Received: 15 January 2021 | Revised: 16 May 2021 | Accepted: 26 May 2021

DOI: 10.1002/asi.24532

RESEARCH ARTICLE

JASIST WILEY

A bridge too far for artificial intelligence?: Automatic classification of stanzas in Spanish poetry

Álvaro Pérez Pozo[1] | Javier de la Rosa[1] | Salvador Ros[1] | Elena González-Blanco[2] | Laura Hernández[1] | Mirella de Sisto[1]

[1]Universidad Nacional de Educación a Distancia, Madrid, Spain

[2]IE University, Madrid, Spain

Correspondence
Álvaro Pérez Pozo, Universidad Nacional de Educación a Distancia, Madrid, Spain.
Email: alvaro.perez@scc.uned.es

Funding information
H2020 European Research Council, Grant/Award Number: ERC-2015-STG-679528

Abstract

The rise in artificial intelligence and natural language processing techniques has increased considerably in the last few decades. Historically, the focus has been primarily on texts expressed in prose form, leaving mostly aside figurative or poetic expressions of language due to their rich semantics and syntactic complexity. The creation and analysis of poetry have been commonly carried out by hand, with a few computer-assisted approaches. In the Spanish context, the promise of machine learning is starting to pan out in specific tasks such as metrical annotation and syllabification. However, there is a task that remains unexplored and underdeveloped: stanza classification. This classification of the inner structures of verses in which a poem is built upon is an especially relevant task for poetry studies since it complements the structural information of a poem. In this work, we analyzed different computational approaches to stanza classification in the Spanish poetic tradition. These approaches show that this task continues to be hard for computers systems, both based on classical machine learning approaches as well as statistical language models and cannot compete with traditional computational paradigms based on the knowledge of experts.

1 | INTRODUCTION

Traditionally, the analysis of poetry has been carried out by hand, involving the valuable time and knowledge of human experts, and posing a serious limitation when scaling up the corpora. A barebone analysis of poetry generally covers the study of verses, rhyme, rhythm, stanza, and the rhetorical devices employed by the poet. Recent advances in natural language processing (NLP) have attracted many literary scholars on the promise of easing the burden of the manual work. In this regard, new avenues of research in poetry are now possible, allowing studies that were unfeasible to afford before, and presenting the necessary means for validating both previous analysis and the general assumptions of the more traditional scholarship methods. The possibility of processing semi-automatically big corpora enables the paradigm of distant reading (Moretti, 2013), an approach in literary studies that applies computational methods to significant collections of literary data, at the time that it complements traditional literary analysis techniques such as close reading, the in-depth analysis of how a literary text works (Fisher & Frey, 2013), as well as general assumptions in traditional scholarship.

An essential aspect of the analysis of poetry relies on extracting information from the different structures found in a poem (i.e., verse, stanza, or poem). To a certain degree, identifying these structures automatically with the

help of a computer is partially possible now, and such tools already exist for many languages such as Spanish (Gervás, 2000), Portuguese (Araújo & Mamede, 2002), French (McAleese, 2007), English (Heuser & Antiila, 2010), Czech (Ibrahim & Plecháč, 2011), Euskera (Agirrezabal, 2016), and Dutch (De Sisto, 2020). Except for Araújo and Mamede's work (Araújo & Mamede, 2002), which deals with stanza classification for the Portuguese language, the rest of these works focus on automatic scansion, that is, verse length and rhyme. Most approaches (De La Rosa, 2020; Gervás, 2000; Heuser & Antiila, 2010; Ibrahim & Plecháč, 2011; McAleese, 2007) use rule-based systems, while only two of them make use of neural networks techniques (Agirrezabal, 2016; De Sisto, 2020).

Remarkably, the automatic classification of the structure formed by the verses within a poem, that is, the stanzas, remains understudied. A stanza is the minimal structural unit of a poem that usually encapsulates themes or ideas (Kirszner, 2013). The detection and classification of the different kinds of stanzas present in a poem is an especially relevant task for poetry studies since it complements the metrical information of a poem. In this work, we frame this problem as a classification task. We implemented different approaches ranging from traditional computational methods to artificial intelligence (AI)-based solutions (Onan, Korukoglu, & Bulut, 2016; Onan, Korukoğlu, & Bulut, 2016). Given the good results that modern NLP solutions based on neural networks (e.g., embeddings, deep learning) obtain in text classification tasks in different domains as feature selection, (Onan & Korukoğlu, 2017), sentiment analysis (Onan, 2020b), or mining opinion (Onan, 2020a), we expected the IA approaches in this study to perform better than the classical algorithms. However, contrary to our belief, the results show that a traditional knowledge base expert system produces the best results, albeit not being especially accurate. Counting syllables and verses might seem like such a trivial task; still, these results suggest there is an inherent complexity difficult to capture by most approaches.

