

# Syllabic quantity patterns as rhythmic features for Latin authorship attribution

Silvia Corbara<sup>1</sup>  | Alejandro Moreo<sup>2</sup>  | Fabrizio Sebastiani<sup>2</sup> 

<sup>1</sup>Scuola Normale Superiore, Pisa

<sup>2</sup>Istituto di Scienza e Tecnologie dell'Informazione Consiglio Nazionale delle Ricerche, Pisa, Italy

## Correspondence

Silvia Corbara, Scuola Normale Superiore, 56126 Pisa, Italy.

Email: [silvia.corbara@sns.it](mailto:silvia.corbara@sns.it)

## Abstract

It is well known that, within the Latin production of written text, peculiar metric schemes were followed not only in poetic compositions, but also in many prose works. Such metric patterns were based on so-called *syllabic quantity*, that is, on the length of the involved syllables, and there is substantial evidence suggesting that certain authors had a preference for certain metric patterns over others. In this research we investigate the possibility to employ syllabic quantity as a base for deriving rhythmic features for the task of computational authorship attribution of Latin prose texts. We test the impact of these features on the authorship attribution task when combined with other topic-agnostic features. Our experiments, carried out on three different datasets using support vector machines (SVMs) show that rhythmic features based on syllabic quantity are beneficial in discriminating among Latin prose authors.

## 1 | INTRODUCTION

*Authorship Analysis* can be defined “broadly as any attempt to infer the characteristics of the creator of a piece of linguistic data” (Juola, 2006, p. 238), including the author’s biographical information (e.g., age, gender, mother tongue, etc.) and identity. In particular, the set of tasks grouped under the name of *Authorship Identification* (AId) concerns the study of the true identity of the author of a text when it is unknown or debated. The three main tasks of AId are *Authorship Attribution* (AA), *Authorship Verification* (AV), and *Same-Authorship Verification* (SAV). In AA (Koppel et al., 2009; Stamatatos, 2009), given a document  $d$  and a set of candidate authors  $\{A_1, \dots, A_m\}$ , the goal is to identify the real 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  $\{A_1, \dots, A_m\}$ .<sup>1</sup> In AV (Stamatatos, 2016), given a single 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  $\bar{A}$  as the possible classes. In SAV (Koppel & Winter, 2014), given two documents  $d_1$  and  $d_2$ , the goal is to infer whether the two documents  $d_1$  and  $d_2$  are by the same author or not; SAV is thus also a binary classification problem, with SAME and DIFFERENT as the possible classes.

Generally speaking, the goal of AId is to find a way to spot the “hand” of a given writer, so as to clearly separate his/her written production from those of other authors. Hence, the core of this practice, also known as “stylometry,” does not rely on the investigation of the artistic value or the meaning of a written work, but on a *quantifiable* characterization of its style. This characterization is typically achieved through an analysis of the frequencies of linguistic events (also known as “style markers”), where the frequencies of these events are assumed to remain more or less constant throughout the production of a given author (and, conversely, to vary substantially across different authors) (Juola, 2006, p. 241). These linguistic events are often of apparently

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *Journal of the Association for Information Science and Technology* published by Wiley Periodicals LLC on behalf of Association for Information Science and Technology.

minimal significance (such as the use of a punctuation symbol or a conjunction), but are assumed to be out of the conscious control of the writer, and hence hard to modify or imitate. In his essay *Clues*, the noted historian Carlo Ginzburg (1989) describes the emergence of this analytical approach (which he traces back to the late 18th century) in a number of fields of human activity, and calls it the *evidential paradigm*.

As hinted above, (computational) AId tasks are often solved according to a *text classification* approach, in which the texts of unknown authorship are the objects of classification and the classes represent authors (as in AA or AV) or same/different authorship (as in SAV). In turn, text classification is usually solved via *supervised machine learning*, whereby a general-purpose supervised learning algorithm trains a classifier to perform authorship identification by exposing it to a set of training examples (i.e., texts by the authors of interest and whose authorship is certain).

In this work we focus on the AA task for Latin prose documents, and experiment with the idea of using *syllabic quantity* (SQ) (Sturtevant, 1922) in order to derive an additional set of stylistic features for this task. Syllables can be “long” or “short” based on their “quantity” (see section 2.1), and peculiar sequences of long and short syllables were used by Latin authors as metric (i.e., rhythmic) patterns. Our idea to use these sequences as features for performing AId is based on accumulated evidence (again, see section 2.1) suggesting that some Latin authors show a preference, more or less conscious, for specific rhythmic patterns obtained by specific sequences of long and short syllables, even in prose texts. In order to assess the plausibility of this idea we run a number of experiments, using three different datasets, in which we evaluate the impact of SQ-based features on the accuracy of AA.

The rest of this paper is organized as follows. In section 2 we give some theoretical background on Latin prosody and on the concept of SQ, and we discuss how we extract the latter from text. In section 3 we discuss our experiments, including the datasets we employ, the experimental protocol we follow, and the results of the experiments. In section 4 we present some related work, while in section 5 we present final remarks and discussion of avenues for future research.

The code to reproduce all our experiments is available at [https://github.com/silvia-cor/SyllabicQuantity\\_Latin](https://github.com/silvia-cor/SyllabicQuantity_Latin).

## 2 | METHODOLOGICAL SETTING

In section 2.1 we give an introduction to Latin prosody, while in section 2.2 we describe the tool we use for extracting SQ from Latin prose texts.

### 2.1 | A brief introduction to Latin prosody

As other languages, Latin is based on *syllables*, that is, sound units a single word can be divided into, which can be viewed as oscillations of sound in the pronunciation of the word. Every Latin word has as many syllables as it has vowels or diphthongs.<sup>2</sup> Generally speaking, a Latin word is divided into syllables according to the following rules:

- A single consonant and the vowel that follows it belong to the same syllable, for example, “pater” (“father”) divides into two syllables as “pa-ter.”
- Two adjacent consonants belong to two adjacent syllables, for example, “mitto” (“I send”) divides as “mit-to,” and “arma” (“weapons”) divides as “ar-ma.”
- Compounds generate different syllables, for example, “abest” (“he/she/it is missing/away”), being composed of the preposition “ab” (“from”) and the verb “est” (“he/she/it is”), divides as “ab-est.”

A syllable is characterized by its *quantity*, which indicates the amount of time required to pronounce it. Specifically, a syllable can be *long* or *short*, and this is determined first and foremost by the quantity of its vowel, and then by the consonant sounds that follow it. In fact, a single vowel has its own quantity, which depends on the structure of the word or on its etymology: for example, a vowel before another vowel (when the two do not form a diphthong) is short, while a vowel originating from a contraction, such as “nil,” contracted from “nihil” (“nothing”), is long. In the study of SQ, long vowels are traditionally marked with a *macron* ( $\bar{a}$ ), while short vowels are (only sometimes) marked with a *breve* ( $\breve{a}$ ). A syllable is said to be *short* if it contains a short vowel, *long* “*by nature*” if it contains a long vowel (or a diphthong), and *long* “*by position*” if it contains a short vowel followed either by two consonants or by a double consonant (“x” or “z”).

Note that the explanations given here of the syllabification and the quantification rules for Latin are rather generic, and many more detailed rules exist, with exceptions and specific cases, making the study of prosody a nontrivial matter; see the expositions by Allen et al. (1903), Ceccarelli (2018), Harrison (2021), and Sturtevant (1922) for a more complete discussion.

