Physica A 598 (2022) 127387

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Finding contrasting patterns in rhythmic properties between prose and poetry

Henrique Ferraz de Arruda [a,b,1], Sandro Martinelli Reia [a,c], Filipi Nascimento Silva [d], Diego Raphael Amancio [e,*], Luciano da Fontoura Costa [a]

[a] São Carlos Institute of Physics, Universidade de São Paulo, São Carlos, Brazil
[b] ISI Foundation, Turin, Italy
[c] Lyles School of Civil Engineering, Purdue University, West Lafayette, USA
[d] Indiana University Network Science Institute, Bloomington, USA
[e] Institute of Mathematics and Computer Sciences, Universidade de São Paulo, São Carlos, Brazil

ARTICLE INFO

Article history:
Received 15 October 2021
Received in revised form 30 March 2022
Available online 16 April 2022

Keywords:
Complex systems
Text analysis
Text classification
Time Series
Machine learning
Neural networks

ABSTRACT

Poetry and prose are written artistic expressions that help us appreciate the reality we live in. Each of these styles has its own set of subjective properties, such as rhyme and rhythm, which are easily caught by a human reader's eye and ear. With the recent advances in artificial intelligence, the gap between humans and machines may have decreased, and today we observe algorithms mastering tasks that were once exclusively performed by humans. In this paper, we propose a computational method to distinguish between poetry and prose based solely on aural and rhythmic properties. In order to compare prose and poetry rhythms, we represent the rhymes and phonemes as temporal sequences, and thus, we propose a procedure for extracting rhythmic features from these sequences. The performance of this procedure is evaluated by the use of popular machine learning classifiers, and the best accuracy was obtained with a multilayer perceptron neural network. Interestingly, by using an approach based on complex networks to visualize the similarities between the different texts considered, we found that the patterns of poetry vary more than prose. Consequently, a richer and more complex set of rhythmic possibilities tends to be found in that modality.

1. Introduction

It has been frequently observed that arts and science share many characteristics, especially creativity. Consequently, a continuous search for innovation underlies both these areas, giving rise to new approaches and conventions. At the same time, these works are typically subsumed into major areas. While we have research areas such mathematics, humanities, biology, etc., in arts we have styles and genres. The classification of specific works in major genres requires the respective works to share some similar characteristics. Therefore, the organization of works of arts into genres and styles is characterized by an interesting coexistence of dissimilarity (required for innovation) and similarity (required for being

* Corresponding author.
E-mail addresses: hfarruda@gmail.com (H. Ferraz de Arruda), smreia@gmail.com (S.M. Reia), filsilva@iu.edu (F.N. Silva), diegoraphael@gmail.com (D.R. Amancio), ldfcosta@gmail.com (L. da Fontoura Costa).

[1] This author was at the *São Carlos Institute of Physics, University of São Paulo*, until 31st May 2021.

https://doi.org/10.1016/j.physa.2022.127387
0378-4371/

grouped into a same category). In other words, the classification of works of art needs to take into account an interplay between homogeneity (within a group) and heterogeneity (between groups). However, even the works belonging to a same group will present some dispersion, reflecting the creativity and innovation aspects expected from works of art. The study of these structures represent an interesting and important endeavor that has progressively been approached by using computational concepts and methods [1].

Within the related literature, two major areas have been typically identified: prose and poetry. Each of these have been extensively developed along centuries, giving rise to a large number of masterpieces, while contributing to human culture. Poetry has been frequently described as a literary form emphasizing rhythm and rhymes, while prose would not involve so much attention to these two aspects. Yet, every piece of prose will incorporate some level of rhythm and rhyme, to the point that a specific genre, namely prose poetry, has also been identified and developed. Interestingly, while humans seem to have some intuitive ability to distinguish between artistic and literary styles and genres, it remains an interesting and relatively challenging question to understand in a more objective and quantitative manner the two major areas of prose and poetry.

Some researchers have also considered the classification of poetry. For instance, Jamal et al. [2] found that Support Vector Machines can classify poems into different classes via a bag-of-words approach. Moreover, Gopidi and Alam [3] have used grammar, meter and rhyme as features, and Random Forest and KNN as classifiers, to find that the similarity between poetry and prose can vary according to time. By using information theory, Calin [4] showed that the entropy associated to English poetry changes with time, and also that the entropy depends on the language and on the author considered. Other researches dealt with the problem of automatically generate poetry [5,6]. For instance, Tikhonov and Yamshchikov [5] proposed long short-term memory artificial neural network with phonetic and semantic embeddings to generate stylized poetry. In the latter study, they found that the poetry generated by their method outperforms random and baseline models. Furthermore, the conversion from poetry to prose have also been studied [7].

Costa and Arruda [8] studied a model of the relationship musical notes in terms of harmonic series, which was represented as a complex network. In another study, researchers considered the analysis of patterns of the poetry sounds [9]. The differences between the creative thinking and the conceptual representations of the human mind when it comes to prose and poetry were explored in [10]. The authors studied poetic texts from Dylan Thomas and John Gay, and prose texts from F. Scott Fitzgerald and George Orwell. They found that poetry has a wider distribution of conceptual associations than prose, and a more complex scenario is drawn when a semantic network and a neural network are considered.

In this work, we propose a method for extracting the musical patterns – namely rhymes and rhythm – from written, artistic expressions, such as prose and poetry. Relatively to the current literature about rhythmic properties of prose and poetry, the classification of texts based solely on rhythmical features remains a subject of interest. There are several motivations for developing computational approaches capable of distinguishing between poetry and prose. These include the incorporation of such a resource in browsers and recommendation systems, which would allow more specific and accurate search for genres of texts. In addition, this resource is essential for performing research related to the characterization of literary works, as well as for providing insights about the way in which humans distinguish between these two types of texts.

First, texts are converted into temporal sequences of phonemes, giving us the ability to identify the existing rhymes in a given time window. Second, we propose a set of features aimed at extracting the rhythmic patterns from the phonetic sequences, which are later used by the classification algorithms employed to discriminate between the two classes considered (prose and poetry). In order to avoid spurious effects on our findings, we consider prose and poetic extracts with similar sizes so as to focus on their inherent construction/structure instead of their sequence length.

Our results indicate that the classifiers we tested here were able to successfully identify the type of text under consideration with an accuracy of at least 75%. Interestingly, the visualization of the similarities among the items of our corpus via complex networks reinforces the idea that the proposed algorithm for feature extraction can grasp meaningful information. On the one hand, the networks show that prose texts are more densely connected, meaning they usually share the same rhythmic patterns. On the other, poetry texts are weakly connected, meaning that poetry may present a wide range of rhythmic patterns, so it is less likely to find two poetic texts with the same rhyme pattern.

The effect of the structure on the feature extraction algorithm is further explored by comparing the original poetry and prose texts with their shuffled versions. For poetry, this experiment reveals that the shuffling of the words (while the punctuation remains fixed) results in an accuracy of about 60%. So, there is no evident difference between both classes in terms of the considered features. The accuracy value, found to be slightly higher than the null case (50%), indicates that the structure alone plays a marginal role in defining poetry. For the prose, we found that the accuracy between the classification of prose and shuffled prose is about 70%, meaning that in this case, the order of words is more relevant for their characterization.

