University and Research under the NextGenerationEU program; and by the European Community programme under the funding scheme ERC-2018-ADG (Grant 834756) "XAI: Science and technology for the explanation of AI decision making." The authors' opinions do not necessarily reflect those of the funding agencies.
Authors' Contact Information: Mattia Setzu, Dipartimento di Informatica, Università di Pisa, Pisa, Italy; e-mail: mattia.setzu@unipi.it; Anna Monreale, Dipartimento di Informatica, Università di Pisa, Pisa, Italy; e-mail: anna.monreale@unipi.it; Silvia Corbara (Corresponding author), Scuola Normale Superiore, Pisa, Italy and Istituto di Scienza e Tecnologiedell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy; e-mail: silvia.corbara@isti.cnr.it; Alejandro Moreo, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy; e-mail: alejandro.moreo@isti.cnr.it; Fabrizio Sebastiani, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy; e-mail: fabrizio.sebastiani@isti.cnr.it.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2024 Copyright held by the owner/author(s).
ACM 1556-4711/2024/6-ART44
https://doi.org/10.1145/3654675

1 Introduction

Authorship analysis can be broadly defined as "any attempt to infer the characteristics of the creator of a piece of linguistic data" [33, p. 238], where these characteristics include the author's biographical information (age group, gender, mother tongue, etc.) and identity. Since the pioneering work of Mosteller and Wallace [54], the field of authorship analysis has made extensive use of computational methods, particularly those tailored to text mining and **Natural Language Processing (NLP)**, thereby contributing to the work of many scholars in the field of cultural heritage and providing them with new tools and perspectives in the study of historical documents of different languages and periods.

One important group of tasks in authorship analysis goes under the name of **Authorship Identification (AId)** and concerns the study of the true identity of the author of a written document of unknown or disputed paternity. The three main tasks in the AId group are **Authorship Attribution (AA)**, **Authorship Verification (AV)** and **Same-AV (SAV)**. In AA [40, 67], given a document d and a set of candidate authors $\{A_1, \ldots, A_m\}$, the goal is to identify the most likely author of d among the set of candidates; AA is thus a single-label multi-class classification problem, where the classes are the authors in $\{A_1, \ldots, A_m\}$.[1] In AV [68], given a candidate author A and a document d, the goal is to infer whether A is the real author of d or not; AV is thus a binary classification problem, with A and $\overline{A}$ as the possible classes. In SAV [16], given two documents d_1 and d_2, the goal is to infer whether they are written by the same (possibly unknown) author or not; SAV is thus also a binary classification problem, with *SameAuthor* and *DifferentAuthor* as the possible classes. All of these tasks are usually approached as *text classification* tasks, whereby a supervised machine learning algorithm, using a set of labelled documents, is used to train a classifier to perform the required prediction task.

A close analysis of the AId literature reveals that, while researchers have devoted significant effort to test the relative performance of different learning methods in AId tasks, to check the usefulness of different types of features for capturing written style and to apply the techniques thus developed to a number of AId case studies, little to no attention has been paid to providing users with *explanations* regarding the predictions of the above algorithms. This is unsatisfactory, since machine-learned classifiers are usually opaque (i.e., they provide predictions but do not provide intuitive explanations of the reasons behind these predictions), and most users of AId systems hardly assign any value to a "bare" automated prediction, being instead interested in understanding the *reason* behind the system's prediction.

The goal of this work is to make progress towards filling this gap, by carrying out an in-depth analysis of the suitability of a set of well-known general-purpose **eXplainable Artificial Intelligence (XAI)** methods, i.e., methods for explaining the predictions of a machine-learned system, to the three main AId tasks. The users of AId systems that we have in mind are scholars working in cultural heritage (such as philologists, historians, linguists), who are typically not machine learning experts. Note that, in this research, our goal is not to devise a new XAI method but to examine the suitability of existing XAI methods to AId tasks and to the user group identified above.

This article is organised as follows. After a discussion on (computational) AId and on the importance of explanations for the predictions issued by machine-learned AId systems (Section 2), in Section 3, we survey relevant related work. In Section 4, we explain the three major classes of methods for explaining the predictions of machine-learned systems that we explore in this article, i.e., feature ranking, transformer probing and factual/counterfactual selection. In Section 5, we explain our experimental setup, while in Section 6, we showcase the application of the aforementioned methods to AId tasks and analyse their relative benefits for the specific purposes within the cultural heritage domain. Section 7 concludes, pointing at avenues for future research.

[1] In classification, *multi-class* (as opposed to *binary*) means that there is a set of $m > 2$ classes to choose from; there are instead just 2 classes to choose from in the binary case. On the other hand, *multi-label* (as opposed to *single-label*) means that zero, one or more than one class may be attributed to each item; exactly one class must instead be attributed to any given item in the single-label case.

2 Background: AId and the Need for Explanations

As mentioned in the introduction, AId tasks are usually tackled as text classification problems [1] and solved by using supervised machine learning algorithms. For instance, to solve the AV task, a machine learning algorithm trains a binary "A vs. $\overline{A}$" classifier using a training set of labelled texts, where the training examples labelled as A are texts by the candidate author and the training examples labelled as $\overline{A}$ are texts by other (ideally, stylistically similar) authors.

Generally speaking, AId techniques attempt to spot the "hand" of a given writer, thus distinguishing their written production from the production of others. The core of this practice, also known as "stylometry" [28], does not rely on the investigation of the artistic value or the meaning of the written text but on a quantifiable characterisation of its style. This characterisation is typically achieved via an analysis of the frequencies of linguistic events (also known as "style markers") occurring in the document of interest, which are assumed to remain more or less constant throughout the production of a given author, while conversely varying substantially across different authors [33, p. 241]. These linguistic events are often of seemingly minimal significance (such as the use of a punctuation symbol or a preposition) but are assumed to be out of the conscious control of the writer and hence to occur in patterns that are hard to consciously modify or imitate.

AId methodologies are profitably employed in many fields, ranging from cybersecurity [65] to computational forensics [14, 42, 56, 60]; yet another important area of application for AId techniques is the cultural heritage field, which is the focus of the present article. Indeed, researchers might use AId techniques to infer the identity of the authors of texts of literary or historical value, whose paternity is unknown or disputed. In these cases, unknown or disputed authorship may derive from authors attempting to conceal their identity (whether for a desire to remain anonymous or for the malicious intent to disguise themselves as someone else), or simply as a result of the passing of time, which is a common occurrence when dealing with ancient texts [6, 34, 39, 64, 70, 72].

While many efforts in AId have focused on testing the accuracy of different learning algorithms (see e.g., the surveys by Stamatatos [67], Juola [33], Grieve [21] or the annual editions of the popular PAN shared task [36, 69]), or on proposing new sets of features that these algorithms could exploit [16, 62, 75], or simply on applying known techniques to case studies of literary interest [6, 34, 39, 64, 70, 72], little or no effort has been devoted to endowing these systems with the ability to generate explanations for their predictions.

This fact represents indeed a very important gap in the literature and a hindrance to a more widespread adoption of these technologies in cultural heritage and other fields. The ability to provide justifications for their own predictions is a very important property for machine-learned systems in general, and even more so when these systems are involved in significant decisions-making processes, such as deciding on the authorship of written documents, with all its legal and ethical implications. We might even claim that *an authorship analysis system is almost useless, unless it is endowed with the ability to explain its own decisions*. Indeed, when such a system is applied to, say, determining the authorship of an important literary work of controversial paternity [17, 18, 37, 72], it is paramount that the prediction is presented to the domain experts along with a comprehensive explanation of the reasons why the system made such a prediction. There are two main reasons for this.

The first reason is that a domain expert who has devoted a sizeable intellectual effort to determining the authorship of a given document is unlikely to blindly trust the prediction of an automatic system, unless the possibility to examine the reasons of its prediction and/or the inner working of the system is provided [58]. Indeed, a domain expert might want to check whether the AId system is actually focusing on the writing style of the document under investigation (and on features deemed important by the expert), and that the system is not instead focusing on other possibly misleading aspects of the document, such as its topic. A similar argument can be applied to an automated prediction meant to be used as evidence in a criminal case: in this case, it would be necessary to put the judge and the jurors in the condition to form their own opinion regarding the output of the automatic system, by giving them as much information as possible on the system and on the reasons that have led it to make that specific prediction [11, 25, 42].

The second reason is that, in the case of cultural heritage applications, the knowledge regarding the process of an AId system might inspire the domain expert with new possible working hypotheses that had not been considered before (e.g., by highlighting a linguistic event that prominently occurs in one author's works but not in the production of other authors). In this regard, it is interesting to note that, in authorship analysis studies, the domain expert and the automatic system often employ complementary methodologies. For instance, when performing authorship analysis for cultural heritage texts, a domain expert may (1) analyse the historical facts described in the text and check whether a certain candidate author could possibly have been aware of these facts; (2) analyse the stand that a candidate author takes towards a certain issue and check whether this stand is compatible with what we already know about the author's ideas and (3) in general, bring to bear their knowledge of a given candidate author, of the historical period in which the candidate operated, of the cultural *milieu* that surrounded the candidate, and decide whether all these are, or are not, compatible with the hypothesis that the candidate may be the real author of the disputed document. Current automatic AId systems can do none of the above. More in general, while the domain expert can use *exogenous* real-world knowledge (i.e., knowledge external to the document), an automatic AId system is typically only able to use *endogenous* knowledge (i.e., knowledge extracted from the document—plus potentially some external linguistic knowledge, in the form of dictionaries or sets of word embeddings or similar). However, an automatic system is capable of doing fine-grained statistical analyses that would be difficult, or impossible, for any human to perform;[2] stylometric analysis is indeed one such type of analysis, where an automatic system can analyse a huge amount of linguistic traits of apparently minimal significance that, altogether, can define an author's style. In other words, this "lower-level" analysis of the text provides a useful complement to the "higher-level" analysis that the domain expert carries out.

To summarise, the role of an automatic system in tasks such as AId should not be that of an opaque, cryptic oracle, but that of a tool that supports the domain expert, who is in charge of delivering the final authorship hypothesis. In other words, the automatic system should be integrated within a pre-existing workflow; by doing so, it could be perceived not as an attempt to replace the domain experts, which would understandably elicit a negative reaction on their part, but as an attempt to support them in their job.