It is well known that classical Latin (and Greek) poetry followed metric patterns based on SQ, that is, on well-chosen sequences of short and long syllables. In particular, syllables were combined in what is called a *foot*, and a series of feet composed the *metre* of a verse. For

example, one of the most renowned meters is the *dactylic hexameter* (employed, among others, by Virgil in his *Aeneid*), which is composed of 6 *dactyl feet* (each consisting of a long and two short syllables), with the possibility of a substitution with a *spondee* (which consists of two long syllables) in most positions; additionally, the sixth foot can be a *trochee* (consisting of a long and a short syllable). An example of a dactylic hexameter is<sup>3</sup>

–UU| – UU| – –| – –| – UU| – XII

Arma virumque calnō, Trōliae quīl p̄rimus abl̄ ōrīs||

Similar metric schemes were followed also in many prose compositions, in order to give a certain cadence to the discourse, and to focus the attention on specific parts of it. The end of sentences and periods was deemed to be especially important in this sense, and it is known as *clausula*. Orators such as Cicero were particularly aware of the effects of such rhythmic endings, as in the example below (consisting of a *molossus*, that is, a foot made of 3 long syllables, followed by a *cretic*, that is, a foot consisting of the sequence “long-short-long”):

– – –| – U – ||

cōsulūm scelus, cupiditās, egestās, au|dācia!||

During the Middle Ages, Latin prosody underwent a gradual but profound change, that also propagated to romance languages: the concept of SQ lost relevance in favor of the *accent*, or *stress*. As a matter of clarification, both phenomena, SQ and accent, were present in both classic and medieval Latin. However, in classic Latin stress did not have a role in rhythmic composition (as we have seen), and was pronounced with a higher *pitch*. Instead, medieval Latin speakers gradually stopped “hearing” the quantities of word syllables, in favor of a higher *intensity* given by stress, even though a stressed syllable typically requires also a longer time to be pronounced. The modern consequences of this process can be seen, for example, in Italian poetry, where a verse is characterized by the number of syllables (but not their quantities) and the positions of the accents. Moreover, Latin accentuation rules are largely dependent on SQ; so, for example, words longer than two syllables are accented on the next-to-last syllable if this syllable is long, for example, “amicus” (“friend”), otherwise they are accented on the third-to-last syllable, for example, “dōmīnus” (“master”).

In the middle of these transformations, medieval writers retained the classical importance of the *clausula*, although it followed the change in paradigm and became based on stress rather than quantity. Stress-based rhythmic patterns are known as *cursus*. We can distinguish the following three main types of *cursus*<sup>4</sup>

*Cursus planus*: – + | + – + ||, illum dedúxit

*Cursus tardus*: – + | + – + + ||, ire tentáverit

*Cursus velox*: – + + | + + – + ||, saécula saeculórum

Many scholars (see, e.g., Janson, 1975; Keeline & Kirby, 2019) have shown that certain Latin authors preferred specific types of rhythmic patterns, and that differences can be detected even between authors who cannot be assumed to consciously care for such rhythmic patterns at all, both in metric based on quantities (Keeline & Kirby, 2019, p. 187) and in metric based on accents (Janson, 1975, p. 20). This is an important point: many Latin authors consistently show a certain (more or less conscious) preference for specific rhythmic constructions, even if they do not follow the prosodic canons of the time.

An author’s use of certain rhythmic patterns might thus play an important role in the identification of that author’s style; in fact, it has already been used (in studies of a noncomputational nature) in cases of debated authorship, for example, regarding some works traditionally attributed to Dante Alighieri (Hall & Sowell, 1989; Toynbee, 1918).

Given these premises, the goal of our work is to investigate whether SQ-based features can be profitably employed for computational AA in Latin prose texts. This seems reasonable also for medieval Latin, since accents are heavily based on SQ, as already explained. Moreover, features derived from SQ are content-agnostic (e.g., a sequence of syllables such as “long-short-long” can stand for hundreds or thousands of different 3-syllable sequences), and thus they could be a valuable tool for authorship analysis problems, since they are free of unwanted influence from the domain.

## 2.2 | Extracting syllabic quantity for Latin prose texts

An important part of the computational system that needs to be assembled in order to carry out our experiments on SQ is a module that, given a piece of Latin written text, extracts SQ from it, in order to generate SQ-based features that can be used for classification. Since developing such a module would be a major endeavor, due to the complexities of Latin prosody already mentioned in section 2.1, we decided to use an off-the-shelf tool, chosen among those that are publicly available.

One such tool we considered is the one that resulted from the *Cursus in Clausula* project (Spinazzè, 2014): it is a web application that extracts all the forms of *cursus* from an uploaded text and allows performing some statistical analyses on it. However, this tool analyses only the final portions of periods and sentences (see the definitions of *clausula* and *cursus* in section 2.1). We prefer instead to analyze the entire document, and not only the final portions of periods and sentences, since this would

**TABLE 1** Example results of the two modules from the CLTK library on a Latin prose text

Original text	Quo usque tandem abutere Catilina patientia nostra. Quam diu etiam furor iste tuus nos eludet
Macronizer	quō usque tandem abūtēre catilīna patientia nostra. Quam diū etiam furor iste tuus nōs ēlūdet
Scanner	[−U − U − − UUU − UUU − UU − X, −UUU − UU − UU− − − X]

highlight potential rhythmic preferences of an author in creating the general structure of the discourse; additionally, we should remember that there are rhythmic rules also for the beginning of the sentence (Janson, 1975).

We eventually implemented our SQ extraction module by using the Classical Language ToolKit library (CLTK; Johnson et al., 2021): among many tools for the study of ancient languages, it offers specific ones for the study of Latin prosody, in particular the two modules `MACRONIZER` (which places a macron over long vowels) and `SCANNER` (which produces a sequence of the symbols denoting the quantity of a syllable, that is, short, long, and end of sentence). The output of each module is illustrated in the respective entry in Table 1.

### 3 | EXPERIMENTAL SETTING

As already stated, we focus specifically on the authorship attribution (AA) problem: as mentioned in section 1, given document  $d$  we assign it to exactly one class among a set  $\{A_1, \dots, A_m\}$  of candidate classes, that is, possible authors. We evaluate the contribution of SQ-derived features using three different datasets. The quality of SQ-derived features is inferred from the difference between the performance of a method without SQ-based features and the performance of the same method equipped with them.

The programming language we employ in this project is Python; in particular, we use several modules from the scikit-learn (Pedregosa et al., 2011) package.

#### 3.1 | The datasets

Currently, no Latin dataset is considered a standard in AA studies, and even the available ones are few. In this section we present the three Latin datasets we perform our experiments on: one of them was assembled by us (section 3.1.1) while the other two were originally presented in other AId works (sections 3.1.2 and 3.1.3).

#### 3.1.1 | LatinitasAntiqua

For this work we have created an ad hoc Latin dataset to best suit our needs. For this we exploited the *Corpus Corporum* repository,<sup>5</sup> and in particular its subsection called *Latinitas Antiqua*, which contains various Latin works from the Perseus Digital library<sup>6</sup>; these works are meticulously tagged in XML. From this section we further selected a group of texts that (a) are not poetry works, since our study only deals with prose writings, and (b) are not theatrical pieces, since these latter have a very peculiar format based on dialogue and scenes. The resulting dataset is presented in detail at <http://nmis.isti.cnr.it/sebastiani/LatinitasAntiqua.pdf>. It consists of 90 prose texts by 25 Latin authors, spanning the Classical, Imperial and Early Medieval periods, and a variety of genres (mostly epistolary, historical, and rhetoric).