The present paper is organized as follows. In Section 2, we describe the materials and methods used here, including details about the dataset used in our analysis, the description of how we represent the data, the proposed method for feature extraction, the classification algorithms we use, and an overview of the network representation method for visualizing the similarities between the texts. Our findings are presented in Section 3, where we compare the texts of our corpus by looking at their basic statistics and analyze the performance of the classification algorithms. Finally, in Section 4 we offer our concluding remarks along with the perspectives for future works.

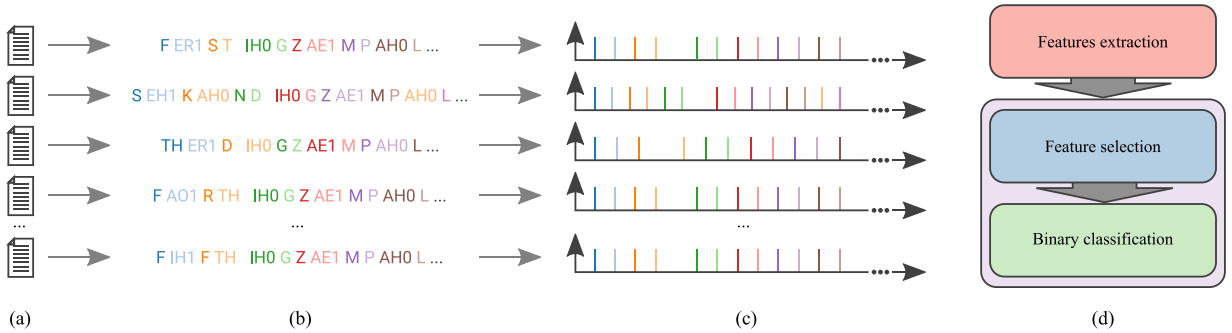

Fig. 1. Pipeline of the proposed analysis. (a) Set of texts to be analyzed. (b) The texts are converted into sequences of phonemes. (c) The phone sequences are represented as sequences of phoneme repetitions. (d) The sequences obtained in (c) are classified into poetry and prose.

2. Materials and methods

In this section, we present the employed datasets and the methodology used to represent texts as sequences. Fig. 1 illustrates the proposed pipeline of analysis. The employed texts, as well as the used dictionary of phonemes are described in Section 2.1 (see Fig. 1(a) and (b)). In Section 2.2, we describe the methodology for representing the data and the measured features, as illustrated in Fig. 1(c). Furthermore, in Section 2.4, we describe the feature selection and the classifiers used to classify sequences into prose or poetry. Afterward, each text is characterized by a set of features extracted from the corresponding sequence of phonemes. Since some features might not be relevant to discriminate between poetry from prose writing styles, we used a feature selection algorithm to identify the ones that contribute the most to our goal. These methodological steps are described in Section 2.4 and illustrated in Fig. 1(d).

2.1. Employed data

The dataset considered here comprises extracts of texts with similar length. The majority of the samples are obtained from the Project Gutenberg website Gut [11], which is an online library that makes available over 60,000 ebooks. In particular, the *poetry* corpus is equally composed of odes, ballads and sonnets from different books, in a total of 60 samples. The *prose* corpus encompasses the same number of samples of technical, novel books, and pieces of news (20 of each). However, the latter were obtained from the Brown Digital Repository Bro [12]. More specifically, we select pieces of texts from the 20 first texts in the category of *news*. More details regarding the dataset characteristics are shown in Section 3.1.

Words from the text samples (Fig. 1(a)) are tokenized with the natural language toolkit (NLTK) python package [13], and the tokens are converted into phonemes with the use of the pronouncing python library[2] (Fig. 1(b)). The pronouncing library is based on the *Carnegie Mellon University Pronouncing Dictionary*[3] that is an open-source pronunciation dictionary for North American English containing about 134 thousand words and their pronunciations. It is useful for speech recognition since it maps words phonemes their pronunciations in the ARPAbet phoneme set [14–16], having 39 phonemes for standard English pronunciation.

As a result, each text is represented as a sequence of phonemes as shown in Fig. 1(c), in which each colored vertical bar represents a different phoneme. The intervals between phonemes take into account the unities of time presented in Table 1, which is further explained in the next section.

2.2. Data representation

A music representation capable of quantifying rhythmic and aural patterns was proposed by Costa [17]. We propose a similar methodology, but here we aim at comparing texts by considering their rhyme and rhythmic structure. For this purpose, we introduce a methodology based on *phonemes* and *rhymes*. The central concept here is to characterize the temporal distribution of rhymes, which we believe can be related to rhythm in sounds. Fig. 2 illustrates our approach.

The representation is created for each text separately. The entire algorithm is shown in Fig. 3. The method starts with a text (one example of text is shown in Fig. 2(a)). First, a simple pre-processing step is executed, in which consecutive break lines are reduced into a single break line. Next, we identify all tokens in the text, which include punctuation marks, numbers, and line breaks. For all tokens, the respective phonemes are found; see Fig. 2(b). No phonemes are attributed for punctuation and other tokens without a respective phoneme in the dictionary. The set of tokens preceded by fixed punctuation are selected, and all words that rhyme with these tokens identified, as represented in Fig. 2(b). We say that

2 https://github.com/aparrish/pronouncingpy.

3 http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

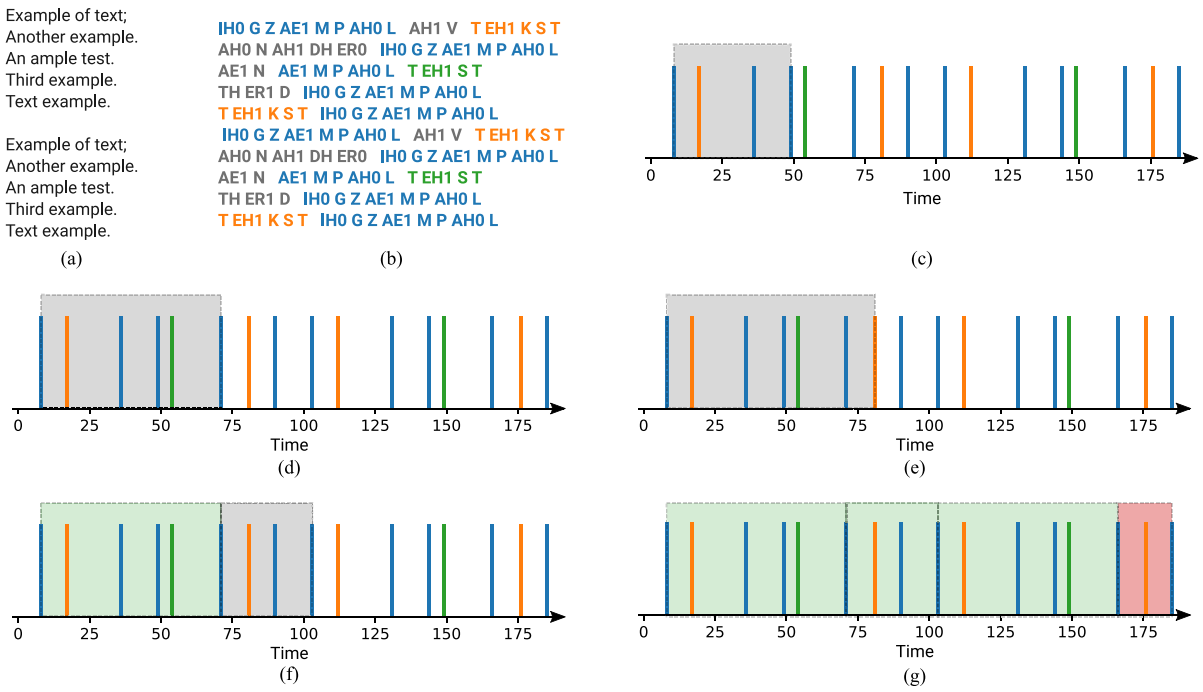

Fig. 2. Example of the proposed approach. (a) Text to be processed. (b) Tokens converted into phonemes and colors representing rhymes. The gray phonemes, non-related to rhymes, are not considered in the method. Next, from (c) to (g), an example of how the windows are defined is shown, in which the green windows are detected. The bars represent words that rhyme within the text, where the colors indicate the different rhymes. The red window, shown in (g), is not considered. Here we employ the following parameters: $L_0 = 2$ and $\Delta = 0.2$.