There are three main obstacles in devising an explainable AId system. First, the vector space typical of text-related prediction tasks usually has a very high dimensionality; indeed, many of the tools that have been developed in the XAI literature are more suited to the low dimensionality typical of structured data. Second, the linguistic events employed as features in AId tasks are usually of minimal significance (e.g., the occurrence of a specific character 3-gram), a significance that may be hard to grasp for the person to whom the explanation is addressed; this is indeed an intrinsic problem stemming from the different approaches that humans and machines employ when facing AId tasks. The third obstacle (which is inherently related to the first two) is that, in text-related prediction tasks, a prediction is obtained thanks to the contribution of *many* features, all representing linguistic events of minor importance; in other words, it is difficult to isolate one or few such events that are responsible for the final prediction by themselves. Moreover, as noted by Halvani [25], the bag-of-features representation, which is usually employed in AId tasks, loses the contextual information of the individual features, making it difficult to understand how such features relate to each other with regard to the final output. This means that presenting the user with a concise explanation of the prediction (in terms of the features that have contributed to it) is usually a very difficult matter.

3 Related Work

In recent years, XAI has gained more and more attention in the NLP and text mining communities; see, e.g., the general surveys on XAI by Hamon et al. [27], Guidotti et al. [23], Carvalho et al. [12] and Linardatos et al. [46],

[2]Domain experts sometimes do, in fact, analyse the same features as automatic systems, e.g., they may notice that an author tends to use a specific spelling of a given word, or that an author tends to start a sentence with a certain word or sequence of words. However, it is undeniable that a human carries out this type of analyses with greater difficulty and only on a limited scale.

the surveys on XAI applied to NLP and text classification by Lertvittayakumjorn and Toni [44], Danilevsky et al. [19], and the recent proposals discussed in the works of Rajagopal et al. [57], Wiegreffe and Pinter [73], Liu et al. [47] and Gu et al. [22].

XAI methods are usually sub-divided into *local explainers* and *global explainers*. A local explainer is a method that returns an explanation for a specific prediction of the classifier, while a global explainer is a method that returns an explanation of the behaviour of a classifier in general. Understandably, each approach has its own pros and cons, but both can be used to offer insight in the rationale of a classification decision. Since the two approaches focus on different kinds of information, they can be complementary, and multiple local explanations can be combined to gain a general understanding of the behaviour of the classifier [10, 51].

Despite the growing interest that XAI has witnessed in recent years, little or no attention has been given to its application to AId, possibly also due to the difficulties mentioned at the end of Section 2. Some recent attempts towards providing explanations for the predictions of text classifiers consist of creating a saliency mask [23], visually displaying the textual elements most important for the classifier's decision directly within the document (this is thus an example of a *local* explainer). An example is provided by the AV study by Corbara et al. [17] regarding the *Epistle to Cangrande*, whose Dantean paternity has been long debated. In this study, the authors highlight the 90 paragraphs of the *Epistle* with different colours based on the classification scores obtained when classifying each paragraph individually (see Figure 1). This visualisation serves a dual purpose; On the one hand, the score assigned to each paragraph serves as an estimate of its contribution to the overall prediction for the entire document, while on the other hand, in light of theories suggesting that only certain sections of the *Epistle* may be spurious (see Section 5.1 for details), it enables a finer-grained analysis from this perspective. Theophilo et al. [71] obtain a similar effect at the feature level by adapting the popular **Local Interpretable Model-Agnostic Explanations (LIME)** algorithm[3] to process character 4-grams. To offer an explanation for the decisions of his compression-based SAV algorithms, Halvani [25] proposes to colour two texts based on their differences (the higher the discrepancy, the stronger the colour), thus providing an intuitive and straightforward representation of areas of the texts that play a more important role in the prediction. Halvani [25] also proposes to display the element-wise Manhattan distance between the two values of the same feature, which represents how much the feature influences the similarity of two documents. Alternatively, when working with architectures based on neural networks, researchers have focused on the visualisation of the attention weights [10] or on the derivative of the output given the embedding of a word in the input [66].

While saliency maps and similar visualisation devices may help the user to focus on areas of the text that have played an important role in the system's decision, they are incomplete explanations, since they place on the user the burden of understanding *why* the system has reached exactly that decision. An alternative method consists of ranking the features used by the classifier by their importance (this is thus an example of a *global* explainer), where this "importance" can be assessed in different ways. For example, in their work on native language identification (the task of detecting the native language of the author of a text), Berti et al. [7] use the weights associated to the features in a linear classifier as indications of which features best separate the classes, since the absolute value of these weights is proportional to the discriminative power of the respective features.[4] Other studies, such as the one by Sapkota et al. [61], assess the effect of different feature types (e.g., character

[3]Specifically, given a non-linear model Φ and an instance x, LIME [58] employs a perturbation algorithm that generates a neighbourhood of x. Leveraging this neighbourhood and the prediction made by Φ on said neighbourhood, LIME learns a new linear classifier that is a good approximation of Φ (i.e., it outputs similar predictions). This linear classifier is intrinsically interpretable, since it provides coefficients for each input feature, which allows the user to understand which features have contributed most to the prediction of Φ on x. The original LIME formulation for text is restricted to using word and character unigrams as interpretable components.

[4]For example, in the *Spanish* versus *NonSpanish* classifier, the weight of *especial*, a misspelling of the English word *special*, is high and positive, leading to the class *Spanish*, since native speakers of Spanish have a tendency to prefix a spurious *e-* to many English words starting with an *s*, due to an interference from their mother tongue. As a result, when a text classified as *Spanish* contains the term *especial*, this occurrence constitutes a (partial) explanation of this classification decision.

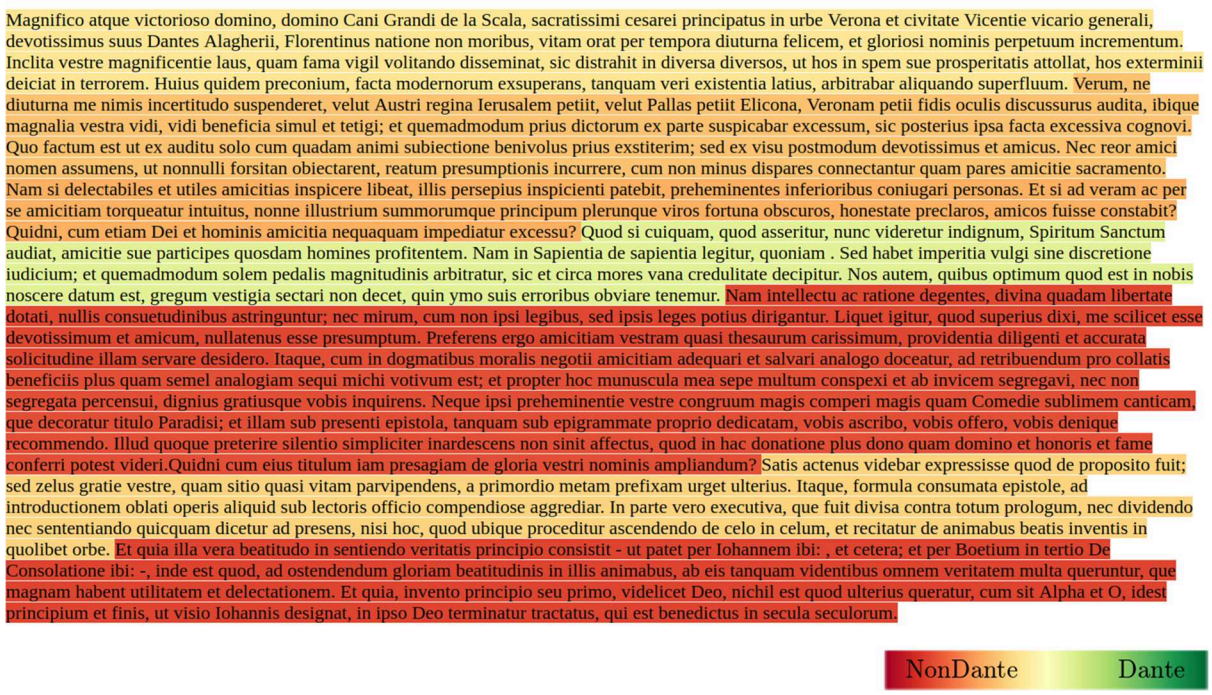

Fig. 1. Visualisation of the *Epistle to Cangrande* (from Corbara et al. [17]), whose attribution to Dante Alighieri is uncertain; paragraphs on the red side of the spectrum are those that the authorship classifier believes to be "less Dantean," while those on the green side of the spectrum are those that the authorship classifier believes to be "more Dantean."

n-grams) by evaluating the performance of a classifier trained after subtracting the feature types under study. For neural networks, and particularly for **Convolutional Neural Networks (CNNs)**, an approach similar to the above consists of listing the input elements that generate the highest activation values aggregated over all filters, or the input elements that generate a significant activation value for the highest number of filters [66]. However, these approaches are admittedly a long way from constituting satisfactory explanations for AId decisions, because they provide explanations that are partial and/or difficult to grasp for a scholar who is not a machine learning expert.

Another widely used technique consists of displaying the documents of interest in a bidimensional space obtained through dimensionality reduction (e.g., via **Principal Component Analysis (PCA)**), to provide a visual idea of the characteristics of the data [8, 20, 37]. As an example, in Figure 2, texts are mapped to a bidimensional space together with the words whose use most differentiates the candidate authors. A reader is thus able to see how texts by the same author are clustered together, and how the classifier has found the use of specific words to be characteristic to specific authors.

4 Methodology

As briefly discussed in Section 3, there are several general-purpose XAI methodologies that allow to better understand a trained classifier or a specific prediction. In this work, we experiment with three of these options, analysing their suitability to AId tasks in general, and to the specific public of cultural heritage professionals in particular. Within this context, we discuss the possible contribution of both global explainers and local explainers.