By exploiting the XML tagging, from each text we delete foreign words (e.g., Greek) and the direct quotations from other authors, in order to retain only the “pure” production of the writer. We then remove every remaining XML tag.

#### 3.1.2 | KabalaCorpusA

This dataset was developed by Kabala (2020). In particular, of the four datasets that he assembled, we exploit CorpusA, the biggest one, which consists of 39 texts by 22 authors from the 11th and 12th centuries, with the texts belonging to various genres (history, theology, political theory). Long quotations and passages of poetry have been already removed from the texts by the author.

#### 3.1.3 | MedLatin

This dataset was developed by Corbara et al. (2021). The authors originally divided it into two subdatasets, MedLatinEpi and MedLatinLit, both containing Latin prose works 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 various nature, especially comments on treatises and literary works.<sup>7</sup> For this project we combine the two subdatasets together. We delete the quotations from other authors and the parts in languages other than Latin, both marked in the texts.

### 3.2 | Preprocessing the data

We automatically preprocess all the documents in the three datasets in order to clean them, as much as possible, from

TABLE 2 Information regarding the datasets we use

LatinitasAntiqua	Entire texts	# entire texts	90
		Mean length	40,170
	Fragments	# fragments	23,219
		Mean length	156
KabalaCorpusA	Entire texts	# entire texts	39
		Mean length	34,389
	Fragments	# fragments	7,882
		Mean length	170
MedLatin	Entire texts	# entire texts	294
		Mean length	3,985
	Fragments	# fragments	6,028
		Mean length	194

Note: For each dataset we report the number of items (# entire texts, or # fragments) and the mean number of words for each item (mean length), both for the entire texts and for the resulting fragments.

spurious information and noise. In particular, we delete headings, editors' notes, and other meta-information, if present. We delete symbols (such as asterisks or parentheses) and Arabic numbers, since they are likely bibliographical information inserted by the editor. We normalize punctuation marks: we delete commas, and we replace all question marks, exclamation marks, semicolons, colons and suspension points with full periods. We do this because punctuation was absent or hardly coherent in ancient manuscripts, hence the punctuation we see in current editions follows modern habits, and is mostly due to the editor, not to the author (Tognetti, 1982, p. 57). However, we retain full periods in order to be able to divide the text into sentences. We lowercase the text, and then we normalize it, that is, we exchange (a) all occurrences of character  $v$  with character  $u$ , (b) all occurrences of character  $j$  with character  $i$ , and (c) every stressed vowel with the corresponding nonstressed vowel.<sup>8</sup>

As a final step, we divide each text into sentences, where a sentence is made of at least 5 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). Each nonoverlapping sequence of 10 consecutive sentences is what we call a (text) *fragment*. These fragments are the samples that we give as input to our learning algorithm and classifiers; the final amount of fragments for our three datasets is displayed in Table 2, along with some additional information regarding the composition of each dataset.

### 3.3 | Topic-agnostic features: Base features and distorted views

In surveying the results of the PAN 2011 shared task, Argamon and Juola (2011) also describe the features that

in 2011 were, and have largely remained, standard for the representation of texts in AId tasks. In the survey by Stamatatos (2009), these features are divided into five major types:

- Lexical: features based on words and their patterns of occurrence (e.g., word and sentence lengths, vocabulary richness, word  $n$ -grams, ...).
- Character-based: features based on characters and their patterns of occurrence in the text (e.g., character  $n$ -grams, compression measures, ...).
- Syntactic: features based on syntax (e.g., part-of-speech tags, ...).
- Semantic: features based on semantics (e.g., synonyms, semantic dependencies, ...).
- Application-specific: features specifically engineered for the particular application under study (e.g., HTML tags, use of indentation, ...).

However, it has been frequently noted that certain AId methods run a high risk of involuntarily leveraging the domain (i.e., topic) the text is about, rather than its style; in the terminology of statistics, domain-dependent features here act as *confounding variables*. This means that, as pointed out, for example, in Bischoff et al. (2020) and Halvani et al. (2019), if domain-dependent features are used, an authorship classifier (even a seemingly good one) might not really perform *authorship* identification, as desired, but might unintentionally perform *topic* identification, unwittingly leveraging not the linguistic peculiarities of an author but those typical of a certain topic. Of course, it is true that some authors confine their written production to very restricted domains, but it would clearly be a poor decision to classify a document as written by  $A$  only because  $A$  often or always writes about the same topic the document is about. Word  $n$ -grams and character  $n$ -grams may particularly suffer from this problem (Stamatatos, 2009); in fact, the good performance they usually deliver in AId tasks may be due to the fact that the datasets used for these tasks are often not topic-neutral. It would hence be good practice to avoid, as much as possible, using features that are not topic-independent when implementing authorship analysis algorithms.

With this goal in mind, various techniques can be employed. One possibility consists of using only features that are obviously topic-agnostic, such as function words or syntactic features (Halvani et al., 2020; Jafariaknabad et al., 2020). A second possibility consists of actively *masking* topical content via a so-called “text distortion” approach (Stamatatos, 2018; van der Goot et al., 2018).<sup>9</sup> In this work we follow both routes; we discuss them in sections 3.3.1 and 3.3.2, respectively. We then assess the

effect of SQ in AA tasks by adding SQ-based features to the topic-agnostic representation of the text, using the difference in performance between the two representations as a measure of this effect.

### 3.3.1 | Base features

We employ a set of features that are widely used in the authorship analysis literature and generally considered topic-independent. In this paper they will act as a common base for each classifier, with other types of features added to them. We call this set `BASEFEATURES` (from now on: BFs); it is composed of the following types of features:

- Function words: the relative frequency of each function word. For a discussion about this type of features, see, for example, the study by Kestemont (2014). We use the list of 80 Latin function words at <http://nmis.isti.cnr.it/sebastiani/StopWords.pdf>.
- Word lengths: the relative frequency of words up to a certain length, from a minimum of 1 up to a maximum of 25 characters. These are standard features employed in statistical authorship analysis since Mendenhall's "characteristic curves of composition" (Mendenhall, 1887).
- Sentence lengths: the relative frequency of sentences up to a certain length, from a minimum of 1 up to a maximum of 100 words. These features have been employed in statistical authorship analysis at least since (Yule, 1939).
- POS-tags: the relative frequency of each part-of-speech (POS) tag. POS tags are examples of syntactic features, and are often employed in authorship analysis studies, also thanks to their topic-agnostic nature (see section 3.3). We extract POS tags using the TnT tagger module of the Classical Language Toolkit; the extraction results in 12 POS tags being assigned to our data.<sup>10</sup>

For each such type of features we compute a matrix  $f \times t$ , where  $f$  is the number of fragments and  $t$  is the number of features of the specific type, and we further scale each vector to unit norm. Given the four resulting matrices, we concatenate them into a single final matrix  $f \times 217$ , where  $(80 + 25 + 100 + 12) = 217$  is the total number of BFs.