Table 1
Unities of time defined for the considered *rhythm punctuation*.

Symbol	Unities of time
,	3
.	4
;	4
!	5
?	5
–	5
–	5
break line	1

two words rhyme if both words have the same last phoneme. This set of words is henceforth referred to as *rhyme words*. In this study, we considered the punctuation set as: ".", ":", ";", "!", and "?". This punctuation set is called *rhythm punctuation*. Further details regarding the choice of *rhythm punctuation* are shown in Section 3.1.

We define the time scale in terms of the number of phonemes for each token, so that one phoneme correspond to a single time unit. In the case of words that are not part of the dictionary of phonemes, we added one unit of time. Furthermore, one unit of time is added between consecutive words. For punctuation and break lines, we considered the dictionary shown in Table 1.

By considering the position of the phonemes in the time scale, a signal is assigned in the position of the last phoneme of each *rhyme word*. The rhymes are discriminated by type; all the words that rhyme are represented by the same signal type. The step of the definition of the time series, which includes the information of rhymes and the time unities, is represented by the function *find_time_series* of Fig. 3. One example of temporal representation is shown in Fig. 2(c), and the rhymes are discriminated by considering different colors.

With the rhyme sequence in hand, we clustered the signals with a similar variety of gaps into windows. The clustering defined by the while loop of the algorithm is shown in Fig. 3. This method begins with a window that incorporates L_0 pairs of consecutive signals with the same type. Note that it is possible to have signal pairs with different rhyme types. The window begins at the first signal. In the example of Fig. 2(c), we used $L_0 = 2$, and the two pairs are defined between blue signals. By considering this window, the coefficient of variation is calculated for the time differences between signals, which is defined as $cv(T_w) = \sigma/\mu$, where T_w is a vector with the considered time differences of the window w, and μ

Data:
T; //input text.
L_0; //Initial number of pair of consecutive signals.
Δ; //parameter Δ.
Begin

W; //Starts as an empt list.

$T \leftarrow preprocess(T)$;

$tokens \leftarrow tokenize(T)$; // tokenize the text.

$phonemes \leftarrow find\_phones(tokens)$

$t_s \leftarrow$ find_time_series($tokens$, $phonemes$);

$w \leftarrow$ find_initial_window(t_s, L_0); //w stores the initial and final positions of the window.

$T_w \leftarrow$ find_time_differences(w, t_s);

while there are possible windows **do**:

$w_2 \leftarrow$ find_the_next_window(t_s, w);
$T_{w_2} \leftarrow$ find_time_differences(w_2, t_s);
if $|cv(T_w) - cv(T_{w_2})| > \Delta$ **then**:
w is inserted in W;
$w \leftarrow$ start_new_window(t_s, L_0, w); //the final position of the previous window (w) is the first in the new window.
$T_w \leftarrow$ find_time_differences(w, t_s);
else:
$w \leftarrow w_2$;
$T_w \leftarrow T_{w_2}$;

End

Fig. 3. Pseudocode of the proposed algorithm for clustering the signals.

and σ are average and standard deviation of the T_w, respectively. In Fig. 3, the coefficient of variation is represented by the function cv.

For each step, new signals are incorporated into the window until another pair of related signals is obtained (given by the function *find_next_window* in the algorithm of Fig. 3). This new window w_2 gives rise to another set of time differences T_{w_2}. Next, cv is calculated for both T_w and T_{w_2}. Another variable, Δ, is defined to represent the maximum difference between T_w and T_{w_2}. More specifically, if $|cv(T_w) - cv(T_{w_2})| > \Delta$ is reached, the process stops, the window w is stored in W, and a new window starts from the last signal of w. Otherwise, T_w is replaced by T_{w_2} and the process resumes into its signal-pairing stage. The algorithm finishes when there is no possibility to create a new window. Furthermore, if a window does not finish at the end of the algorithm, it is not added to W.

In the example of Fig. 2, the first window begins with two pairs of the blue signal, as shown in Fig. 2(c). The next possible signal is tested in Fig. 2(d), and the difference of cv is lower than Δ. So, the next possible window is tested (see Fig. 2(e)). In this case, the orange signal is added, which gives rise to a relatively high time difference. Consequently, the window, w, finishes, and this signal is not added to W. Fig. 2(f) shows the first defined window in green, and the start of a new window, in gray. The signal taking part of the end of a window is the first in the next one. This process is repeated for all possible signals, and three windows in green are created (see Fig. 2(g)). In red, we illustrate the signals that did not give rise to a new window.

2.3. Data characterization

We propose some metrics to analyze the time sequences and the window sizes identified in the previous section. These measurements are employed to characterize the texts and, as the next step, to compare between *poetry* and *prose*. In the following, we itemize the employed measurements:

- μ_l: mean of the time intervals between the first and the latter signal in the windows;
- $cv(l)$: coefficient of variation of the time intervals between the first and the latter signal in the windows;

- μ_d: mean of the differences between pairs of consecutive signals of the same type, which is calculated for each window;
- σ_d: the standard deviation of the differences between pairs of consecutive signals of the same type, which is calculated for each window;
- $\mu_l \times cv(l)$: in order to understand if there is a relationship by considering both quantities together, we also considered $\mu_l \times cv(l)$.

Because the measurements are computed for each detected window, we considered both the average and standard deviation of the described features to characterize documents.

2.4. Classification

In order to identify the characteristics associated with each type of text, we used feature selection algorithms. Furthermore, the quality of this set of features is quantified by considering some different classifiers. All these methodologies are described in this section.

As an attribute selection, we use the *Information gain* [18,19], which is based on information theory. This is supervised approach and consists in a comparison between the employed feature with the classes. In order to quantify the relationship between features and classes, the normalized mutual information (NMI) [19] is calculated. All the features are ranked according to their NMI. We chose this approach since the features are computed separately. In this fashion, we can better understand their relationship with the obtained rhyme sequences.