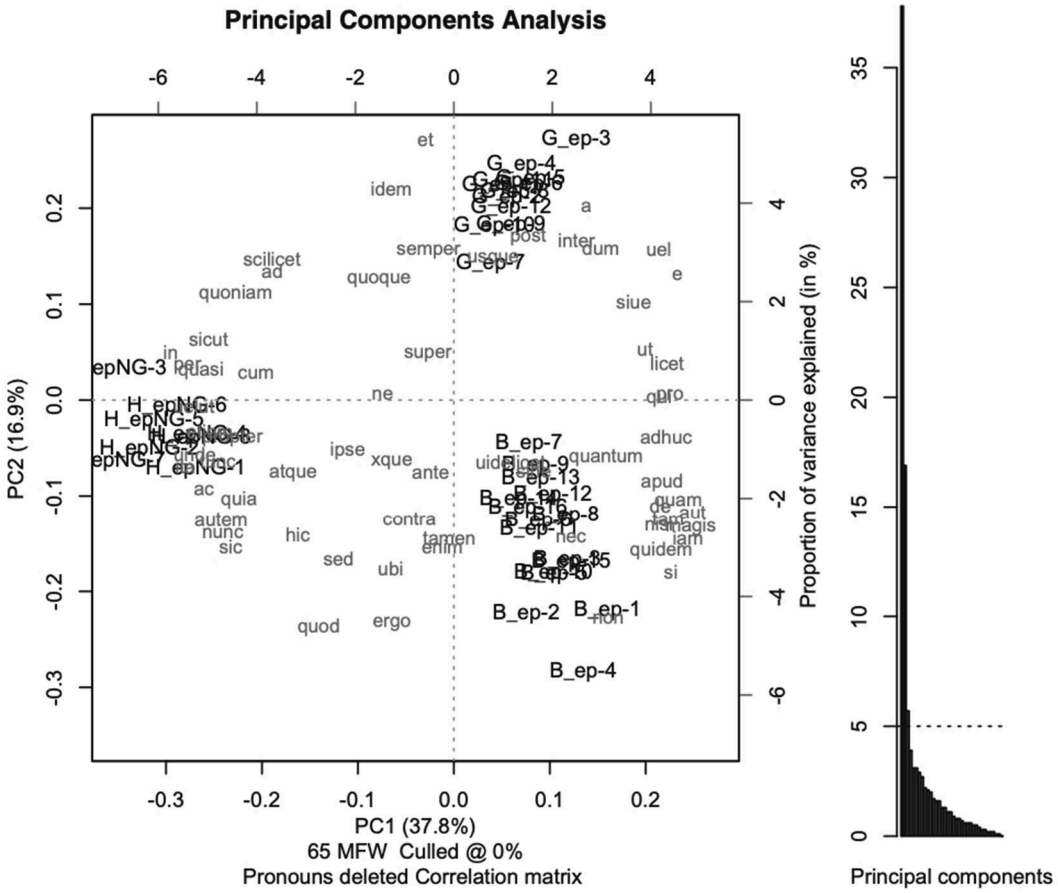

Fig. 2. Visualisation (from Kestemont et al. [37]), obtained via PCA, of texts from three different authors (here identified by the letters B, G and H), showing that the technique used separates them well; strings *A*_ep-*n* identify the *n*th text by author *A* in the dataset. The words that are located near the texts by author *A* are the ones that occur more frequently in the texts by *A* than in the texts by the other two authors.

In Section 4.1, we show how to gain insight into the features that a linear classifier deems most important for the classification task; we do so by directly employing the weights of the trained model and show how to obtain both global and local explanations. Note that, in the case of linear classifiers, and more generally in the case of "classic" machine learning methods (such as **Support Vector Machines (SVMs)**, logistic regression, decision trees and shallow neural networks), the features used to train the learning algorithm are identified *a priori* by the researcher, in the so-called "feature engineering" phase. In other words, the model, and thus the explanation, is constrained to use only the features (and combinations thereof) defined by the researcher.

Conversely, deep neural networks are by design able to discover novel discriminative features from the data and can thus carry out the feature engineering phase autonomously. While some feature ranking solutions for deep-learning models are present in the literature [52], they are extremely sensitive to the input data and are based on a number of assumptions (such as uniform data distribution) which are often unrealistic [29]. Thus, in the case of AId solutions based on deep learning, rather than providing hardly interpretable and/or unreliable

explanations, we employ a more sophisticated solution, known as "model probing," applied to a RoBERTa-based model, to obtain global explanations (see Section 4.2).

Finally, in Section 4.3, we discuss how to extract prototypical examples from the training set. These representative examples provide the user with instances that the model deems most similar to the test example and that belong to the same (or different) class as the model prediction, exposing the internal representation learned by the model, and thus acting as local explainers.

4.1 Feature Ranking

A common strategy for offering a global explanation (i.e., providing a general understanding of the behaviour of the classifier) is to show the features the classifier mostly focuses on during prediction, as already explained in Section 3. In these XAI methods, given a trained model, each feature is associated to a score, and the employed features are presented in decreasing order based on their score.

The score of a feature can be obtained in various ways. In the case of linear models, the most direct way is to employ the coefficients (or weights) of the classifier. By design, a binary linear classifier has the form $h : \mathbf{x} \cdot \mathbf{w} + b$, where $\mathbf{x}$ is the feature vector that represents data item x, $\mathbf{w}$ is a vector of weights learned from the training data (one weight for each feature), and b is the intercept of the function; item x is assigned the positive class when $h(\mathbf{x}) > 0$, and it is assigned the negative class otherwise. In the application scenario we discuss in Section 5.2, where the feature vectors fed to the linear SVM are positive definite, the higher the absolute value of the weight w_i associated to the ith feature, the larger is the contribution of such feature towards the prediction.

Note that linear methods compute a set of coefficients for each binary classification problem (which is the case of SAV and AV). In the case of multi-class classification (which is the case of AA), the learner computes a set of coefficients for each class; for prediction explanation purposes, these sets must be examined individually.

The coefficients of the model can also be used to obtain a form of local explanation: by multiplying the feature value extracted from a test example by the correspondent coefficient, we can assess how much the feature contributed to the prediction for the document. This allows us to understand the model both on a global level and on a local level, explaining both how the model *generally* reasons, and how it reasons on *specific instances*.

There are more sophisticated ways to get feature scores (and they are a mandatory resource in the case of non-linear methods, such as neural networks). For example, the **SHapley Additive exPlanations (SHAP)** algorithm [52] is a widely used family of algorithms for model-agnostic XAI (meaning that it can be applied to any learning algorithm). Unlike LIME, SHAP performs perturbations on the feature set, then queries the model to estimate the importance of each feature by leveraging the change in prediction that each perturbation has produced on the outcome of the model. By default, SHAP scores are local explanations, but the scores from multiple examples can be averaged to reach a global explanation. However, since the number of features employed in textual settings is usually extremely high, the number of perturbations that the SHAP algorithm should compute would be exponentially large, making it computationally prohibitive; in these cases, perturbations can be approximated through random sampling, but this is only a band-aid solution.

Even though in this case the features employed by the classifier are defined *a priori*, an explanation of the type described above can be extremely useful for the scholar. For instance, among the tens of thousands character n-grams that can be extracted from the texts, what are the most discriminative for the author(s) of interest? Thanks to the explanations mentioned above, in theory a scholar might find out, e.g., that a certain author tends to avoid certain patterns of characters, or vice versa has a preference for specific syntactic constructs.

4.2 Probing

As already shown, obtaining an indication of the importance of the features by using the feature weights of the model is straightforward for linear classifiers; however, it is not as straightforward for non-linear classifiers, such as the ones exploiting neural networks. Nevertheless, explainability is even more important for these "black-box"

architectures, for at least two reasons. On the one hand, since the features are not identified *a priori* by the designer (as it is instead the case with "traditional" learners), an explanation method may allow the scholar to check if the classifier is using the features that they indeed deem important for the recognition of authorial style, and thus it might help them to trust the classification system (see Section 2). On the other hand, an explanation method may allow the scholar to check if the system has discovered new features that are interesting for identifying the authorship of written documents, and that can be interesting to investigate further.

Indeed, many recent studies have tackled the far-from-trivial task of developing XAI techniques that can show which features these models are actually leveraging in their predictions. Among these studies, the method of "probing" has recently gained vast popularity [5]. Probing allows a user to understand if a certain feature of interest (not defined *a priori* by the designer) has been learned and used by the model. For instance, probing has been used to discover that some famous pre-trained language models, such as **Bidirectional Encoder Representations from Transformers (BERT)** and **Robustly Optimized BERT Approach (RoBERTa)**, are not really capable of understanding basic mathematical concepts [45] but seem to have learnt some form of common sense directly from data [32]. The main idea behind the process of probing is to input the latent representation computed by the neural network model (from now on, the *main model*) to a second, very simple model (from now on, the *probe*), whose task is to predict whether the feature of interest is present in the latent representation or not. Given the simplicity of the probe and the complexity of the representation, the underlying assumption is that, if the presence of a feature can be found even by a simple probe, then that feature is encoded by the main model in the latent representation.

Specifically, given a non-linear model Φ and a hypothetical feature f, to probe the model (i.e., to try to provide an answer to the question: "Does Φ internally learn from f?"), we create a dataset of the form $\{\phi(x_i), f(x_i)\}_{i=1}^{n}$, in which x_i is a textual document, $\phi(x_i)$ is the internal representation of x_i created by Φ, and $f(x_i)$ is a function that characterises x_i in terms of the feature f. For example, $f(x_i)$ may be binary, returning 1 or 0 to indicate that a given feature is present or absent in x_i, respectively. Conversely, $f(x_i)$ may be categorical, returning a class label in the range $\{1, \ldots, n\}$ when the characteristics of f in x_i allow us to distinguish among n different groups of documents (see Section 6.2). We then train a linear model with this dataset, and we use the resulting classifier to estimate (e.g., via cross-validation) the extent to which the characteristics encoded by the feature under study are directly learnable from the internal representation of Φ. We repeat this process for every feature we conjecture could be playing a role in the decision function that the model implements.

In particular, in Section 6.2, we exemplify this approach by developing five types of probing:

- ***Part-of-Speech*** (*POS*) *n-grams*: We probe the model for features extracted from the concatenation of POS tags, which are nowadays a standard feature type for AId (see e.g., [31]).
- ***Syllabic Quantities*** (*SQ*) *n-grams:* We probe the model for features extracted from the concatenation of SQ, which have been first proposed for AId tasks in the Latin language by Corbara et al. [16].[5]
- *Word lengths*: We probe the model for the frequency of word lengths, which have been employed as features in the AId field since the proposal by Mendenhall [53].
- *Function words*: We probe the model for the frequency of function words, which are widely employed as features in the AId field [8, 35].
- *Doc genre*: We probe the model for the genre of the document, to see whether the model encodes the characteristics of the genre into the latent representation.