### 3.3.2 | Distorted views

We experiment with the four text "distortion" (i.e., masking) methods presented by Stamatatos (2018),

which aim to preserve the document's stylistic characteristics while at the same time hiding its topical content; each such method generates what Stamatatos (2018) calls a *distorted view* (from now on: DV). Given a list  $F$  of function words,<sup>11</sup> the four DVs are:

- Distorted View Single asterisk (DVSA): every word not included in  $F$  is masked by replacing it with an asterisk (\*).
- Distorted View Multiple asterisks (DVMA): every word not included in  $F$  is masked by replacing each of its characters with an asterisk (\*).
- Distorted View Exterior characters (DVEX): every word not included in  $F$  is masked by replacing with an asterisk (\*) each of its characters except the first and the last one. The rationale of DVEX is that the ending of a word and the beginning of the following word might create phonetic effects that certain authors may want to avoid (e.g., using a word that begins with the same character as the ending of the preceding word), or, conversely, to actively employ in their writing.
- Distorted View Last 2 (DVL2): every word not included in  $F$  is masked by replacing with an asterisk (\*) each of its characters except the last two. Underlying DVL2 is the attempt to capture morpho-syntactic information (e.g., number, tense), that is often encoded in language via word suffixes.

The logic behind these masking methods is to remove any type of topic-dependent information from the representation of the text, while at the same time retaining topic-independent information. Some of the information that is retained with these methods is independent from word order, such as function words, word lengths, sentence lengths, beginning and ending characters of words, and their frequencies; some of this information is already captured by the base features of section 3.3.1. However, some of the information that is retained is instead positional, that is, dependent on word order; examples are:

- For DVMA, DVEX, DVL2: the lengths of words that follow (or precede) specific function words, the lengths of words that are used as the first (or last) word of the sentence, the lengths of words that follow (or precede) short words (or long words), and their frequencies.
- For DVEX: the frequencies with which a word begins (or ends) with certain characters, the frequencies with which a word that ends with a certain character is followed by a word that begins with another given character, etc.
- For DVL2: the frequencies with which a word that ends with a given sequence of two characters is followed by a short word, etc.

In other words, these DVs allow capturing phenomena that transcend the lexical level, and that thus pertain to the structure of the sentence.

By using the methods described in section 2.2 and in this section, we thus obtain five different “encodings” of each document, that is, the one representing SQ, and the four DVs described by Stamatatos (2018). From these five encodings we can extract various kinds of features; we discuss the feature extraction methods for SQ and DVs in section 3.4.1. Note that we also use the combination of the features extracted from all four DVs; we call such a combination ALLDV.

### 3.4 | Experimental protocol

We assess the performance of the different classifiers on a given dataset by randomly splitting the dataset into a training set (containing 90% of the data) and a test set (10%). After performing this split, we further remove from the training set 10% of its data, in order to use it as validation data.<sup>12</sup> This tri-partition of the dataset into training set/validation set/test set is stratified, meaning that the class distribution in the original dataset is preserved in all three resulting subsets. For a given dataset, we use the same tri-partition for all the classifiers being tested. As the evaluation measure we use the well-known *macro-averaged*  $F_1$  (hereafter:  $F_1^M$ ) and *micro-averaged*  $F_1$  (hereafter:  $F_1^μ$ ) functions.

As anticipated, we aim to compute the difference in performance between a method employing SQ-based features and the same method without SQ-based features, using this difference as an indicator of the contribution of SQ to AA for Latin prose texts. To this aim, we also compute the statistical significance of the above difference, via McNemar’s paired nonparametric statistical hypothesis test (McNemar, 1947). Since the test applies to binary results (instead of categorical results), we convert the predictions of the two methods of interest into binary values, where 1 stands for a correct prediction and 0 stands for a wrong prediction. We take 0.05 as the confidence value for statistical significance.

#### 3.4.1 | Support vector machines

As the learning algorithm, we use support vector machines (SVMs), a standard learning algorithm widely used in AId (see also section 4.1).

The SVM implementation we employ in this study is LINEARSVC, from the scikit-learn package<sup>13</sup> (Pedregosa et al., 2011). This implementation employs by default a linear kernel and a one-vs-rest multi-class strategy, which

TABLE 3 Number of features extracted in the different feature sets from the combination of training and validation sets

	LatinitasAntiqua	KabalaCorpusA	MedLatin
BFs	217	217	217
SQ	3,242	2,929	2,592
DVMA	474	456	469
DVSA	470	453	466
DVEX	1,245	1,028	1,014
DVL2	1,416	1,139	1,109

Note: For the SQ-grams we report the total number of features extracted, before feature selection.

the developers have found to be similar in performance to the method by Crammer and Singer (2001) (a standard algorithm for turning binary SVMs into multiclass SVMs), but less demanding in terms of computational cost.<sup>14</sup>

We experiment with various SVM-based classifiers, each characterized by a specific feature set. In order to feed the five different encodings of the text (the SQ-based encoding and the four DV-based encodings) to a SVM, we extract character  $n$ -grams from the encoded texts. Given the matrix of BFs (see section 3.3.1), any additional feature set is simply concatenated to it, so that the  $f \times 217$  matrix of BFs becomes an  $f \times k$  matrix, where  $(k - 217)$  is the number of additional features, that is, character  $n$ -grams extracted from the various encodings of the text. We show the number of features extracted as BFs and from each encoding in Table 3.

In particular, for the SQ encoding we use character  $n$ -grams (where a “character” ranges on the three SQ symbols “U,” “–,” “X”) with all values of  $n$  in the range  $[\alpha, \beta]$ . We set  $\alpha = 3$  since many metric feet in Latin poetry are based on 3 syllables, and we set  $\beta = 7$  because the most important cursus rhythms are based on schemes between 5 and 7 syllables long (see section 2.1). On the other hand, for each of the DV encodings we follow Stamatatos (2018) and use character 3-grams that appear at least 5 times in the training set.

For all the features derived from the five encodings, we perform feature weighting via TFIDF.<sup>15</sup> Since the SQ encoding gives rise to a large number of features (that we will call *SQ-grams*),<sup>16</sup> we perform filter-style feature selection (i.e., we retain the  $p$  top-scoring features) using  $\chi^2$  (probably the most frequently used feature selection function in machine learning) as the feature scoring function.<sup>17</sup> We do not perform feature selection on the BFs set and, following Stamatatos (2018), neither on the  $n$ -grams extracted from the DVs encodings.

We perform the optimization of two parameters: the SVM parameter  $C$ , which sets the trade-off between the

training error and the margin, and the feature selection factor  $r$ , which is the fraction of SQ-grams that are retained as a result of the feature selection phase. In particular, our approach is as follows:

1. We create a list of possible configurations for the classifier, where a configuration is made of a possible value for parameter  $C$  (we explore the range of values [0.001, 0.01, 0.1, 1, 10, 100, 1000]) and, if the method employs SQ-grams, a possible value for parameter  $r$  (we explore the range of values [0.1, 0.2, 0.3, 0.4, 0.5, 1.0]). Thus, a possible configuration is ( $C = 10$ ) if the method does not employ SQ-grams, or ( $C = 10$ ,  $r = 0.5$ ) if the method does.
2. For all configurations, we train a classifier with a certain configuration on the training set, and assess the performance of the classifier on the validation set.
3. Using the configuration that has scored the highest value of  $F_1^M$  on the validation set, we train the final classifier from the union of training set and validation set.
4. We assess the final classifier on the test set, evaluating  $F_1^M$  and  $F_1^H$ .

### 3.5 | Results

In our experiments we compare various models with SQ-based features against the same models without SQ-based features, in order to assess the performance gain (if any) obtained by the addition of these features. The results of our experiments are displayed in Table 4.