We use five classifiers based on different assumptions. Thus, we can identify if the obtained results are consistent among different classification techniques [20]. In the following, we list the employed classifiers, along with the considered parameters:

- *LDA*: the Linear Discriminant Analysis (e.g. [21]) attempts to find a linear combination of features that can be used to classify two or more classes. In this case, we considered a single LDA dimension;
- *RF*: the Random Forest method (e.g. [22]) considers an ensemble of decision trees that are merged to yield a more accurate prediction or classification. We set the maximum depth of the tree as 2 and the random state as 0. The remaining parameters were set as default;
- *KNN*: the K Nearest Neighbors classifier (e.g. [23]) basically assumes that similar objects are closer to each other according to some metrics (such as Euclidean distance in multidimensional space). A parameter K consists of the number of considered neighbors. Here we set $K = 5$, and the remaining parameters as default;
- *SVM*: the Support Vector Machine (e.g. [24]) tries to find the right hyper-planes that maximizes the distance between it and the objects in the training set. We employed a linear kernel, and other parameters are set as default;
- *MLP*: the Multi-layer Perceptron (e.g. [25]) is a multilayer artificial neural network. We set the maximum number of iterations and the hidden layer sizes as 10,000 and 40, respectively. The main used parameters are: the solver optimization is Adam (with $\beta_1 = 0.9$ and $\beta_2 = 0.999$), the activation function is *relu*, and the initial learning rate is 0.001 [26,27]. The remaining parameters were set as default.

Because the classifiers we used here are fundamentally different from each other, we believe the individual results we obtained are complementary thus leading to a better perspective of the classification problem. All the features were standardized before the classification.

In order to avoid overfitting we use the leave-one-out as cross-validation [28]. More specifically, the training set is defined with all samples, except one considered the test. The same process is repeated, and, in separated steps, all features are considered as being the training set. We consider the standardization, attribute selection, and classification model fitted only with the training set in this process. All the methods presented in this section were implemented by using the scikit-learn [29] in Python language.

2.5. Networked approach

In order to better understand the relationship between the analyzed classes, we compare the proposed representation by using a network-based approach. More specifically, we visualize the similarity texts by mapping the corpus into a complex network [30]. In this case, each node i of the networks represents a text, with its own set of features v_i. Then, two nodes i and j are linked with an edge weighted according to the cosine similarity [31] between the corresponding set of features v_i and v_j. The obtained network is visualized by using a *force directed algorithm* [32], implemented by Silva et al. [33]. In order to better understand the relationship among samples, we remove the edges with weights lower than a threshold τ.

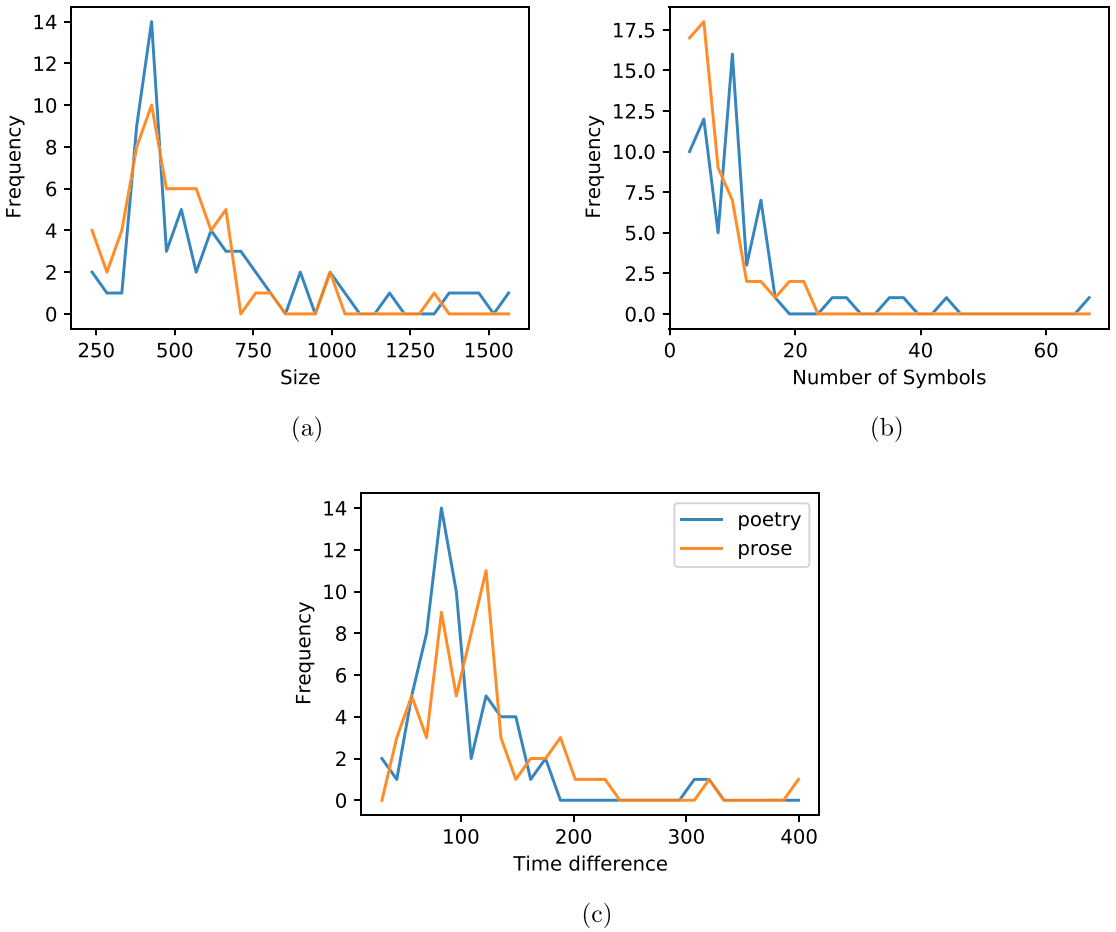

Fig. 4. Histograms of the basic statistics. (a) Sizes of all proses and poetries in terms of the number of phonemes and characters. (b) Histogram of *rhythm punctuation*. (c) Histogram of the average differences between the consecutive considered punctuation set (“.”, “:”, “;”, “!”, and “?”). In this case, we considered the difference in terms of the number of characters.

3. Results and discussions

In this section, we present the results regarding the similarities between poetry and prose. We considered the performance of the classification algorithms to discriminate *poetry* from *prose* based on the set of proposed rhythmic features. For this purpose, we begin analyzing the dataset and describing a few basic statistics in Section 3.1.

Once the poetry and prose corpus are characterized, we address the relationship between them in Section 3.2, in which we show that the features collection/extraction method we are proposing can capture rhythmic patterns since the Precision, Recall, and Accuracy are higher than the random baseline (0.50). Finally, in Section 3.3, we propose a null model to explore the robustness of our findings with respect to the text structure and the words chosen by the authors.

3.1. Dataset analysis and basic statistics

Before performing the analysis regarding the data representation (described in Section 2.2), we briefly describe a few basic information regarding the employed data. First, in order to certify that the text length is not influencing the analysis, we select texts for both classes with similar numbers of phonemes, as shown in Fig. 4(a). Another essential piece of information is the set of punctuation to be considered as the *rhythm punctuation*. In this case, we searched for a particular subset of punctuation symbols (from the set shown on Table 1) whose frequency is similar for prose and poetry. Fig. 4(b) illustrates the histogram by considering the following set of symbols: “.”, “:”, “;”, “!”, and “?”.

Similar results were found for the comparison between the interval between the considered punctuation. In order to consider the simplest statistics possible, we employ the differences in terms of number of characters. As can be seen in Fig. 4(c), the distributions were found to be similar for both classes.