Given that the probing approach does not rely on specific feature types, domain experts can explore any feature they might find interesting.

[5]In the Latin language, words can be divided into syllables, which can be long or short depending on their quantity; see Corbara et al. [16] for more information.

4.3 Selection of Factuals and Counterfactuals

Given a prediction $\hat{y}$ on an item $\mathbf{x}$, it might be useful for a domain expert to check the items that the classifier considers most similar to $\mathbf{x}$. By doing this, the domain expert can (1) judge whether the similarities detected by the classifier align with the expert's knowledge (e.g., the classifier considers documents from the same historical period similar) and (2) discover possible similarities among the documents that the expert might have been unaware of, but the classifier has brought to light.

To this aim, a standard method is to retrieve the training instances that are most similar to $\mathbf{x}$ according to the model. Among these training items, some would have the same label $\hat{y}$ that $\mathbf{x}$ has, while others would have a label different from $\hat{y}$. The former items are called *factuals*, and the scholar may find them useful when trying to understand the characteristics of the class that has been assigned to $\mathbf{x}$, while the latter are called *counterfactuals*, and they may be useful in allowing the scholar to gauge the minimal requirement for the classifier to predict a class $y \neq \hat{y}$.

In the case of linear models with no internal representation, the similarity of two instances can be computed by applying any standard similarity measure directly to the input vectors; in the case of deep-learning models, it can instead be computed by applying the similarity measure to the latent representations of the instances. In the case of linear models, it is also possible to easily spot the features that most contributed to the similarity of the two items.

5 Experimental Setup

In this work, we exemplify the use of the well-known XAI techniques discussed above by drawing examples from the three major AId tasks, i.e., AA, AV and SAV. We show how these XAI methodologies deal with predictions issued in each of these three tasks on a dataset of medieval Latin [18]; this dataset well exemplifies the kind of data that users from the cultural heritage field may have to deal with.

In the following paragraphs, we present our experimental setup. In particular, in Section 5.1, we present the dataset we use, while in Section 5.2, we explain our classification methodology, along with the learning algorithms we employ.

The Python code to replicate our experiments is available at: https://github.com/silvia-cor/XAId.

5.1 Dataset

In this study, we employ the MedLatin dataset developed by Corbara et al. [18]. This dataset was created to study, via computational AV methods, the authorship problem of the *Epistle to Cangrande*.

The *Epistle* is the thirteenth of the letters from Dante Alighieri's epistolary corpus that have survived until our times. Written in Latin, it is addressed to Cangrande I, and contains an important exegesis of Alighieri's *Divine Comedy*, and in particular a commentary of the first few lines of its third part, the *Paradiso*, which would make it the only analysis we have by Dante Alighieri of his own masterpiece. However, the debate regarding its authenticity, started in the 19th century, has not been resolved yet. Scholars are divided between those who believe that the *Epistle* is a partial or complete forgery—these authors point out numerous passages in the composition where the logical sequence of discourse is cumbersome, or even incoherent with itself or with other writings by Alighieri [13], and highlight a profound dissimilarity between the first and second portions of the letter in terms of themes, style and rhythm [24]—and those who support its authenticity—these authors stress a lexical coherence and an inner cohesive logic in the entire composition [3, pp. 280–281], and observe that, paradoxically, a forger would have followed more closely Alighieri's prose [2] and would have refrained from exposing non-traditional and potentially controversial interpretations [3, 63].

The authors originally divided the dataset into two sub-datasets, MedLatinEpi and MedLatinLit, both containing works in medieval Latin prose, mostly dating to the 13th and 14th centuries; MedLatinEpi is composed of 294 texts of epistolary genre, while MedLatinLit is composed of 30 texts of different types, especially literary works

and chronicles. For this project, we use documents written by only five authors: Dante Alighieri and Giovanni Boccaccio (who have documents in both sub-datasets), Pier della Vigna (the author that contributes most to MedLatinEpi in terms of total number of words), Benvenuto da Imola and Pietro Alighieri (the two authors that, after Giovanni Boccaccio, contribute most to MedLatinLit in terms of total number of words). We delete any direct quotation from other authors, and we delete the parts in languages other than Latin (both are explicitly marked in the MedLatin texts). Following Corbara et al. [16], we also divide each text into sentences, where a sentence consists of at least five distinct words (we attach shorter sentences to the next sentence in the sequence or to the previous one in case the sentence is the last one in the document); we use each non-overlapping sequence of 10 consecutive sentences as a textual example. By doing this, we end up with 2,729 text examples in total. We randomly split the corpus into a training set (90% of the examples) and a test set (the remaining 10%) in a stratified fashion.

For the SAV task, we do not employ all the pairs of examples that can be created within the training and test sets, since their number is excessive, and using them all would drastically slow the computation. In particular, given a set of authors $\{A_1, \ldots, A_z\}$, we create n *SameAuthor* pairs for each author A_i (each consisting of two random texts by A_i) and m *DifferentAuthor* pairs in total (where a *DifferentAuthor* pair consists of one random text for each of two different random authors in $\{A_1, \ldots, A_z\}$); the pairs are unique. In our experiments we set $n = 5,000$ and $m = 25{,}000$ for both the training set and the test set; therefore, both the training set and the test set are balanced.

For the AV task, we select Dante Alighieri as the author of interest, in line with the experiments reported by Corbara et al. [18].

It is worth noting that all the XAI methods we discuss in this article are independent of the specific characteristics of the dataset being analysed, such as the number of authors involved, the genre of the documents, or the period that the corpus dates back to. While the features extracted for generating vectorial representations of medieval Latin documents may largely differ from the features extracted for other languages (e.g., modern English), this difference has no impact on the usability of XAI techniques.

5.2 Learning Methods

In this study, we experiment with offering explanations for the output of AId systems trained by one representative "classic" machine learning method and by one representative deep-learning method.

For the former, we employ a linear SVM, a very popular learner in AId tasks [38, 76]; we use the implementation available from the scikit-learn library [55]. We fine-tune the hyperparameter C (with values in the range $[0.001, 0.01, 0.1, 1, 10, 100, 1{,}000]$) by performing three-fold cross-validation on the training set. To train the algorithm, we compute the TfIdf values of all character n-grams with $n \in \{2, 3\}$, which is a common strategy in AId tasks (see e.g., the 2019 PAN shared task [38]). We then perform feature selection by selecting the k most relevant features via χ^2, with $k = 1{,}000$. We tackle SAV in the style of Corbara et al. [15], i.e., we create a single feature vector by computing the absolute difference among the feature values of the two documents that make up the document pair, and label the pair as either *SameAuthors* or *DifferentAuthors*.

For the deep-learning experiments, we employ a RoBERTa model [49] from the HuggingFace Transformers library [74] specifically trained with Latin data.[6] We fine-tune the model for five epochs on the training set, employing the AdamW optimiser [50] with the initial learning rate set to 0.0001, and cross-entropy as the loss. For the SAV task, note that RoBERTa is able to directly classify a sequence of two texts; it is sufficient to concatenate the two texts, separated by the appropriate separator token [SEP]. Note also that RoBERTa works with a fixed maximum length of 512 tokens; we thus truncate the textual samples accordingly.

In Table 1, we report the evaluation results for each model and for each task. As we can see, both algorithms show very high performance in all the tasks. Interestingly, although the RoBERTa transformer performs nearly as

[6]Documentation is available at: https://huggingface.co/pstroe/roberta-base-latin-cased3.

Table 1. Evaluation Results for the Two Classifiers We Employ (SVM and RoBERTa)

	SVM		**RoBERTa**	
	Acc	F_1	*Acc*	F_1
SAV	0.836	0.838	**0.957**	**0.956**
AV	**0.985**	0.894	**0.985**	**0.900**
AA	**0.989**	**0.981**	0.978	0.963

Acc, accuracy. The evaluation measures we use are *Acc* and F_1. For SAV and AV, which are binary tasks, F_1 is defined in the standard way, while for AA, which is a single-label multi-class task, the reported F_1 values are obtained by macro-averaging (i.e., they are computed as the arithmetic mean of the class-specific F_1 values). Micro-averaged F_1 values are not reported here, since micro-averaged F_1 and *acc* are the same measures in single-label multi-class classification. The best model result for each task is in bold.

Table 2. Bottom Five and Top Five Features of the SVM Classifier for the SAV Task by Coefficient Value (coef)

***n*-grams**	**coef**
"gab"	0.193
"tto"	0.179
"mac"	0.178
"mbi"	0.175
"aia"	0.171
...	
"auc"	−1.454
"_ai"	−1.586
"ait"	−1.725
"ae_"	−3.976
"ae"	−4.792

White spaces in the feature names are indicated with "_."

well as the SVM classifier in the AV task and outperforms it in the SAV task, it exhibits slightly lower performance than the SVM classifier in the multi-class setting of AA.

6 Tools for XAI: A Comparative Analysis

We here present our results divided by type of explanation, namely feature ranking (Section 6.1), probing (Section 6.2) and factual–counterfactual selection (Section 6.3).

6.1 Feature Ranking for SAV

In Table 2, we show the top five and bottom five features by coefficient value for the SVM that we have trained for the SAV task. Note that we only exemplify this method as applied to SAV, but the considerations we make here also apply to AV and AA.

In our case, all the feature values are positive, since we employ TfIdf values, and the intercept is positive as well (2.29); thus, features associated with positive weights are indicative of the positive class (*SameAuthor*), and features associated with negative weights are indicative of the negative class (*DifferentAuthor*). The reader might find this notion confusing: since features associated with positive weights are indicative of the positive class, and since the feature values we employ for SAV are the result of the absolute difference between the original feature vectors, does it mean that higher differences (i.e., higher feature values in SAV) are associated with the two documents being by the same author? Indeed, this is counter-intuitive. We can speculate that these features are not discriminative in the common sense but act as a threshold: in order for a textual example to be classified as negative, it must have features values (where the weight is negative) that jointly exceed these "non-discriminative" feature values (where the weight is positive). The fact that the positive coefficient values seem relatively smaller if compared with the negative ones might support this hypothesis.