As can be seen in Table 4, in most cases the accuracy of the classifier is improved by the addition of the SQ-grams. This effect is very substantial in *LatinitasAntiqua*, where the presence of the SQ-grams always improves the accuracy, irrespectively of feature set and evaluation measure, and in half the cases does so in a statistically significant sense. The improvement remains considerable in *KabalaCorpusA*, although the SQ-grams cause a (statistically insignificant) decrease in performance in one case. Finally, it is difficult to give a proper assessment for the *MedLatin* dataset, since the SQ-grams result in a statistically significant difference in only two cases, one with a negative outcome and one with a positive outcome. We conjecture that this might be due to rhythmic patterns suffering from the limited size of the documents and from the authors being located in a small timeframe, both facts being true for *KabalaCorpusA* and even more for *MedLatin*. It is also worth noting that the increased accuracy obtained through the use of SQ-grams tends to be lower in methods employing the DVEX and DVL2 masking methods; this is natural, since these methods

reach very high  $F_1$  values anyway, which means that for them the margin of improvement is narrower.

In general, it is worth noting that

- For 4 out of 6 combinations (evaluation measure, dataset), the best-performing feature set involves SQ-based features.
- For 8 combinations (feature set, dataset), SQ-based features bring about a statistically significant improvement in performance, while a statistically significant deterioration in performance due to the introduction of these features is observed only in one case.

Overall, this confirms that the idea of deriving rhythmic features from syllabic quantity and to apply them to authorship attribution for Latin prose text is a fruitful one.<sup>18</sup>

## 4 | RELATED WORK

### 4.1 | Machine learning for AId tasks

AId tasks are usually tackled by employing methods based on machine learning (thereby viewing these tasks as instances of text classification) or on distance metrics. The annual PAN shared tasks (see, e.g., Bevendorff et al., 2020, 2021; Kestemont et al., 2019) offer a very good overview of the most recent trends in AId, often posing challenging problems in cross-domain and/or open-set settings.

In particular, the baselines presented in the 2019 edition of PAN (Kestemont et al., 2019) mirror the most frequently employed systems, that is, simple classifier-learning algorithms such as support vector machines (SVMs) or logistic regression, distance functions based on compression algorithms, and variations on the well-known Impostors method (Koppel & Winter, 2014). In particular, SVMs are a standard learning algorithm for many text classification tasks, due to their robustness to high dimensionalities and to their general applicability. In various settings they are often shown to outperform other learning algorithms such as decision trees and even neural networks (NNs) (Zheng et al., 2006).

Despite the good results obtained in other natural language processing tasks (Young et al., 2018), for a long time NNs have rarely been employed in AId tasks, arguably due to the huge quantity of training data they usually require. Even though one of the first appearances of NNs at PAN dates back to Bagnall (2015), winner of the 2015 PAN edition (Stamatatos et al., 2015), until recently it was generally accepted that “simple approaches based on character/word  $n$ -grams and well-known



TABLE 4 Results of our experiments

	LatinitasAntiqua			KabalaCorpusA			MedLatin						
	$F_1^M$	$\Delta\%$	$F_1^H$	$M$	$\Delta\%$	$F_1^H$	$F_1^M$	$\Delta\%$	$F_1^H$	$M$	$\Delta\%$	$F_1^H$	$M$
BFs	0.620		0.721			0.692	0.659		0.764			0.814	
BFs + SQ	0.718	+15.79	0.801	✓	+11.05	0.717	0.690	+4.74	0.721	✓	+3.66	0.814	+0.00
BFs + DVMA	0.718		0.803			0.716	0.678		0.754			0.819	
BFs + DVMA + SQ	0.751	+4.70	0.823	✓	+2.47	0.752	0.715	+5.44	0.691	✓	+4.96	0.791	-3.44
BFs + DVSA	0.693		0.790			0.712	0.675		0.760			0.824	
BFs + DVSA + SQ	0.750	+8.36	0.826	✓	+4.52	0.743	0.709	+5.05	0.780	✓	+4.27	0.842	+2.21
BFs + DVEX	0.842		0.891			0.828	0.821		<b>0.904</b>			0.915	
BFs + DVEX + SQ	0.856	+1.67	0.898	x	+0.68	0.834	0.825	+0.54	0.795	x	+0.77	0.894	-2.36
BFs + DVL2	0.849		0.894			0.868	0.860		0.817			0.902	
BFs + DVL2 + SQ	0.857	+0.86	0.901	x	+0.77	0.866	0.852	-0.99	0.835	x	-0.29	0.915	+1.47
BFs + ALLDV	0.879		0.922			0.864	0.855		0.846			<b>0.947</b>	
BFs + ALLDV + SQ	<b>0.888</b>	+0.99	<b>0.927</b>	x	+0.56	<b>0.885</b>	<b>0.887</b>	+3.74	0.834	✓	+2.35	0.935	-1.23

Note: Pairs of experiments, one without SQ-based features and the other with SQ-based features, are reported on two consecutive rows. For each experiment we report (a) the values  $F_1^M$  and  $F_1^H$ , derived on a single train-validation-test split, (b) the percentage of improvement (indicated as  $\Delta\%$ ) resulting from the addition of SQ-based features, and (c) the results of the McNemar statistical significance test (M) against the baseline (✓: statistical significance confirmed; x: statistical significance rejected). Boldface indicates the best result obtained for the given dataset and evaluation measure.

classification algorithms are much more effective in this task than more sophisticated methods based on deep learning” (Kestemont et al., 2018, p. 9). While NN methods are nowadays employed more frequently also in AId tasks (Bevendorff et al., 2020, 2021), for ancient languages such as Latin their use is still problematic, due to the fact that, in these contexts, training data are much less abundant than for modern languages.

## 4.2 | AId for the Latin language

Beside applications in cybersecurity and forensics (Afroz et al., 2014; Leonard et al., 2017), AId has also been used to help philologists and literature scholars to untangle (or at least to provide additional evidence for) some longstanding authorial debates. Indeed, researchers have applied AId methods to historical documents whose authorship has been lost, or hidden, during the passing of centuries. In the present study we limit ourselves to the application of these methods to the Latin language (both classical and medieval), but many more cases have been studied in other languages, starting with the seminal study of the Federalist Papers (Mosteller & Wallace, 1963). However, ancient languages such as Latin pose additional problems when compared to other (widely spoken) modern languages, like the alterations due to the textual tradition and the heavy limitations in data availability. Still, notable results have been obtained in various studies, for example in the research on the *Corpus Caesarianum* (Kestemont et al., 2016) and on the writing of Hildegard of Bingen (Kestemont et al., 2015), in the identification of Apuleius as the most probable author of a newly found manuscript (Stover et al., 2016), in the inquiry of Tuccinardi (2017) regarding the authenticity of one of Pliny the Younger’s letters, in the investigation of Kabala (2020) on the identities of the Monk of Lido and Gallus Anonymous, and in the study regarding the so-called “13th Epistle” of Dante Alighieri (Corbara et al., 2019, 2021). In parallel to studies on Latin, other studies tackle other ancient European languages; see, e.g., Savoy (2019).

AId methodologies can be applied even to literary pieces whose authorship is well-known and certain, in order to find possible stylistic influences from other authors; for example, the goal of Forstall et al. (2011) is to verify a supposed influence by Catullus on the poetry of Paul the Deacon.