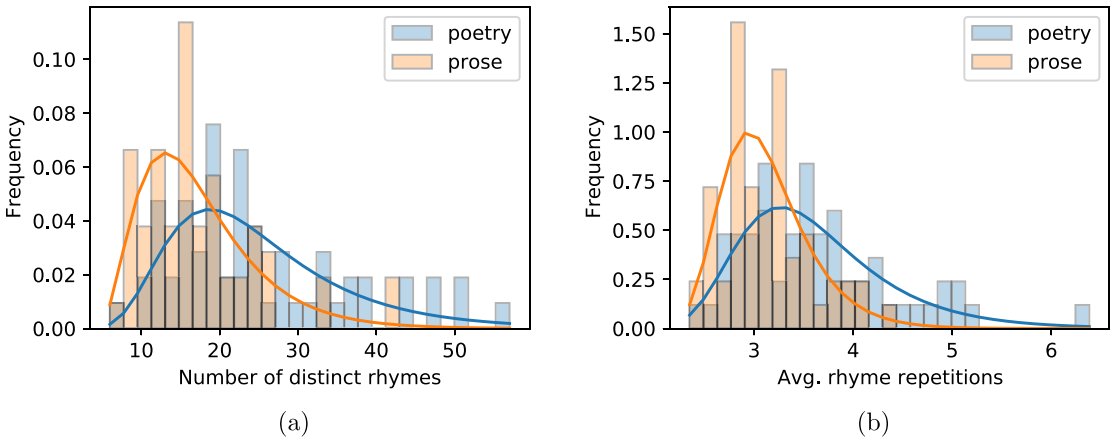

Fig. 5. Histograms representing patterns of rhymes. The lines indicate that the empirical data are well described by Weibull distributions. Under the null hypothesis that the empirical and Weibull distributions are identical, the *KStest* returns a p-value > 0.74 for all the cases.

All in all, the results shown in Fig. 4 shows that the distributions for both classes are similar. So, by considering the frequencies and punctuation presented here, the employed dataset seem not to influence the results shown in the comparisons described in the following sections.

Since our methodology is based on rhymes, we compared both classes in terms of their distributions (see Fig. 5). More specifically, in Fig. 5(a), we plot the histograms of the number of distinct rhymes, and in Fig. 5(a) the average number of rhyme repetitions. In contrast with the previously presented results, here, the histograms are visually different. To show these differences, for each histogram, we fit a curve of the Weibull distribution [34]. For both measurements, it is possible to note that the histograms regarding prose tend to be more concentrated on the left side of the plot. Furthermore, poetry tends to have a more spread pattern of rhymes.

3.2. Comparison between poetry and prose

For each text, we create the respective phoneme time sequences as described in Section 2.2. In order to better understand the differences between the classes, we classified the sequences with the methods presented in Section 2.4. We start by using the *information gain* as an attribute selection to rank the features according to their relevance. For all tests we considered the following set of parameters: $L_0 \in \{2, 5, 10, 15, 20\}$ and $\Delta \in \{0.01, 0.05, 0.10, 0.15, 0.20\}$.

In Fig. 6 we show the frequency distribution of the most relevant feature across the poetry and prose corpus. While the average $cv(l)$ is spread in a wide range of values in the poetry corpus, this metric is highly concentrated close to zero for the prose texts. It means that the proposed algorithm for clustering signals, described in Section 2.2, was not able to detect a window due to one of the following reasons: (i) a pair of signals of the same type were not found in the whole text or (ii) the time distances are too regular and, consequently, the method finished without enclosing a window. Similar results were found for the other features with high values *information gain*.

In order to compare the classes, we employ all the classification methods proposed in Section 2.4. Because we are interested in the information obtained by the proposed features, we test the classifiers to obtain the best accuracy while using the smallest number of features. More specifically, we executed the classifiers for different numbers of features, from 4 to 50 features, which were ordered according to the *information gain* attribute selection.

Table 2 illustrates the results obtained for the classifications. Interestingly, the obtained values of accuracy are similar. In spite of its simplicity, LDA obtained a relatively high accuracy using only the $n_f = 4$ most relevant features, being outperformed only by the MLP with $n_f = 15$. However, the variation of the results of precision and recall were found to be higher than for the values of the accuracy. In the case of LDA, we observe the highest value of precision in the classification of poetry. In this case, a high value of precision means that when a text is classified as poetry, there is a high chance that the classification is correct. On the other hand, a high value of recall means that the classification method is correctly classifying all the poetry texts, which is the case of SVM. In the case of prose, SVM presents the highest precision, and LDA presents the highest recall.

Despite the differences in performance, the rhythmic-based features were found to properly describe the two types of text. More specifically, independently of the nature of the employed classifier, relatively high values of accuracy were obtained. It is important to highlight that the aim of this paper is not to propose features that outperforms competing approaches in classification. Here we are more interested in demonstrating that the characteristics of rhythm can be measured. Departing from the premise that poetry and prose have different rhythms when read, our method successfully captured these differences.

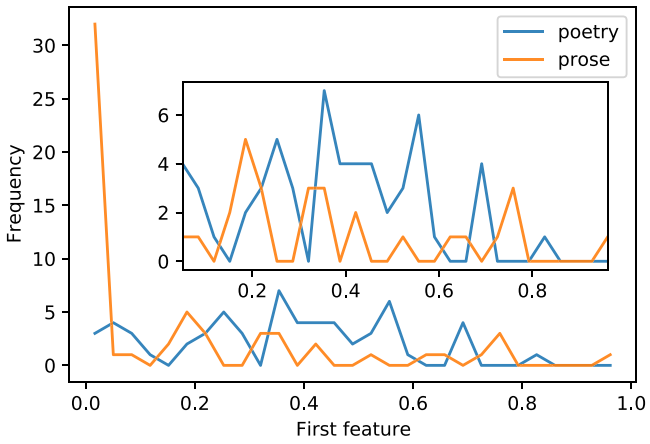

Fig. 6. Histogram of the average of $cv(l)$, which was considered by *information gain* as being the most important. The employed parameters are: $L_0 = 2$ and $\Delta = 0.1$. The inset depicts a zoom for values higher than zero.

Table 2
Performance of classifiers LDA, RF, KNN, SVM and MLP on poetry and prose extracts.

Classifier	n_f	Precision		Recall		Accuracy
		poetry	prose	poetry	prose	
LDA	**4**	**0.81**	0.74	0.70	**0.83**	0.77
RF	36	0.73	0.77	0.78	0.72	0.75
KNN	5	0.73	0.78	0.80	0.70	0.75
SVM	14	0.72	**0.80**	**0.83**	0.68	0.76
MLP	15	0.79	0.76	0.75	0.80	**0.78**

In Fig. 7, we present the data analysis through a network science approach (as described in Section 2.5). By considering the most relevant features (measured via *information gain*), we depict three different numbers of features. In the first, Fig. 7(a), only the four features used for the LDA classifier were considered. As can be seen, there is natural segregation among nodes of poetry and prose. We also considered 15 features, which gave rise to the best result using the MLP classifier (see Fig. 7(b)). In comparison with the previous case, with 15 features, poetry tends to be much more spread on the visualization. A similar result was found when all features were considered, as shown in Fig. 7(c). Interestingly, for the cases of Fig. 7(b) and (c), poetry was found to be more diverse than prose in terms of its feature vectors.