Indeed, we note a disproportion in the coefficient values among positive and negative weights, where the positive values appear smaller than the negative ones in absolute value, slowly decreasing from the first position towards the value zero. Also, we note that the two features with the highest (in absolute value) negative coefficient are "ae_" and "ae," meaning that a discrepancy in the frequency of use of this feature is indeed an indicator that the authors are different or, put it another way, that the frequency of use of these features is quite stable in the production of an author, and is thus a characteristic trait (either because the author tends to use it a lot, or only rarely) of an author. This specific case might be connected with a transitional phase in medieval Latin, when scholars started representing the diphthong "ae" with the single letter "e" instead of with the two separate letters "ae" as it was written through antiquity. A large difference in the frequency of use of these features might thus be an indication that the authors are different, one having a preference for "ae" and the other instead preferring "e."

We can check that the ranked features are indeed useful for SAV by running an ablation experiment, also known as **Iterative Removal of Features (IROF)** [59]. This approach consists of first assessing the performance of the classifier equipped with the entire feature set; then, sequentially, the feature with the highest absolute coefficient value is removed from the feature set (by setting the associated weights to zero in the classifier), and finally the performance of the classifier is re-assessed (without re-training the classifier).

The fact that the model performance drops as we iteratively remove features should come as no surprise. However, a good ranking of features that effectively reflects feature importance would cause performance to degrade much faster (i.e., in fewer iterations) than any other uninformative ranking. This is shown in Figure 3, in which we compare the drop in performance as a function of the number of features removed, by considering our feature ranking (in blue) versus (10 trials of) a random ranking (in orange). The fact that the model becomes a dummy classifier after removing very few features following our ranking proves that the importance criterion is indeed informative.

As already explained in Section 4, we can also employ the coefficients to obtain a form of local explanation, by multiplying the feature value extracted from a document by the correspondent coefficient. We randomly select two examples for *SameAuthor* and *DifferentAuthor*, and show the results in Figure 4. This visualisation highlights the biggest drawback of this XAI technique when applied to textual examples: if we limit the investigation to just a few features, we might risk to convey an incomplete (and thus wrong) picture, especially to a scholar not expert in machine learning. In fact, what we observe is that, on the basis of these examples, the outcome is often contradictory. First, many of the features displaying positive weights happen to be absent in the selected examples. Second, features displaying negative weights behave inconsistently across the examples, e.g., showing relatively high values (examples numbered 2 and 4) or very low values (examples numbered 1 and 3) regardless of their class labels. Hence, it seems clear that restricting the study to only a selected number of features is not enough to convey the full picture of the model's behaviour.

Regarding this XAI approach, we can thus conclude that it contributes to justifying the decisions of the system in the eyes of the domain expert to some degree, but it is also rather problematic; as already noted,

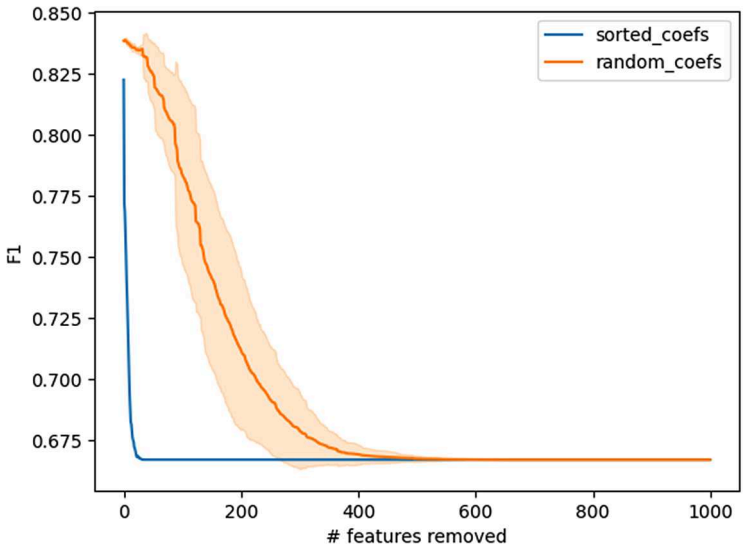

Fig. 3. Results of the IROF test on our SVM classifier for SAV: we iteratively remove one feature at a time, following the descending order of the absolute values of the coefficients (`sorted_coefs`) or a random feature ranking (`random_coefs`). In particular, for the latter we show the mean F_1 value obtained at the n feature removed for 10 random feature rankings, where the coloured shadow is the standard deviation.

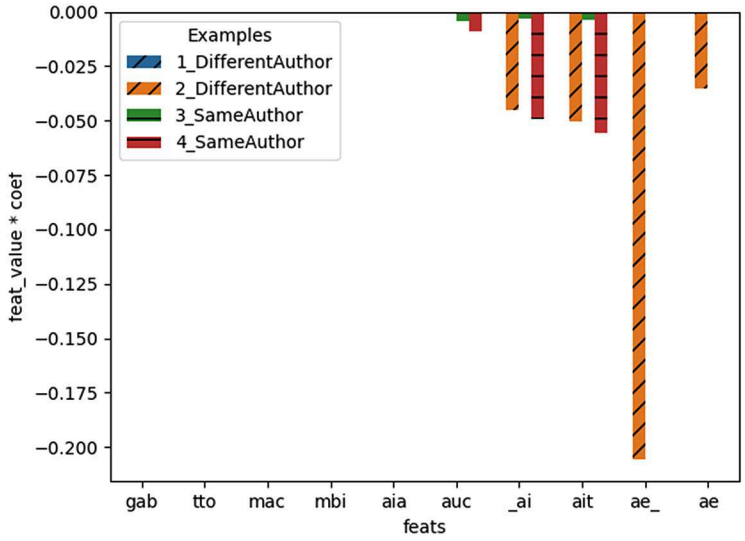

Fig. 4. Local explanations for four examples in the test set, given the features listed in Table 2; we colour the *SameAuthor* class with the "-" pattern and the *DifferentAuthor* class with the "//" pattern. The scores of many features are zeroes for all four examples.

AId tasks (as any other applications of text classification) tend to be characterised by a high number of features, each one providing only a tiny contribution to the final classification decision. In other words, it is unlikely that there are just a handful of features that, by themselves, determine a classification decision. However, as we have shown, limiting the investigation to only a small portion of the features actually employed

by the classifier incurs in the risk to convey a picture that is just too narrow and simplistic. Thus, it is of primary importance to offer an analysis that includes the entirety of the feature set used by the classifier in the most user-friendly way possible, allowing a scholar to personalise and navigate the exploration to its full extent.

6.2 Probing the Transformer for AA

In our experiments, we train a simple Logistic Regression model as a probe, since the classification head on top of the RoBERTa transformer has an equivalent complexity and thus could not gain any more information from the latent representation. In fact, employing non-linear models as probes could be counterproductive: their accuracy might be caused by the memorisation of surface patterns, instead of the information actually captured by the latent representation [30]. We take the training set obtained in Section 5.1 and further split it in a stratified fashion into a training set and a test set for the probe, consisting of 90% and 10% of the instances, respectively. We fine-tune the model hyperparameters via three-fold cross-validation on the probe training set. We then retrain the resulting model on the full training set of the probe before evaluation. In our experiments we probe the main model trained on the AA task. Note that we only exemplify this method as applied to AA, but the method can be applied to AV and SAV as well. However, probes for SAV should be handled carefully, since the RoBERTa latent representation involves both texts.

As illustrated in Section 4.2, we test the transformer with five different probings:

- *POS n-grams*: We probe the model for POS n-grams, with $n \in \{5, 10\}$. In particular, the probe is asked to predict whether a certain POS n-gram is present ($f(x_i) = 1$) or absent ($f(x_i) = 0$) in the document; the labelling function $f(x_i)$ is thus binary. We extract POS tags via the *LatinCy* pipeline of the SpaCy library.[7] We restrict the analysis to the five POS n-grams in the corpus that best discriminate authors, for which the POS n-grams are evaluated via χ^2.
- *SQ n-grams*: This probing is equivalent to the POS n-grams probing, with $n \in \{10, 15\}$; we extract syllabic quantities via the prosodic scanner of the Classical Language ToolKit library.[8]
- *Word lengths*: We probe whether the model takes the word-length distribution into account or not. To do so, we represent each document x_i by means of a histogram $(b_i^{(1)}, b_i^{(2)}, \ldots, b_i^{(B)})$, in which bin $b_i^{(j)}$ accounts for the relative frequency of words of length j (i.e., the fraction of words of exactly j characters) in the document. Then, we cluster the documents thus represented to identify natural groups based on their word-length distribution; we use k-means as our clustering algorithm and choose the optimal number of clusters via the Elbow method within the range $[2, 10]$. Each cluster is assigned a numerical ID, so that the labelling function $f(x_i)$ is categorical in this case, and the probe is asked to label each document with the respective cluster ID. Note that the histogram representation is only used as a means for deciding the cluster to which each document belongs; that is, the probe is still trained and tested using the internal representations $\phi(x_i)$ of the model.[9]

[7] Documentation is available at: https://spacy.io/universe/project/latincy.

[8] Documentation is available at: http://cltk.org/.

[9] We use the implementation of k-means provided by the scikit-learn library [55], which relies on the Euclidean distance (also known as L2) for computing the clusters. This turns out to be sub-optimal in the case of word lengths, since the histograms actually represent ordered distributions, and since L2 does not take into account the order of the dimensionalities of the feature spaces with which it operates. For example, the L2 distance between the pair of (normalised) vectors $v_1 = (1, 0, 0, \ldots, 0)$ and $v_2 = (0, 1, 0, \ldots, 0)$ is as large as the L2 distance between the same vector v_1 and $v_3 = (0, 0, 0, \ldots, 1)$, despite the fact that v_1 and v_2 represent documents that tend to use *very short* words while v_3 instead represents a document that tends to use *very long* words. To counter this, in this case we represent our documents by means of cumulative distributions; in our example, this means that the distance between the (cumulative distributions) $v_1' = (1, 1, 1, \ldots, 1)$ and $v_2' = (0, 1, 1, \ldots, 1)$ turns out to be much smaller than the distance between v_1' and $v_3' = (0, 0, 0, \ldots, 1)$. A different solution would be to adopt, in place of L2, a distance function suitable for ordinal data, such as the Wasserstein distance (also known as Earth Mover's Distance). However, we do not explore this possibility here since the scikit-learn implementation of k-means does not allow customising the distance function.