## 4.3 | Prosodic features in AId

In this landscape the idea of employing prosodic features is not a new one. Of course, their most natural use is in

studies focused on poetry, such as in the study of Neidorf et al. (2019) on the Old English verse tradition, or in the already cited investigation by Forstall et al. (2011) on the supposed influence of Catullus on Paul the Deacon’s writings. Nevertheless, rhythmic or prosodic features have also been employed in authorship analysis of prose text. However, these works usually consist of the study of word repetitions, like the anaphora (the repetition of a word, or a sequence of words, from a previous sentence at the beginning of a new sentence) (Lagutina et al., 2021), or are based on mapping the texts into the corresponding sequences of sounds before extracting the  $n$ -grams, such as in the research by Forstall and Scheirer (2010), where the authors employ the CMU Pronouncing Dictionary for the conversion.<sup>19</sup> Finally, syllables have been used as base units in other AId works (Sidorov, 2018) and more generally in other NLP tasks, such as poem generation (Zugarini et al., 2019). These works, while close in nature to our project, explore a linguistic dimension different from the one we aim to capture with the study of syllabic quantity.

Some studies closer to ours employ the distribution of accent in order to derive rhythmic features for AA in English. The work by Dumalus and Fernandez (2011) is a pioneering one in this sense: using the CMU Pronouncing Dictionary, they extract the pronunciation of each word and transform it into a “stress string,” where the symbols {0, 1, 2} represent the absence of stress, a primary stress, and a secondary stress in the syllable, respectively. Ivanov et al. (2018) improve on this work: since many English words are homographs (i.e., they have the same spelling but different pronunciation and meaning), they select the correct pronunciation, and hence the correct stress string, by studying the parts of speech of the words in the text. Similarly, Plecháč (2021) employs the frequencies of “rhythmic types” (where a rhythmic type is a bit string representing the distribution of stressed and unstressed syllables in a line) as features in tackling the attribution problem for *Henry VIII*. Accentuation, as explained in section 2.1, is related to the concept of syllabic quantity (at least in the Latin language); however, to the best of our knowledge, syllabic quantity has never been employed for any AId tasks concerning prose texts.

## 5 | CONCLUSION AND FUTURE WORK

In this project we exploit the notion of “syllabic quantity” in order to derive features for the computational authorship attribution of Latin prose texts; these features correspond to sequences of syllables marked according to SQ,

and are meant to capture rhythmic aspects of textual discourse. In comparative experiments over three different Latin datasets and different modes for representing topic-independent textual content, we show that using SQ-derived information has a generally beneficial effect. For 4 out of 6 combinations of 2 evaluation measures  $\times$  3 datasets, the best performance is obtained with a setup that involves SQ-derived information, and for 8 combinations (feature set, dataset), SQ-based features bring about a statistically significant improvement in performance, while a statistically significant deterioration in performance due to the introduction of these features is observed in only one case.


Future work along these lines might take at least three different directions. First, it would be interesting to see if the results we have obtained on prose works are confirmed also on theatrical pieces, a type of text that is not present in our datasets (and that we have deliberately excluded from *LatinitasAntiqua* when creating it). In some sense, this literary genre seems somehow intermediate between poetry and prose, since it is rich in text of a declamatory nature; it is thus conceivable that the analysis of SQ might be beneficial here too, maybe even more than in the case of prose texts. Second, in this project we focus on the AA task, but SQ-based features could easily be employed in other authorship identification tasks, such as authorship verification and same-authorship verification; we conjecture that SQ-derived features might be beneficial in these other settings too. Third, and perhaps more important, the fact that we have found SQ to have generally positive effects on authorship attribution encourages to pursue the investigation of the importance of rhythm on authorship identification tasks on languages other than Latin, starting from ones linguistically close to Latin, such as Italian or Spanish. In particular, it would be fascinating to study whether modern-day prose writers unconsciously opt for specific rhythmic patterns, to the point of being uniquely recognizable thanks to them.

## ACKNOWLEDGMENTS

The first exploratory steps for this research have been conducted during the preparation of the BSc thesis of Canapa (2021), co-supervised (aside from the first author and third author) by Vittore Casarosa; we thank both Giulio and Vittore. We thank Mirko Tavoni for feedback on an earlier draft of this paper. Open Access Funding provided by Scuola Normale Superiore within the CRUI-CARE Agreement.

## ORCID

Silvia Corbara  <https://orcid.org/0000-0002-5284-1771>

Alejandro Moreo  <https://orcid.org/0000-0002-0377-1025>

Fabrizio Sebastiani  <https://orcid.org/0000-0003-4221-6427>

## ENDNOTES

- <sup>1</sup> 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.
- <sup>2</sup> Diphthongs are combinations of two vowels that count as a single, long vowel. In Latin, only the combinations “ae,” “au,” “ei,” “eu,” “oe,” “ui” are diphthongs.
- <sup>3</sup> Regarding the following notation, “–” stands for a long syllable, “⌣” for a short syllable, and “X” for an *anceps*, which can be either a short or a long syllable; the vertical bar indicates where one foot ends and the other begins, and the double vertical bar indicates where the dactylic hexameter ends. Concerning this example, we observe that the particle “-que” is always a short syllable, and that an “i” between vowels has a consonant function.
- <sup>4</sup> Regarding the following notation, “–” stands for a stressed syllable and “+” stands for an unstressed syllable. See, for example, Oberhelman and Hall (1984) and Janson (1975) for an in-depth analysis of this stylistic technique.
- <sup>5</sup> <http://www.mlat.uzh.ch/MLS/>
- <sup>6</sup> <http://www.perseus.tufts.edu/hopper/>
- <sup>7</sup> From *MedLatinEpi* we exclude the texts from the collection of Petrus de Boateriis, since the collection consists of a miscellanea of different authors, often represented by just one epistle each; as such, this collection is hardly useful for our goals.
- <sup>8</sup> We do this in order to standardize the different approaches that different editors might follow. For example, in medieval written Latin, instead of the two modern graphemes “u-U” and “v-V”, there was only one grapheme, represented as a lowercase “u” and a capital “V”; some contemporary editors follow this canon while others to modernize the written text with the two separate graphemes.
- <sup>9</sup> “Distortion” is the term originally used by the author of this approach, but we will instead often use the term “masking”, since the former term implies the introduction of artificial noise in one’s content, which does not occur in this approach, while the latter term implies the act of hiding part of one’s content, which is indeed the case here.
- <sup>10</sup> <https://legacy.cltk.org/en/latest/latin.html#pos-tagging>
- <sup>11</sup> We employ the same list of function words of section 3.3.1.
- <sup>12</sup> Note that since our texts are split into fragments (see section 3.2), it might well be that some fragments belonging to a text end up in the training data while other fragments from the same text end up in the test data. It might be argued that, as a consequence, the attribution task is unduly facilitated, since patterns encountered in the test data may have already been encountered in the training data. However, even assuming this to be the case, the task would be equally facilitated for a system that employs syllabic quantity and for a system that does not employ it, which means that the comparison between them is hardly going to be invalidated. We also want to stress that enforcing a stricter

separation between training data and test data would be hardly possible for us, since for several authors that we consider (e.g., 15 authors out of 25 in the *LatinitasAntiqua* dataset) we only have 1 text (typically, an entire book that gives rise to several thousand fragments). Such problems arise routinely when dealing with ancient texts, since in these cases the number of available labeled texts may be extremely limited, and is anyway upper-bounded by the known production of the authors considered. The above-mentioned lack of a stricter separation between training and text data can thus be found in many authorship analysis works that deal with ancient/cultural heritage texts, for example, Gamon (2004), Kestemont et al. (2015, 2016), Tuccinardi (2017), Boyd (2018), Koppel et al. (2007), Luyckx and Daelemans (2008).