The performance of the features is also captured by $a_i = (\sum_{j\in N_i} \delta_{ij} w_{ij})/(\sum_{j\in N_i} w_{ij})$, in which N_i is the set of nodes connected to i, w_{ij} is the weight (similarity) of the link between nodes i and j, and $\delta_{ij} = 1$ if nodes i and j belong to the same text type (prose or poetry) or $\delta_{ij} = -1$ otherwise. Node that $a = 1$ if a node is only connected to nodes of the same type, and $a = -1$ if a node is only connected to nodes of the other type. Fig. 8 shows the rank plot of a_i for the three networks exhibited in Fig. 7. We observe that the percentages of nodes with $a > 0$ are higher than the baseline (50%) for the three networks, indicating strong ties between nodes representing the same text type. The high concentration of a around 0 for network (a) indicates that a high fraction of nodes are similarly connected to both text types, suggesting a mix between the different nodes. The area between a and the zero (dashed) line for networks (b) and (c) indicates that there is a clear separation between both text types, with a small fraction being connected to nodes representing different text types.

3.3. Comparison with random texts

In this section, we compare the prose and poetry of original texts with their respectively shuffled versions. We also compared between both shuffled versions. In order to create the shuffled version, we identified all tokens in the text. Next, the position of the punctuation was fixed, and the remaining tokens were shuffled. The measurements shown in the following tables represent averages and standard deviations for separated classifications with 50 distinct versions of the shuffled texts. This number of samples is good enough to allow us to observe the convergence to the mean of the performance of the classifiers. In order to reduce the computational cost, we test the classifiers with a subset comprising the most relevant features n_f. In particular, we employed between 4 and 36 features.

First, we analyze the distribution of the features. Fig. 9 illustrates the first feature selected by *information gain* for a single execution. Notice that the order of the features changes for each execution. Both the samples of original and shuffled poetry are highly overlapping. The next 10 selected features present similar distributions. This characteristic of

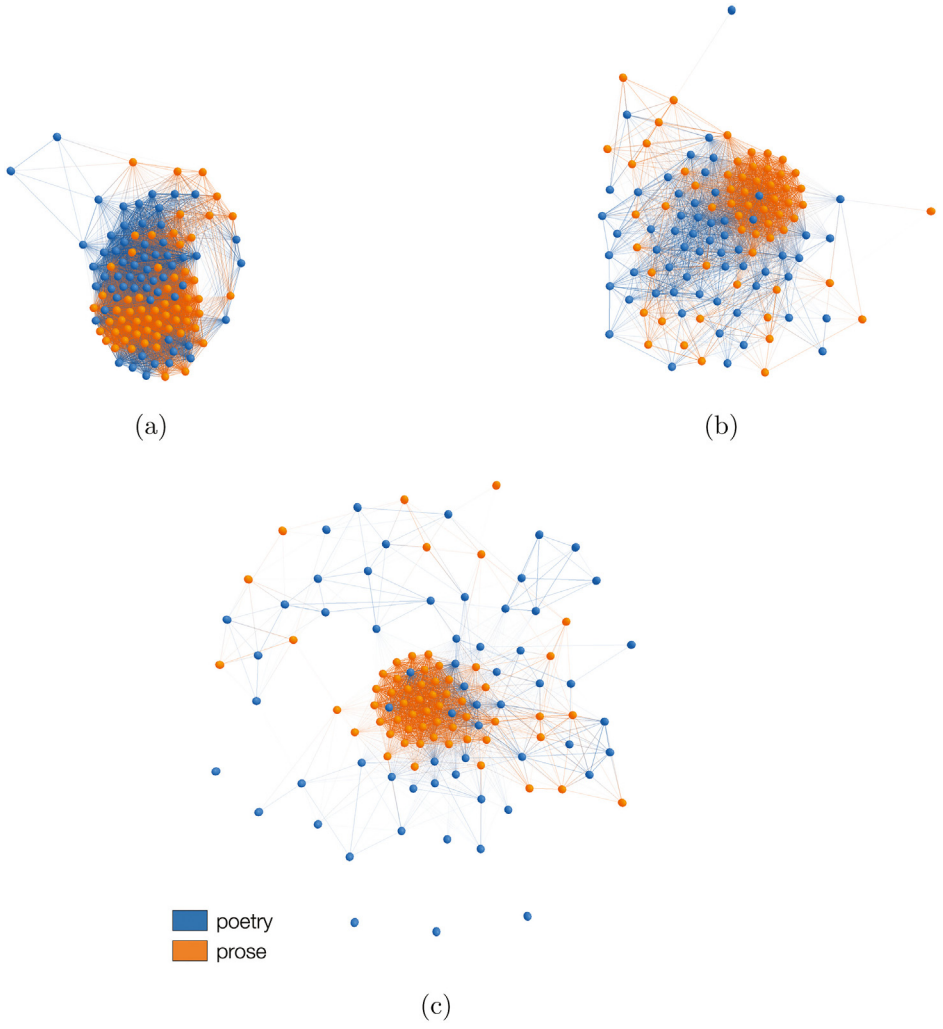

Fig. 7. Visualization of the networks, where nodes represent the similarities between the features of each text items (a), (b), and (c) represent 4, 15, and all features, respectively. Here, we removed all edges with weight lower than $\tau = 0.5$.

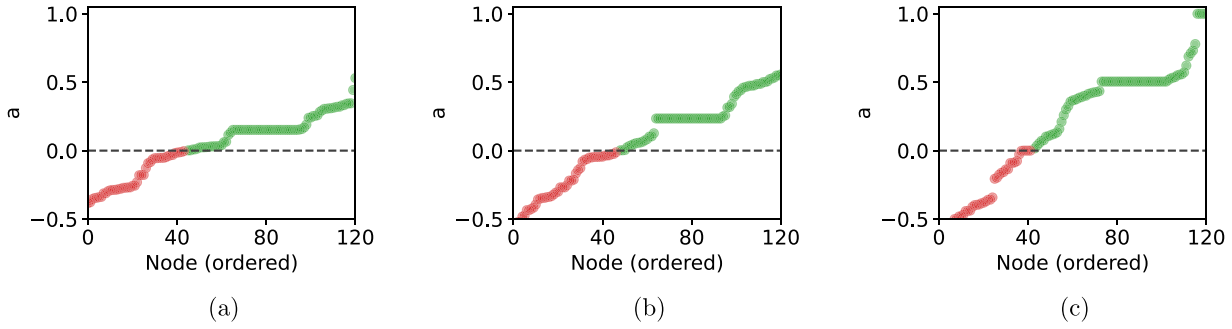

Fig. 8. Measurements of a for the networks described in Fig. 7, in which items (a), (b), and (c) represent 4, 15, and all features, respectively. The nodes under the dashed line (in red) represent values lower than zero. The percentages of a values above zero are 61% 63%, and 65% for (a), (b), and (c), respectively.

shuffled poetry to be similar to the original texts is also reflected on the classification results (see Table 3). All in all, the classification quality measurements are much worse than the cases presented in the previous section. Since in poetry, the

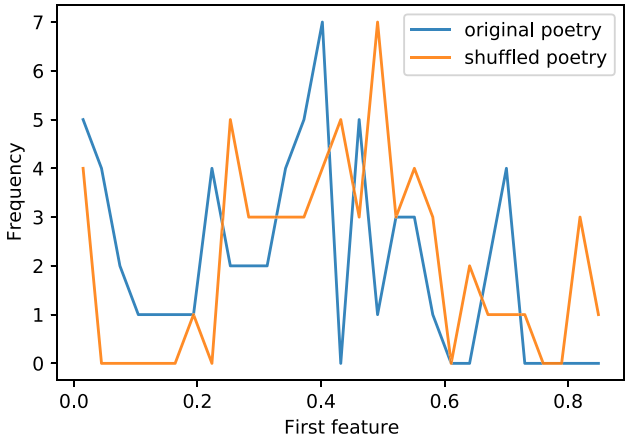

Fig. 9. Histogram of the average of $cv(l)$, which was considered by *information gain* as being the most important. The employed parameters are: $L_0 = 2$ and $\Delta = 0.15$.