Table 3. Results for the POS and SQ Probes

	n-gram	*Acc*	*P*	*R*	F_1
POS	adj noun adj noun verb	0.825	0.869	0.825	0.845
	adj noun noun adj noun	0.882	0.922	0.882	0.900
	adp noun adj noun verb	0.821	0.853	0.821	0.836
	noun adj noun adj noun	0.873	0.909	0.873	0.890
	noun adj noun verb verb	0.853	0.873	0.853	0.863
	n-gram	*Acc*	*P*	*R*	F_1
SQ	∪ ∪ ∪ ∪ ∪ – ∪ ∪ ∪∪	0.670	0.684	0.670	0.674
	∪ ∪ ∪ ∪ ∪ ∪ ∪ – ∪∪	0.642	0.647	0.642	0.644
	∪ ∪ ∪ ∪ ∪ ∪ ∪ ∪ ∪–	0.654	0.664	0.654	0.657
	∪ ∪ ∪ ∪ ∪ ∪ ∪ ∪ ∪∪	0.601	0.614	0.601	0.601
	∪ ∪ ∪ ∪ ∪ ∪ ∪ ∪ ∪ ∪ ∪	0.626	0.670	0.626	0.639

Probes try to predict the presence of the given POS *n*-gram or SQ *n*-gram in the latent representation of the model. For SQ we here employ the standard notation where "∪" stands for a short syllable and "–" stands for a long syllable. *P*, precision; *R*, recall.

— *Function words*: We create a probe to check the extent to which the model learns from the frequency of use of the function words. To this aim, we apply a strategy that is similar to the aforementioned case for word lengths. That is, we first represent each document x_i as a histogram $(b_i^{(w_1)}, b_i^{(w_2)}, \ldots, b_i^{(w_B)})$, in which the $b_i^{(w_j)}$ accounts for the relative frequency of function word w_j in x_i. We consider the list of 80 function words for Latin used by Corbara et al. [16].[10] As before, we label each document with the cluster ID to which it is assigned by a k-means algorithm based on the histogram-based representations. The function f is thus again categorical.
— *Genre*: We probe the model for the genre of the documents; in particular, we ask the probe to classify the documents based on the sub-corpus they belong to, *MedLatinEpi* or *MedLatinLit*. As such, we try to assess whether the transformer encodes the stylistic characteristics of documents of epistolary nature ($f(x_i) = 1$) versus documents a different literary nature. ($f(x_i) = 0$); the labelling function $f(x_i)$ is thus binary.

We show the results of POS probing in the first portion of Table 3. The probes show high performance for all the POS *n*-grams considered, with F_1 always above 0.8, indicating that the transformer is likely learning from the syntax of the documents. These results are in line with the current literature on language model probing [48], and confirm that, even in authorship analysis, models leverage POS *n*-grams in downstream tasks. On the other hand, the performance of SQ probing, displayed in the second portion of Table 3, is much lower, with values between 0.6 and 0.7. However, these results indeed show knowledge of the concept of syllabic quantity by the transformer; this is an interesting discovery since, to our knowledge, this is the first work in which this kind of information is sought in the latent space generated by a transformer.

Regarding word lengths and function words, we show the results of the two multi-class classifications in the first and second portion of Table 4, respectively; interestingly, the optimal number of clusters is six for both

[10]The full list of function words is: *a, ab, ac, ad, adhuc, ante, apud, atque, aut, autem, circa, contra, cum, de, dum, e, enim, ergo, et, etiam, ex, hec, iam, ibi, ideo, idest, igitur, in, inde, inter, ita, licet, nam, ne, nec, nisi, non, nunc, nunquam, ob, olim, per, post, postea, pro, propter, quando, quasi, que, quia, quidem, quomodo, quoniam, quoque, quot, satis, scilicet, sed, semper, seu, si, sic, sicut, sine, siue, statim, sub, super, supra, tam, tamen, tunc, ubi, uel, uelut, uero, uidelicet, unde, usque* and *ut.*

Table 4. Results for the Word-Lengths and Function-Words Probes

	# of clusters	*Acc*	*P*	*R*	F_1
Word lengths	6	0.487	0.487	0.487	0.486
Function words	6	0.617	0.628	0.617	0.617

Probes try to predict the word-lengths cluster or function-words cluster in the latent representation of the model. *P*, precision; *R*, recall.

Table 5. Results for the Genre Probes

	Acc	*P*	*R*	F_1
Genre	0.979	0.979	0.979	0.979

Probes try to predict the sub-corpus of the documents (between *MedLatinEpi* and *MedLatinLit*) in the latent representation of the model. *P*, precision; *R*, recall.

experiments. The probe shows poor or mediocre results, getting higher scores when inferring the function-words distribution of the documents. This highlights the importance of elements such as function words in the characterisation of literary authors [35]. We can hypothesise that the transformer's apparent lack of encoding information regarding word-lengths distribution might stem from the authors sharing similar backgrounds, and thus similar habits regarding their vocabulary usage.

Regarding probing for the genre of the documents, the results are displayed in Table 5. The probe is clearly able to determine the sub-corpus the document comes from, thus indicating that the transformer indeed encodes the genre of the document into the latent space. This result could help warn the human expert against the risk that the neural model under investigation may be exploiting domain information, which should be avoided in AId studies [9, 26]. The classifier should focus on style-related information, and should not label a document as written by author A simply on the ground that A often writes in the same genre or topic as the document in question.

Summing up, this analysis could indeed reveal to a scholar some of the inner workings of a high-level model, by showing which features it leverages and which it does not, thus reassuring the scholar on the outcome of the classification. In particular, the probing task would be well suited for an active interaction with the scholar who, prompted by their deep knowledge on the literary matter, could propose promising features to analyse, in the form of a "human-in-the-loop" process. However, an important limitation scholars should be aware of is that probes can only be constructed around features that can be automatically extracted, unless one wants to incur the cost of manually labelling the documents according to more complex features.

6.3 Factuals and Counterfactuals for AV

In our experiments, we retrieve one factual and one counterfactual for both SVM and RoBERTa trained for the AV task. Note that we only exemplify this method as applied to AV; however, it can be applied to SAV and AA as well. In particular, we obtain the TfIdf vectors (in the SVM case) or the encodings of the final hidden state (in the RoBERTa case) for a random test instance x and the entire training set; we then compute the Euclidean distances among x and all the training instances and select the training instance closest to x which has the same (different) label as the predicted label of x. Of course, both the number of (counter)factuals returned and the similarity measure to employ are parameters that can be modified.

The selected test example, which is an epistle from author Pier della Vigna, is the following:

> Fridericus uniuersis mundi *principibus* de sinistris rumouribus **Terrae Sanctae** Etsi tam iusta quam uehemens causa doloris et motus fuerit in nobis cum ad presentiam nostram **frater S.a uenerabili patre patriarcha** Antiocheno dilecto *amico nostro* presentium baiulus litterarum accessit ipsum tamen infeste uidere nequiuimus qui mittentem affectione quadam diligimus singulari. Uerum etiam tunc temporis cordis nostri neruum pertingerat rumour infestus et subitae nuntius tempestatis qui Coheminorum pestem ab originalibus sedibus Tartarea clade depulsam uelut molem ingentem per abrupta montium et decliuium fulminis ictibus deuolutam in **Sanctam Ciuitatem** irruisse crudeliter nuntiauit. Quae forte desolationis suae tempore habitatore continui solita defensari cateruatim undique concurrentibus *populis* colebatur dederatque cursui famosi tamen loci longis retro temporibusa **Christicolis** maxime desiderata securitas et sinistris **auspiciis** diebus illis obtenta quarumdam occasione *treugarum* quas *soldanus* Damasci et Nathasar *soldanus* Craci qui prius *hostes* et *aduersarii* fuerant *concordiam* inuicem facientes ipsam cum **Christianis** ea condicione fecerunt quod tota regni Hierosolimitani terra quam **Christiani** possederant trans Iordanem retentissibi uillis et montanis aliquibus restituta **Christiani** *soldanis* eisdem in expugnatione *soldani* Babiloniae deberent assistere toto posse. Qua *confederatione* tamquam in sui perniciem inita *soldanus* accinctus predictam gentem *Barbaricam* Coheminorum per deserta uagantem et uelut feram in saltibus ante uenabulum fugientem ad suae defensionis *auxilium* conuocauit. quisibi reputantes oblatum praesidium potrius quam petitum ad designata loca subito non minus taciti quam celeres peruenerunt ut inuisos *hostes* aduenisse maturis nostrorum uigilantia nouerit quam uenturos. sicque factum est ut **Christianorum** excercitu cum soldanis predictis in guerram soldani Babiloniae apud Gazaram commorante **patriarcha** Hierosolimitanus de partibus Cismarinis ad partes illas athleta nouus accessit. […]

It is the narration of an episode of the Crusades in the Holy Land, with the characteristics of historical chronicles. It is among the numerous letters written by the author during their station as chancellor of Emperor Frederick II.

The same factual example is selected for both the SVM and RoBERTa models; it is again an epistle, this time by author Giovanni Boccaccio:

> Celeberrimi nominis *militi* Iacobo Pizinge serenissimi *principis* Federici Trinacrie regis logothete. Generose *miles* incertus mei Neapoli aliquamdiu fueram uere preterito. hinc enim plurimo desiderio trahebar redeundi in *patriam* quam autumpno nuper elapso indignans liqueram nec minus reuisendi libellos quos immeritos omiseram sic et *amicos* aliosque caros. inde uero urgebar ut consisterem atque detinebar nunca uenerabili uiolentia nunc suasionibus nunc precibus incliti uiri Hugonis de *comitibus* **Sancti** Seuerini cuius credo splendidam famam noueris. Curabat enim uir eximius etiam me inuito totis uiribus ut me interueniente subsidio serenissime *domine* Iohanne Ierusalem et sicilie *regine* apud Parthenopeos placido locaret in otio. qua perplexitate angebar nimium nulla adhuc in parte satis firmato consilio. Et dumsic uariis agitarer curis quo pacto non memini factum tamen est ut ad aures deueniret meas uenerabile nomen **religiosi** hominis Ubertini de **ordine** Minorum **sacre theologie** professoris et conciuis tui cuius auditis meritis eumque ea tempestate Neapoli moram trahere pro quibusdam arduis tui suique *regis* in desiderium uenit tam conspicuum uidere uirum. a pueritia quippe mea etiam ultra tenelle etatis uires talium auidissimus fui. Nec mora. exhibiturus *reuerentiam* debitam ad eum accessi atque adaperto capite primo paxillum miratus hominem quam **deuotissime** et humillime potui salutaui eum. Ipse autem graui quadam maturitate obuius factus me leta facie miti eloquio et morum laudabili comitate suscepit.