<sup>13</sup> <https://scikit-learn.org/>

<sup>14</sup> See the documentation at <https://scikit-learn.org/stable/modules/svm.html> for more information.

<sup>15</sup> For the `scikit-learn` module we employ in order to compute TFIDF, see [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_extraction.text.TfidfTransformer.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html).

<sup>16</sup> For instance, for the *LatinitasAntiqua* dataset the SQ encoding gives rise to a number of features one order of magnitude larger than either the DVSA or the DVMA encodings.

<sup>17</sup> We use the  $\chi^2$  implementation provided by `scikit-learn`, see [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_selection.chi2.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html)

<sup>18</sup> The reader unfamiliar with authorship attribution and machine learning might wonder what are the reasons behind the fact that 100% accuracy (i.e.,  $F_1^M = F_1^U = 1$ ) is not reached. In general, no authorship attribution algorithm ever reaches 100% accuracy (nor it is expected to) on nontrivial scenarios. Authorship attribution is a complex task, in which even the best systems do sometimes fail; reasons vary, but certainly include (a) the fact that a text might be too short to reveal its author's style, (b) the fact that we, as theorists, are not able to grasp the complete range of phenomena that might be used as cues to determine authorship, and that even if we did grasp them, some of them might simply be too difficult to turn into features that a classifier could use, (d) the fact that the amount of training data is limited, which means that we have only a limited window on an author's writing style, and (e) the fact that an author's style may vary in time and depending on the circumstances. In general, no machine-learned classifier (independently of the task it has been put at, which may range from attributing authorship to filtering spam messages to forecasting volcanic eruptions) reaches 100% accuracy on non-trivial data, for the simple reason that we resort to machine learning when the task at hand is too difficult to solve deterministically.

<sup>19</sup> Available at: <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>. For example, with this conversion the word “reason” becomes “R IY1 Z AH0 N.”

## REFERENCES

- Afroz, S., Islam, A. C., Stolerman, A., Greenstadt, R., & McCoy, D. (2014). Doppelgänger finder: Taking stylometry to the underground. In *Proceedings of the 35th IEEE symposium on security and privacy (S&P 2014)* (pp. 212–226). Berkeley.
- Allen, J. H., Greenough, J. B., Kittredge, G. L., & Howard, A. A. (Eds.). (1903). *Allen and Greenough's new Latin grammar for schools and colleges*. Ginn & Company.
- Argamon, S., & Juola, P. (2011). Overview of the international authorship identification competition at PAN-2011. In *Notebook papers of the 2011 conference and labs of the evaluation forum (CLEF 2011)*. CEUR-WS.org.
- Bagnall, D. (2015). Author identification using multi-headed recurrent neural networks. In *Working notes of the 2015 conference and labs of the evaluation forum (CLEF 2015)*. CEUR-WS.org.
- Bevendorff, J., Chulvi, B., la Peña Sarracén, G. L. D., Kestemont, M., Manjavacas, E., Markov, I., Mayerl, M., Potthast, M., Rangel Pardo, F. M., Rosso, P., Stammatos, E., Stein, B., Wiegmann, M., Wolska, M., & Zangerle, E. (2021). Overview of PAN 2021: Authorship verification, profiling hate speech spreaders on Twitter, and style change detection. In *Proceedings of the 2021 international conference and labs of the evaluation forum (CLEF 2021)* (pp. 419–431). Springer.
- Bevendorff, J., Ghanem, B., Giachanou, A., Kestemont, M., Manjavacas, E., Markov, I., Mayerl, M., Potthast, M., Pardo, F. M. R., Rosso, P., Specht, G., Stammatos, E., Stein, B., Wiegmann, M., & Zangerle, E. (2020). Overview of PAN 2020: Authorship verification, celebrity profiling, profiling fake news spreaders on Twitter, and style change detection. In *Proceedings of the 2020 international conference and labs of the evaluation forum (CLEF 2020)* (pp. 372–383). Springer.
- Bischoff, S., Deckers, N., Schliebs, M., Thies, B., Hagen, M., Stammatos, E., Stein, B., & Potthast, M. (2020). *The importance of suppressing domain style in authorship analysis*. arXiv: 2005.14714.
- Boyd, R. L. (2018). Mental profile mapping: A psychological single-candidate authorship attribution method. *PLoS One*, 13(7), e0200588.
- Canapa, G. (2021). *La quantità sillabica nella computazionale authorship attribution per testi latini* (BSc thesis). University of Pisa, Pisa, Italy.
- Ceccarelli, L. (2018). *Prosodia e metrica latina classica, con cenni di metrica greca*. Società Editrice Dante Alighieri.
- Corbara, S., Moreo, A., Sebastiani, F., & Tavoni, M. (2019). The Epistle to Cangrande through the lens of computational authorship verification. In *Proceedings of the 20th international conference on image analysis and processing (ICIAP 2019)* (pp. 148–158). Springer.
- Corbara, S., Moreo, A., Sebastiani, F., & Tavoni, M. (2021). MedLatinEpi and MedLatinLit: Two datasets for the computational authorship analysis of medieval Latin texts. *ACM Journal of Computing and Cultural Heritage* (forthcoming).
- Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2, 265–292.
- Dumalus, A. F., & Fernandez, P. L. (2011). Authorship attribution using writer's rhythm based on lexical stress. In *Proceedings of the 11th Philippine computing science congress (PCSC 2011)* (pp. 82–88). Computing Society of the Philippines.
- Forstall, C., & Scheirer, W. (2010). Features from frequency: Authorship and stylistic analysis using repetitive sound. *Journal of the Chicago Colloquium on Digital Humanities and Computer Science*, 1(2), 1–23.