Table 3
Performance of classifiers LDA, RF, KNN, SVM and MLP on the original and shuffled poetries.

Classifier	n_f	Precision		Recall		Accuracy
		original	shuffled	original	shuffled	
LDA	14 ± 9	0.56 ± 0.04	0.56 ± 0.03	0.56 ± 0.05	0.55 ± 0.06	0.56 ± 0.03
RF	19 ± 9	0.62 ± 0.05	0.64 ± 0.05	0.67 ± 0.06	0.58 ± 0.08	0.63 ± 0.05
KNN	13 ± 9	0.60 ± 0.04	0.60 ± 0.04	0.61 ± 0.05	0.59 ± 0.07	0.60 ± 0.04
SVM	14 ± 9	0.56 ± 0.04	0.56 ± 0.04	0.57 ± 0.06	0.54 ± 0.08	0.56 ± 0.04
MLP	15 ± 10	0.62 ± 0.04	0.61 ± 0.04	0.60 ± 0.05	0.62 ± 0.07	0.61 ± 0.04

Table 4
Performance of classifiers in discriminating original and shuffled prose.

Classifier	n_f	Precision		Recall		Accuracy
		original	shuffled	original	shuffled	
LDA	13 ± 9	0.63 ± 0.04	0.72 ± 0.05	0.79 ± 0.05	0.53 ± 0.08	0.66 ± 0.04
RF	17 ± 10	0.64 ± 0.04	0.71 ± 0.04	0.76 ± 0.04	0.57 ± 0.07	0.67 ± 0.04
KNN	13 ± 10	0.74 ± 0.06	0.68 ± 0.04	0.74 ± 0.06	0.54 ± 0.09	0.64 ± 0.04
SVM	17 ± 9	0.63 ± 0.04	0.73 ± 0.04	0.81 ± 0.04	0.51 ± 0.09	0.66 ± 0.04
MLP	14 ± 10	0.62 ± 0.04	0.71 ± 0.05	0.78 ± 0.05	0.53 ± 0.08	0.65 ± 0.04

author often carefully chooses words to create rhymes, we believe that the observed low accuracy can be the result of rhymes randomly generated in the shuffled corpus.

In the majority of the selected features, there is a frequency peak close to zero that decreases in the shuffled version, meaning that the shuffled texts may have more rhymes than the original ones. The difference in the distributions, mainly due to the highest peak, promotes better classification performance. As a result, the overall discriminability metrics are higher than the comparison between original and shuffled poetry (see Fig. 9 and Table 3).

The comparison between original and shuffled proses is shown in Table 4. In this case, the classification accuracies are higher than the previous case, indicating that the discriminability between them is more evident than between original and shuffled poetries. It is worth noting that the best classifier, RF, obtained an accuracy of 0.67 ± 0.04.

To further investigate the role of the punctuation structure and the words chosen to compose the text, we also compared between shuffled versions of poetry and prose. It is interesting to see in Table 5 that shuffled poetry and proses, which have only their words shuffled but keep their punctuation in the same places as in the original versions, are not that well classified as the original versions (see Table 2). The best performance is also achieved with RF (accuracy of 0.67 ± 0.03). It is worth mentioning that, in this case, the classifier performance is more dependent on the employed classifier. Thus, the features could not discriminate the classes with the same quality as in the comparison between the original texts. This result emphasizes that both text structure and word choice are essential to convey the rhythm that characterizes poetry.

4. Conclusions

One of the several features shared by arts and science is their division into major areas or types of works. While in science one may categorize works into physical and biological sciences, a major division in literature concerns the concepts

Table 5
Performance of classifiers LDA, RF, KNN, SVM and MLP on the shuffled poetry (poetry*) and shuffled proses (prose*).

Classifier	n_f	Precision		Recall		Accuracy
		poetry*	prose*	poetry*	prose*	
LDA	11 ± 8	0.74 ± 0.06	0.62 ± 0.03	0.50 ± 0.08	0.81 ± 0.07	0.66 ± 0.03
RF	18 ± 10	0.74 ± 0.05	0.64 ± 0.03	0.53 ± 0.08	0.81 ± 0.06	0.67 ± 0.03
KNN	16 ± 12	0.69 ± 0.05	0.64 ± 0.04	0.59 ± 0.09	0.73 ± 0.07	0.66 ± 0.04
SVM	12 ± 10	0.75 ± 0.05	0.62 ± 0.03	0.48 ± 0.09	0.84 ± 0.06	0.66 ± 0.03
MLP	14 ± 10	0.70 ± 0.04	0.64 ± 0.04	0.56 ± 0.08	0.76 ± 0.06	0.66 ± 0.04

of prose and poetry. While these two important types of works can often be readily identified by humans, the automated classification of respective literary works constitutes a more substantial challenge. Though rhythm and rhymes are known to be elements typically found in poetry, they also appear to varying degrees in several works understood as prose. The present work aimed at developing a systematic approach to identifying – through concepts from network science, pattern recognition and feature selection – the characteristics that are particularly specific to poetry and prose.

Our main goal in this paper was the proposal of a method for feature extraction that could capture rhythmic patterns from prose and poetry so that these texts could be easily identified by well known classifiers. With this aim, we resorted to prose and poetry texts from the Gutenberg database. We represented the texts in terms of all the identified rhymes and the phonemes. These representations were characterized in terms of some proposed metrics, including the mean and coefficient of variation of the time intervals, which were then selected through *information gain* attribute selector. In order to test the potential of the features, we employed five different classifiers based on different assumptions. To analyze the results, we also considered a network science-based methodology.

As we developed our methodology, many interesting results were found. First, in the analysis of some basic statistics of the texts (e.g., text size and number of symbols), prose, and poetry were found to be similar. However, by considering the number of rhyme repetitions and the average rhyme repetitions, poetry tends to give rise to a larger diversity of rhymes and repetitions. In the following, by considering the features obtained from the proposed representation and the attribute selection method, the best accuracy result was found for the MLP classifier.

In order to better understand the relationship between the classes and the features, we represent the relationship between the samples as a complex network. More specifically, the network nodes and links relate to the texts and their feature similarity, respectively. By varying the number of considered features, it was possible to note that poetry rhyme patterns tended to be substantially more diversified than in prose. Even assuming that there is a fixed metric for many of the considered poetry, the result illustrates how diversely poetry can be written.

Interestingly, the comparison between poetry and shuffled poetry, prose and shuffled prose revealed that the task of classifying between poetry and shuffled poetry is not trivial, which agrees with the results obtained from the complex network analysis. In other words, the classification task is more challenging since there is a wide range of possibilities for poetry. The diversity of rhymes and repetitions could be related to human language characteristics [35,36]. It has been shown that rhyming leads to shorter semantic network distances, therefore facilitating the lexical processing. As a consequence of this effort minimization, diversified strategies for writing in rhymes could arise.