In this case, the selected factual is a letter to a notary of the Kingdom of Sicily; the epistle presents a first-person narration, describing some personal anecdotes happened during the author's stay in Naples. The themes apparently could not be more different from those of the test example, but a closer inspection shows that the two texts share many references to religious orders and political relations. We highlight this by displaying in bold some of the former and in italic some of the latter.

Regarding the counterfactual, the same example is selected (again) for both SVM and RoBERTa; it is yet another epistle, this time by author Dante Alighieri (since in this experiment Dante plays the role of the positive class, while the other authors collectively play the role of the negative class):

> Absit a uiro predicante iustitiam ut perpessus iniurias iniuriam inferentibus uelut benemerentibus pecuniam suam soluat. Non est hec uia redeundi ad *patriam* pater mi. sed si alia per uos anteaut deinde per alios inuenitur que fame Dantisque *honori* non deroget illam non lentis passibus acceptabo. quod si per nullam talem Florentia introitur nunquam Florentiam introibo. Quidni. nonne solis astrorumque specula ubique conspiciam. nonne dulcissimas ueritates potero speculari ubique sub celo ni prius inglorium ymo ignominiosum *populo* Florentino *ciuitati* me reddam. Quippe nec panis deficiet.

Unlike the factual, the counterfactual clearly has a very different domain than the test samples, since it is a personal account of the tribulations brought about by his exile. Still, there are again some references to political concepts (again shown in italic).

All in all, it seems that both models are able to spot similarities and differences in the documents, especially the ones linked with the themes and references of the narration, highlighting similar textual patterns. Analogous to the results of the probing for the genre of the documents in Section 6.2, this could serve as a warning for the human experts examining the model, indicating the potential exploitation of domain-specific information, a practice to be avoided in AId research [9, 26]. However, the burden of spotting these similarities and differences is mainly on the human user, and this can be a difficult and time-consuming task. Coupling this XAI technique with other methods that highlight the textual regions, or features, that mostly determine the similarity among the documents, could be helpful in this sense. We give a very elementary exemplification of this by colouring in the texts the 10 n-grams that have the minimum differences among the feature values in the test example and in the factual (in blue), and among the feature values in the test example and in the counterfactual (in red) (the n-grams that are shared among all three texts are coloured in violet).

Other techniques that exist in the related literature generate *ad hoc* synthetic examples as (counter)factuals (see e.g., Lampridis et al. [41]). While it might be possible in principle to generate synthetic textual instances for our case too, it is not clear how these examples could be useful for the human expert, who would realistically be interested in real-world textual documents only, and not in machine-generated texts.

7 Conclusion and Future Works

In this article, we underline the importance of explainability for computational authorship analysis studies, with a specific focus on the case of cultural heritage.

Despite its importance, there are no existing XAI techniques that were specifically devised for authorship studies in the field of cultural heritage, nor for other applications of authorship analysis. We thus experiment with three existing XAI methodologies proposed in other contexts (namely, feature ranking, probing and factual and counterfactual selection), and we test them on the three main AId tasks (AA, AV and SAV), employing a medieval Latin dataset as a case study. We make the code developed for this study available to other researchers who might want to apply these techniques to other authorship analysis problems.

In this study, we demonstrate that each XAI method tested contributes partially to elucidating the rationales behind the predictions of the model, and that they jointly provide some explanations of different aspects of the model. In particular, while feature ranking and probing shed light on the linguistic events leveraged by the model, (counter)factuals put these important linguistic events in context, showing real examples of the writing production under study.

However, we argue that the explanations that can be obtained with current, general-purpose techniques are still largely insufficient, even for a small-scale dataset with a limited number of authors, since they either convey a rather limited perspective of the inner working of the classifier (feature ranking) or heavily rely on the user's input and intuition (probing, factuals and counterfactuals selection). Employing a combination of these methods,

instead of using them in isolation, would mitigate, but not solve, this problem; in particular, a visualisation tool could help display the disputed document with the occurrences of the different features highlighted, with the highlighting coming in different shades depending on the class considered and the significance of the feature. Moreover, the tool could allow the domain expert to select a particular feature of interest, and show one or more examples from the dataset where the feature has a strong and significant presence. However, while solutions of this kind might help the expert navigate across relevant related cases, we argue they would still fall short of providing a convincing and conclusive explanation.

In future work, the exploration for methods that provide meaningful explanations for supporting the research of scholars should continue. We believe that aiming for concise and informative textual explanations (see e.g., Barratt [4] and Le et al. [43]) is an avenue worth exploring, since this is the format most familiar to cultural heritage scholars.

References

[1] Charu C. Aggarwal and ChengXiang Zhai. 2012. A survey of text classification algorithms. In *Mining Text Data*. Charu C. Aggarwal and ChengXiang Zhai (Eds.), Springer, Heidelberg, 163–222.

[2] Albert R. Ascoli. 1997. Access to authority: Dante in the epistle to cangrande. In *Seminario Dantesco Internazionale/International Dante Seminar 1*. Zygmunt G. Baranski (Ed.), Le Lettere, Firenze, 309–52.

[3] Luca Azzetta. 2016. *Nuova edizione commentata delle opere di Dante*, Vol. 5. Salerno Editrice, Roma, IT, Chapter "Epistola XIII", 271–487.

[4] Shane Barratt. 2017. Interpnet: Neural introspection for interpretable deep learning. Retrieved from https://doi.org/10.48550/arXiv.1710.09511

[5] Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics* 48, 1 (2022), 207–219. DOI: https://doi.org/10.1162/coli_a_00422

[6] Dario Benedetto and Mirko Degli Esposti. 2016. Dynamics of style and the case of the "Diario Postumo" by Eugenio Montale: A quantitative approach. In *Creativity and Universality in Language*. Mirko Degli Esposti, Eduardo G. Altmann and Francois Pachet (Eds.), Springer Nature, Cham, CH, 157–176.

[7] Barbara Berti, Andrea Esuli, and Fabrizio Sebastiani. 2023. Unravelling interlanguage facts via explainable machine learning. *Digital Scholarship in the Humanities* 38, 3 (2023), 953–977. DOI: https://doi.org/10.1093/llc/fqad019

[8] José Nilo G. Binongo. 2003. Who wrote the 15th book of Oz? An application of multivariate analysis to authorship attribution. *Chance* 16, 2 (2003), 9–17.

[9] Sebastian Bischoff, Niklas Deckers, Marcel Schliebs, Ben Thies, Matthias Hagen, Efstathios Stamatatos, Benno Stein, and Martin Potthast. 2020. The importance of suppressing domain style in authorship analysis. Retrieved from https://doi.org/10.48550/arXiv.2005.14714

[10] Benedikt T. Boenninghoff, Steffen Hessler, Dorothea Kolossa, and Robert M. Nickel. 2019. Explainable authorship verification in social media via attention-based similarity learning. In *Proceedings of the IEEE International Conference on Big Data (BigData '19)*. IEEE, 36–45. DOI: https://doi.org/10.1109/BigData47090.2019.9005650

[11] Michael C. Bromby. 2011. Juries and their understanding of forensic science: Are jurors equipped? *International Journal of Science in Society* 2, 2 (2011), 247–256.

[12] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8 (2019), 832. DOI: https://doi.org/10.3390/electronics8080832

[13] Alberto Casadei. 2016. Sempre contro l'autenticita dell'Epistola a Cangrande. *Studi danteschi* LXXXI (2016), 215–46.

[14] Carole E. Chaski. 2005. Who's at the keyboard? Authorship attribution in digital evidence investigations. *International Journal of Digital Evidence* 4, 1 (2005), 1–13.

[15] Silvia Corbara, Alejandro Moreo, and Fabrizio Sebastiani. 2023. Same or different? Diff-vectors for authorship analysis. *ACM Transactions on Knowledge Discovery from Data* 18, 1 (2023), Article 12. DOI: https://doi.org/10.1145/3609226

[16] Silvia Corbara, Alejandro Moreo, and Fabrizio Sebastiani. 2023. Syllabic quantity patterns as rhythmic features for Latin authorship attribution. *Journal of the Association for Information Science and Technology* 74, 1 (2023), 128–141. DOI: https://doi.org/10.1002/asi.24660

[17] Silvia Corbara, Alejandro Moreo, Fabrizio Sebastiani, and Mirko Tavoni. 2020. L'epistola a Cangrande al vaglio della computational authorship verification: Risultati preliminari (con una postilla sulla cosiddetta "XIV Epistola di Dante Alighieri"). In *Atti del Seminario "Nuove Inchieste sull'Epistola a Cangrande"*. Alberto Casadei (Ed.), Pisa University Press, Pisa, IT, 153–192.