- Forstall, C. W., Jacobson, S. L., & Scheirer, W. J. (2011). Evidence of intertextuality: Investigating Paul the Deacon's *Angustae Vitae*. *Literary and Linguistic Computing*, 26(3), 285–296.
- Gamon, M. (2004). Linguistic correlates of style: Authorship classification with deep linguistic analysis features. In *Proceedings of the 20th international conference on computational linguistics (COLING 2004)* (pp. 611–617). COLING.
- Ginzburg, C. (1989). Clues: Roots of an evidential paradigm. In *Clues, myths, and the historical method: Works of Carlo Ginzburg* (pp. 96–214). The Johns Hopkins University Press.
- Hall, R. G., & Sowell, M. U. (1989). “Cursus” in the can Grande epistle: A forger shows his hand? *Lectura Dantis*, 5, 89–104.
- Halvani, O., Graner, L., & Regev, R. (2020). TAVeer: An interpretable topic-agnostic authorship verification method. In *Proceedings of the 15th international conference on availability, reliability and security (ARES 2020)* (pp. 1–10). ACM.
- Halvani, O., Winter, C., & Graner, L. (2019). Assessing the applicability of authorship verification methods. In *Proceedings of the 14th international conference on availability, reliability and security (ARES 2019)* (pp. 1–10). ACM.
- Harrison, R. (2021). *Cogitatorium*. Retrieved from <http://rharriso.sites.truman.edu/>.
- Ivanov, L., Aebig, A., & Meerman, S. (2018). Lexical stress-based authorship attribution with accurate pronunciation patterns selection. In *Proceedings of the 21st international conference on text, speech, and dialogue (TSD 2018)* (pp. 67–75). Springer.
- Jafariakinabad, F., Tarnpradab, S., & Hua, K. A. (2020). Syntactic neural model for authorship attribution. In *Proceedings of the 33rd conference of the Florida artificial intelligence research society (FLAIRS 2020)* (pp. 234–239). AAAI Press.
- Janson, T. (1975). *Prose rhythm in medieval Latin from the 9th to the 13th century*. Almqvist & Wiksell International.
- Johnson, K. P., Burns, P., Stewart, J., & Cook, T. (2021). *CLTK: The Classical Language Toolkit*. Retrieved from <https://github.com/cltk/cltk>.
- Juola, P. (2006). Authorship attribution. *Foundations and Trends in Information Retrieval*, 1(3), 233–334.
- Kabala, J. (2020). Computational authorship attribution in medieval Latin corpora: The case of the Monk of Lido (ca. 1101–08) and Gallus Anonymus (ca. 1113–17). *Language Resources and Evaluation*, 54(1), 25–56.
- Keeline, T., & Kirby, T. (2019). “Auceps Syllabarum”: A digital analysis of Latin prose rhythm. *Journal of Roman Studies*, 109, 161–204.
- Kestemont, M. (2014). Function words in authorship attribution: From black magic to theory? In *Proceedings of the 3rd workshop on computational linguistics for literature (CLFL 2014)* (pp. 59–66). ACL.
- Kestemont, M., Moens, S., & Deploige, J. (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), 199–224.
- Kestemont, M., Stamatatos, E., Manjavacas, E., Daelemans, W., Potthast, M., & Stein, B. (2019). Overview of the cross-domain authorship attribution task at PAN 2019. In *Working notes of the 2019 conference and labs of the evaluation forum (CLEF 2019)*. CEUR-WS.org.
- Kestemont, M., Stover, J., Koppel, M., Karsdorp, F., & Daelemans, W. (2016). Authenticating the writings of Julius Caesar. *Expert Systems with Applications*, 63, 86–96.
- Kestemont, M., Tschuggnall, M., Stamatatos, E., Daelemans, W., Specht, G., Stein, B., & Potthast, M. (2018). Overview of the author identification task at PAN-2018: Cross-domain authorship attribution and style change detection. In *Working notes of the 2018 conference and labs of the evaluation forum (CLEF 2018)*. CEUR-WS.org.
- Koppel, M., Schler, J., & Argamon, S. (2009). Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1), 9–26.
- Koppel, M., Schler, J., & Bonchek-Dokow, E. (2007). Measuring differentiability: Unmasking pseudonymous authors. *Journal of Machine Learning Research*, 8(6), 1261–1276.
- Koppel, M., & Winter, Y. (2014). Determining if two documents are written by the same author. *Journal of the Association for Information Science and Technology*, 65(1), 178–187.
- Lagutina, K., Lagutina, N., Boychuk, E., Larionov, V., & Paramonov, I. (2021). Authorship verification of literary texts with rhythm features. In *Proceedings of the 28th conference of the Finnish-Russian university cooperation in telecommunications (FRUCT 2021)* (pp. 240–251). IEEE.
- Leonard, R. A., Ford, J. E., & Christensen, T. K. (2017). Forensic linguistics: Applying the science of linguistics to issues of the law. *Hofstra Law Review*, 45(3), 11.
- Luyckx, K., & Daelemans, W. (2008). Authorship attribution and verification with many authors and limited data. In *Proceedings of the 22nd international conference on computational linguistics (COLING 2008)* (pp. 513–520). COLING.
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153–157.
- Mendenhall, T. C. (1887). The characteristic curves of composition. *Science*, 9(214), 237–249.
- Mosteller, F., & Wallace, D. L. (1963). Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed Federalist Papers. *Journal of the American Statistical Association*, 58(302), 275–309.
- Neidorf, L., Krieger, M. S., Yakubek, M., Chaudhuri, P., & Dexter, J. P. (2019). Large-scale quantitative profiling of the Old English verse tradition. *Nature Human Behaviour*, 3(6), 560–567.
- Oberhelman, S. M., & Hall, R. G. (1984). A new statistical analysis of accentual prose rhythms in imperial Latin authors. *Classical Philology*, 79(2), 114–130.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Plecháč, P. (2021). Relative contributions of Shakespeare and Fletcher in Henry VIII: An analysis based on most frequent words and most frequent rhythmic patterns. *Digital Scholarship in the Humanities*, 36(2), 430–438.

- Savoy, J. (2019). Authorship of Pauline epistles revisited. *Journal of the Association for Information Science and Technology*, 70(10), 1089–1097.
- Sidorov, G. O. (2018). Automatic authorship attribution using syllables as classification features. *Rhema*, 1, 62–81.
- Spinazzè, L. (2014). “Cursus in Clausula”: An online analysis tool of Latin prose. In *Proceedings of the 3rd AIUCD annual conference on humanities and their methods in the digital ecosystem (AIUCD 2014)* (pp. 1–6). ACM.
- Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3), 538–556.
- Stamatatos, E. (2016). Authorship verification: A review of recent advances. *Research in Computing Science*, 123, 9–25.
- Stamatatos, E. (2018). Masking topic-related information to enhance authorship attribution. *Journal of the Association for Information Science and Technology*, 69(3), 461–473.
- Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., & Stein, B. (2015). Overview of the author identification task at PAN 2015. In *Working notes of the 2015 conference and labs of the evaluation forum (CLEF 2015)*. CEUR-WS.org.
- Stover, J. A., Winter, Y., Koppel, M., & Kestemont, M. (2016). Computational authorship verification method attributes a new work to a major 2nd century African author. *Journal of the Association for Information Science and Technology*, 67(1), 239–242.
- Sturtevant, E. H. (1922). Syllabification and syllabic quantity in Greek and Latin. *Transactions and Proceedings of the American Philological Association*, 53, 35–51.
- Tognetti, G. (1982). *Criteri per la trascrizione di testi medievali latini e italiani, volume 51 of Quaderni della Rassegna degli Archivi di Stato*. Ministero per i beni culturali e ambientali.
- Toynbee, P. (1918). Dante and the cursus: A new argument in favour of the authenticity of the “Quaestio de Aqua et Terra”. *Modern Language Review*, 13(4), 420–430.
- Tuccinardi, E. (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), 435–447.
- van der Goot, R., Ljubešić, N., Matroos, I., Nissim, M., & Plank, B. (2018). Bleaching text: Abstract features for cross-lingual gender prediction. In *Proceedings of the 56th annual meeting of the association for computational linguistics (ACL 2018)* (pp. 383–389). ACL.
- Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep-learning-based natural language processing. *IEEE Computational Intelligence*, 13(3), 55–75.
- Yule, G. U. (1939). On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. *Biometrika*, 30(3/4), 363–390.
- Zheng, R., Li, J., Chen, H., & Huang, Z. (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), 378–393.
- Zugarini, A., Melacci, S., & Maggini, M. (2019). Neural poetry: Learning to generate poems using syllables. In *Proceedings of the 28th international conference on artificial neural networks (ICANN 2019)* (pp. 313–325). Springer.

**How to cite this article:** Corbara, S., Moreo, A., & Sebastiani, F. (2022). Syllabic quantity patterns as rhythmic features for Latin authorship attribution. *Journal of the Association for Information Science and Technology*, 1–14. <https://doi.org/10.1002/asi.24660>