Many possible future works can be developed from the proposed representation and measurements. For instance, one can consider the analysis and comparison between texts of characteristics of literary movements. These features can also be used in more elaborated classification texts combined with other attributes (e.g., word counts). While in some networked text representations only topological properties of texts are considered, mainly reflecting syntax/style [37,38], one could enrich these representations with rhyme information in order to obtain an enhanced classification of texts.

CRediT authorship contribution statement

Henrique Ferraz de Arruda: Conceptualization, Methodology, Software, Validation, Visualization, Investigation, Writing – original draft. **Sandro Martinelli Reia:** Conceptualization, Methodology, Software, Validation, Visualization, Investigation, Writing – original draft. **Filipi Nascimento Silva:** Conceptualization, Methodology, Writing – review & editing. **Diego Raphael Amancio:** Conceptualization, Methodology, Writing – review & editing. **Luciano da Fontoura Costa:** Conceptualization, Methodology, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

H. F. de Arruda acknowledges FAPESP, Brazil for sponsorship (grants 2018/10489-0). S. M. Reia was partially supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. D. R. Amancio thanks CNPq, Brazil (grant no. 304026/2018-2) and FAPESP (grant no. 20/06271-0). L. da F. Costa thanks CNPq, Brazil (grant no. 307085/2018-0). This work has been supported also by the FAPESP, Brazil grant 15/22308-2. H. F. de Arruda thanks Soremartec S.A. and Soremartec Italia, Ferrero Group, for partial financial support (from 1st July 2021). His funders had no role in study design, data collection, and analysis, decision to publish, or manuscript preparation.

References

[1] H. Toivonen, et al., Computational creativity beyond machine learning, Phys. Life Rev. (2020).
[2] N. Jamal, M. Mohd, S.A. Noah, Poetry classification using support vector machines, J. Comput. Sci. 8 (9) (2012) 1441.
[3] A. Gopidi, A. Alam, Computational analysis of the historical changes in poetry and prose, in: Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change, 2019, pp. 14–22.
[4] O. Calin, Statistics and machine learning experiments in english and Romanian poetry, Sci (ISSN: 2413-4155) 2 (4) (2020) http://dx.doi.org/10.3390/sci2040092, URL https://www.mdpi.com/2413-4155/2/4/92.
[5] A. Tikhonov, I.P. Yamshchikov, Guess who? Multilingual approach for the automated generation of author-stylized poetry, in: 2018 IEEE Spoken Language Technology Workshop, SLT, IEEE, 2018, pp. 787–794.
[6] S. Talafha, B. Rekabdar, Poetry generation model via deep learning incorporating extended phonetic and semantic embeddings, in: 2021 IEEE 15th International Conference on Semantic Computing, ICSC, IEEE, 2021, pp. 48–55.
[7] A. Krishna, V.D. Sharma, B. Santra, A. Chakraborty, P. Satuluri, P. Goyal, Poetry to prose conversion in sanskrit as a linearisation task: A case for low-resource languages, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 1160–1166.
[8] L.F. Costa, H.F. Arruda, Syntonets: toward a harmony-inspired general model of complex networks, Eur. Phys. J. B 93 (12) (2020) 1–14.
[9] B. Hrushovski, The meaning of sound patterns in poetry: an interaction theory, Poetics Today 2 (1a) (1980) 39–56.
[10] S. Doumit, N. Marupaka, A.A. Minai, Thinking in prose and poetry: A semantic neural model, in: The 2013 International Joint Conference on Neural Networks, IJCNN, IEEE, 2013, pp. 1–8.
[11] Project gutenberg, 2021, https://www.gutenberg.org/. (Accessed 27 May 2021).
[12] Brown digital repository, 2021, https://repository.library.brown.edu/studio/. (Accessed 27 May 2021).
[13] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, " O'Reilly Media, Inc.", 2009.
[14] M.A. Mines, B.F. Hanson, J.E. Shoup, Frequency of occurrence of phonemes in conversational English, Lang. Speech 21 (3) (1978) 221–241.
[15] J.-H. Oh, K.-S. Choi, An ensemble of grapheme and phoneme for machine transliteration, in: International Conference on Natural Language Processing, Springer, 2005, pp. 450–461.
[16] P. Zegers, Speech recognition using neural networks, 1998, University of Arizona, Arizona.
[17] L.F. Costa, On sound synthesis IV: Rhythm and tempo (CDT-45), 2020, http://dx.doi.org/10.13140/RG.2.2.27563.05927/2.
[18] B. Azhagusundari, A.S. Thanamani, et al., Feature selection based on information gain, Int. J. Innov. Technol. Explor. Eng. (IJITEE) 2 (2) (2013) 18–21.
[19] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Phys. Rev. E 69 (6) (2004) 066138.
[20] D.R. Amancio, C.H. Comin, D. Casanova, G. Travieso, O.M. Bruno, F.A. Rodrigues, L. da Fontoura Costa, A systematic comparison of supervised classifiers, PLoS One 9 (4) (2014) e94137.
[21] J. Friedman, T. Hastie, R. Tibshirani, et al., The elements of statistical learning, in: Springer series in statistics New York, vol. 1, (no. 10) 2001.
[22] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[23] J.L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (9) (1975) 509–517.
[24] T.-F. Wu, C.-J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (Aug) (2004) 975–1005.
[25] G.E. Hinton, Connectionist learning procedures, in: Machine Learning, Elsevier, 1990, pp. 555–610.
[26] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[27] H.F. de Arruda, A. Benatti, C.H. Comin, L.d.F. Costa, Learning deep learning (CDT-15), 2019.
[28] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai, vol. 14, no. 2, Montreal, Canada, 1995, pp. 1137–1145.
[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[30] C.H. Comin, T. Peron, F.N. Silva, D.R. Amancio, F.A. Rodrigues, L.d.F. Costa, Complex systems: Features, similarity and connectivity, Phys. Rep. 861 (2020) 1–41.
[31] G.K. Gupta, Introduction to Data Mining with Case Studies, PHI Learning Pvt. Ltd., 2014.
[32] T.M. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement, Softw. - Pract. Exp. 21 (11) (1991) 1129–1164.
[33] F.N. Silva, D.R. Amancio, M. Bardosova, L. da F. Costa, O.N. Oliveira Jr., Using network science and text analytics to produce surveys in a scientific topic, J. Inf. 10 (2) (2016) 487–502.
[34] H. Rinne, The Weibull Distribution: A Handbook, CRC Press, 2008.
[35] Y.N. Kenett, E. Levi, D. Anaki, M. Faust, The semantic distance task: Quantifying semantic distance with semantic network path length., J. Exp. Psychol: Learn. Mem. Cogn. 43 (9) (2017) 1470.
[36] M. Stella, Cohort and rhyme priming emerge from the multiplex network structure of the mental lexicon, Complexity 2018 (2018).
[37] D.R. Amancio, A complex network approach to stylometry, PLoS One 10 (8) (2015) e0136076.
[38] D.R. Amancio, Probing the topological properties of complex networks modeling short written texts, PLoS One 10 (2) (2015) e0118394.