[18] Silvia Corbara, Alejandro Moreo, Fabrizio Sebastiani, and Mirko Tavoni. 2022. MedLatinEpi and MedLatinLit: Two datasets for the computational authorship analysis of medieval Latin texts. *ACM Journal of Computing and Cultural Heritage* 15, 3 (2022), 571–57:15. DOI: https://doi.org/10.1145/3485822

[19] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing (AACL/IJCNLP '20)*. 447–459.
[20] Christopher W. Forstall, Sarah L. Jacobson, and Walter J. Scheirer. 2011. Evidence of intertextuality: Investigating Paul the Deacon's Angustae Vitae. *Literary and Linguistic Computing* 26, 3 (2011), 285–296.
[21] Jack Grieve. 2007. Quantitative authorship attribution: An evaluation of techniques. *Literary and Linguistic Computing* 22, 3 (2007), 251–270.
[22] Yuling Gu, Bhavana D. Mishra, and Peter Clark. 2021. DREAM: Uncovering mental models behind language models. Retrieved from https://doi.org/10.48550/arXiv.2112.08656
[23] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)* 51, 5 (2018), 1–42.
[24] Ralph G. Hall and Madison U. Sowell. 1989. 'Cursus' in the can grande epistle: A forger shows his hand? *Lectura Dantis* 5 (1989), 89–104.
[25] Oren Halvani. 2021. *Practice-Oriented Authorship Verification*. Ph. D. Dissertation. Technische Universität Darmstadt, Darmstadt, DE.
[26] Oren Halvani, Christian Winter, and Lukas Graner. 2019. Assessing the applicability of authorship verification methods. In *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES '19)*. 1–10. DOI: https://doi.org/10.1145/3339252.3340508
[27] Ronan Hamon, Henrik Junklewitz, and Ignacio Sanchez. 2020. *Robustness and Explainability of Artificial Intelligence*. Technical Report EUR 30040. Publications Office of the European Union, Luxembourg, LU. DOI: https://doi.org/10.2760/57493
[28] David I. Holmes. 1998. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing* 13, 3 (1998), 111–117.
[29] Giles Hooker and Lucas Mentch. 2019. Please stop permuting features: An explanation and alternatives. Retrieved from https://doi.org/10.48550/arXiv.1905.03151
[30] John Hwitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*. 2733–2743.
[31] Fereshteh Jafariakinabad, Sansiri Tarnpradab, and Kien A. Hua. 2020. Syntactic neural model for authorship attribution. In *Proceedings of the 33rd International Conference of the Florida Artificial Intelligence Research Society (FLAIRS '20)*. Virtual Event, 234–239.
[32] Mael Jullien, Marco Valentino, and Andre Freitas. 2022. Do transformers encode a foundational ontology? Probing abstract classes in natural language. Retrieved from https://doi.org/10.48550/arXiv.2201.10262
[33] Patrick Juola. 2006. Authorship attribution. *Foundations and Trends in Information Retrieval* 1, 3 (2006), 233–334. DOI: https://doi.org/10.1561/1500000005
[34] Jakub Kabala. 2020. Computational authorship attribution in medieval Latin corpora: The case of the Monk of Lido (ca. 1101–08) and Gallus Anonymous (ca. 1113–17). *Language Resources and Evaluation* 54, 1 (2020), 25–56. DOI: https://doi.org/10.1007/s10579-018-9424-0
[35] Mike Kestemont. 2014. Function words in authorship attribution. From black magic to theory?. In *Proceedings of the 3rd EACL Workshop on Computational Linguistics for Literature (CLFL '14)*. 59–66.
[36] Mike Kestemont, Enrique Manjavacas, Ilia Markov, Janek Bevendorff, Matti Wiegmann, Efstathios Stamatatos, Benno Stein, and Martin Potthast. 2021. Overview of the cross-domain authorship verification task at PAN 2021. In *Proceedings of the Working Notes of the 2021 Conference and Labs of the Evaluation Forum (CLEF '21)*. 1743–1759.
[37] Mike Kestemont, Sara Moens, and Jeroen Deploige. 2015. Collaborative authorship in the twelfth century: A stylometric study of Hildegard of Bingen and Guibert of Gembloux. *Digital Scholarship in the Humanities* 30, 2 (2015), 199–224. DOI: https://doi.org/10.1093/llc/fqt063
[38] Mike Kestemont, Efstathios Stamatatos, Enrique Manjavacas, Walter Daelemans, Martin Potthast, and Benno Stein. 2019. Overview of the cross-domain authorship attribution task at PAN 2019. In *Proceedings of the Working Notes of the 2019 Conference and Labs of the Evaluation Forum (CLEF '19)*. 1–15.
[39] Mike Kestemont, Justin A. Stover, Moshe Koppel, Folgert Karsdorp, and Walter Daelemans. 2016. Authenticating the writings of Julius Caesar. *Expert Systems with Applications* 63 (2016), 86–96. DOI: https://doi.org/10.1016/j.eswa.2016.06.029
[40] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology* 60, 1 (2009), 9–26. DOI: https://doi.org/10.1002/asi.20961
[41] Orestis Lampridis, Laura State, Riccardo Guidotti, and Salvatore Ruggieri. 2023. Explaining short text classification with diverse synthetic exemplars and counter-exemplars. *Machine Learning* 112, 11 (2023), 4289–4322.
[42] Samuel Larner. 2014. *Forensic Authorship Analysis and the World Wide Web*. Springer, Heidelberg, DE.
[43] Thai Le, Suhang Wang, and Dongwon Lee. 2020. GRACE: Generating concise and informative contrastive sample to explain neural network model's prediction. In *Proceedings of the 26th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD '20)*. Virtual Event, 238–248. DOI: https://doi.org/10.1145/3394486.3403066
[44] Piyawat Lertvittayakumjorn and Francesca Toni. 2019. Human-grounded evaluations of explanation methods for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*. 5195–5205.

[45] Bill Y. Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! NumerSense: Probing numerical commonsense knowledge of pre-trained language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*. Online Event, 6862–6868. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.557

[46] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2020. Explainable AI: A review of machine learning interpretability methods. *Entropy* 23, 1 (2020), 18.

[47] Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*. 5570–5581.

[48] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '19)*. 1073–1094. DOI: https://doi.org/10.18653/v1/n19-1112

[49] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. Retrieved from https://doi.org/10.48550/arXiv.1907.11692

[50] Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR '19)*. DOI: https://openreview.net/forum?id=Bkg6RiCqY7

[51] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence* 2, 1 (2020), 56–67.

[52] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS '17)*. 4765–4774.

[53] Thomas C. Mendenhall. 1887. The characteristic curves of composition. *Science* 9, 214 (1887), 237–249.

[54] Frederick Mosteller and David L. Wallace. 1963. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed Federalist Articles. *Journal of the American Statistical Association* 58, 302 (1963), 275–309.

[55] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[56] Ria Perkins. 2015. Native language identification (NLID) for forensic authorship analysis of weblogs. In *New Threats and Countermeasures in Digital Crime and Cyber Terrorism*. Maurice Dawson and Marwan Omar (Eds.), IGI Global, Hershey, US, 213–234. DOI: https://doi.org/10.4018/978-1-4666-8345-7.ch012

[57] Dheeraj Rajagopal, Vidhisha Balachandran, Eduard H. Hovy, and Yulia Tsvetkov. 2021. SELFEXPLAIN: A self-explaining architecture for neural text classifiers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '21)*. 836–850. DOI: https://doi.org/10.18653/v1/2021.emnlp-main.64

[58] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. 1135–1144.

[59] Laura Rieger and Lars K. Hansen. 2020. IROF: A low-resource evaluation metric for explanation methods. In *Proceedings of the ICLR 2020 Workshop on AI for Affordable Healthcare*. DOI: https://orbit.dtu.dk/en/publications/irof-a-low-resource-evaluation-metric-for-explanation-methods

[60] Anderson Rocha, Walter J. Scheirer, Christopher W. Forstall, Thiago Cavalcante, Antonio Theophilo, Bingyu Shen, Ariadne Carvalho, and Efstathios Stamatatos. 2017. Authorship attribution for social media forensics. *IEEE Transactions on Information Forensics and Security* 12, 1 (2017), 5–33. DOI: https://doi.org/10.1109/TIFS.2016.2603960

[61] Upendra Sapkota, Steven Bethard, Manuel Montes, and Thamar Solorio. 2015. Not all character *n*-grams are created equal: A study in authorship attribution. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '15)*. 93–102.

[62] Yunita Sari, Mark Stevenson, and Andreas Vlachos. 2018. Topic or style? Exploring the most useful features for authorship attribution. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING '18)*. 343–353.

[63] Gennaro Sasso. 2013. Sull'Epistola a Cangrande. *La Cultura* 3 (2013), 359–446.

[64] Jacques Savoy. 2019. Authorship of Pauline epistles revisited. *Journal of the Association for Information Science and Technology* 70, 10 (2019), 1089–1097. DOI: https://doi.org/10.1002/asi.24176

[65] Michael R. Schmid, Farkhund Iqbal, and Benjamin C. Fung. 2015. E-mail authorship attribution using customized associative classification. *Digital Investigation* 14, 1 (2015), S116–S126. DOI: https://doi.org/10.1016/j.diin.2015.05.012

[66] Prasha Shrestha, Sebastian Sierra, Fabio A. González, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL '17)*. 669–674.

[67] Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology* 60, 3 (2009), 538–556. DOI: https://doi.org/10.1002/asi.21001

[68] Efstathios Stamatatos. 2016. Authorship verification: A review of recent advances. *Research in Computing Science* 123 (2016), 9–25.

[69] Efstathios Stamatatos, Mike Kestemont, Krzysztof Kredens, Piotr Pezik, Annina Heini, Janek Bevendorff, Benno Stein, and Martin Potthast. 2022. Overview of the authorship verification task at PAN 2022. In *Proceedings of the Working Notes of the Conference and Labs of the Evaluation Forum (CLEF '22)*. 2301–2313.

[70] Justin A. Stover, Yaron Winter, Moshe Koppel, and Mike Kestemont. 2016. Computational authorship verification method attributes a new work to a major 2nd century African author. *Journal of the American Society for Information Science and Technology* 67, 1 (2016), 239–242. DOI: https://doi.org/10.1002/asi.23460

[71] Antonio Theophilo, Rafael Padilha, Fernanda A. Andaló, and Anderson Rocha. 2022. Explainable artificial intelligence for authorship attribution on social media. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '22)*. 2909–2913.

[72] Enrico Tuccinardi. 2017. An application of a profile-based method for authorship verification: Investigating the authenticity of Pliny the Younger's letter to Trajan concerning the Christians. *Digital Scholarship in the Humanities* 32, 2 (2017), 435–447. DOI: https://doi.org/10.1093/llc/fqw001

[73] Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*. 11–20. DOI: https://doi.org/10.18653/v1/D19-1002

[74] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*. Online Event, 38–45.

[75] Haiyan Wu, Zhiqiang Zhang, and Qingfeng Wu. 2021. Exploring syntactic and semantic features for authorship attribution. *Applied Soft Computing* 111 (2021), 107815. DOI: https://doi.org/10.1016/j.asoc.2021.107815

[76] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. 2006. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technologies* 57, 3 (2006), 378–393.

Received 3 August 2023; revised 10 March 2024; accepted 15 March 2024