The rest of the paper is organized as follow. Section 2 introduces the stanza classification problem. Section 3 develops and shows the different techniques used to classify stanzas and the evaluation results. Section 4 shows the discussion of results and section 5 outlines the principal conclusions.

2 | TASK DESCRIPTION: CLASSIFICATION OF STANZAS

Stanzas are structural units formed by verses, and therefore they are related to the author style and even historical preferences. We can consider them as expressive elements of a poem (Jauralde, 2020). There are a great abundance and variety of stanzas in the Spanish tradition. For this reason, identifying them is a complex task that requires a vast knowledge of the poetry domain. Three aspects determine how a stanza is identified in the Spanish tradition: verse length, rhyme type, and rhyme pattern (Domínguez Caparrós, 2014; Jauralde, 2020; Quilis, 2000; Torre, 2000). Therefore, stanza classification is a problem that can be formulated in three stages (Domínguez Caparrós, 2014):

1. Calculation of verse lengths and verse length pattern.
2. Determining the rhyme type.
3. Extraction of the rhyme pattern.

2.1 | Verse length and stanza length pattern

To calculate the length of a verse, scholars have to count the number of syllables in the verse. However, this is not a simple task in Spanish, since once you have calculated the number of syllables of a verse, it can be affected by some literary devices. One common device in the synaloepha, which joins the last syllable of a word with the first one of the next word based on vocalic sounds (e.g., "vie-jo (2) a-mi-go (3)" vs. "vie-joa-mi-go (4)"). Similarly, syneresis is the union of syllables within the same word (e.g., "Me-de-a (3)" vs. "Me-dea (2)").

It is also possible to find other devices when the separation of vowel groups that would normally go together is considered. This is the case of dialepha, a device that is the opposite of synaloepha and splits the syllables that are part of said device. Alternatively, we can also find another device called "dieresis," when diphthongs are broken for stylistic reasons. It is usually indicated by using the diacritical sign "umlaut" (¨) to indicate the break. For example, "rüido"/"rü-i-do" vs. "ruido"/"rui-do." For the sake of clarity, in this article we will use this diacritical mark to indicate the breaking of diphthongs.

Considering these features, it is possible to obtain the different lengths that a verse could have. If, for example, a verse can contain n literary devices that shorten the syllable length, the number of possible syllable divisions is 2^n. As shown in Table 1, the original verse could have two synaloephas, so there are four possible verse lengths. The application of these literary devices is at the discretion of the author, so even when the most common situation is that all of them were applied, it is difficult to determine the precise length of

TABLE 1 Verse length and literary devices

Original verse												
Pongo estos versos en mi botella al mar **(I put these verses in my bottle to the sea)**												
Length according to orthographic separation												
1	2	3	4	5	6	7	8	9	10	11	12	13
Pon	go	es	tos	ver	sos	en	mi	bo	te	lla	al	mar
Metrical lengths for $n = 2$												
Pon	goes	tos	ver	sos	en	mi	bo	te	llaal	mar		
Pon	goes	tos	ver	sos	en	mi	bo	te	lla	al	mar	
Pon	go	es	tos	ver	sos	en	mi	bo	te	llaal	mar	
Pon	go	es	tos	ver	sos	en	mi	bo	te	lla	al	mar

TABLE 2 Consonant/assonant rhyme examples

Stanza	**Consonant rhyme**
Bravo león, mi coraz**ón**	-ón
Tiene apetitos, no raz**ón**	-ón
	Author: Alfonsina Storni
	Assonant rhyme
Ante una vidriera r**ot**a	-ó-a
Coso mi lírica r**op**a	-ó-a
	Author: Federico García Lorca

TABLE 3 Rhyme pattern example

Stanza	**Rhyme pattern**
Escribí en el aren**a**l	(a)
los tres nombres de la vida:	(−)
vida, muerte, am**o**r.	(b)
Una ráfaga de m**a**r,	(a)
tantas claras veces d**a**,	(a)
vino y nos borr**ó**.	(b)
	Author Miguel Hernández

a verse only with the text of the poem, since the use of these resources is detected when the poem is recited, thus making the textual form of a poem a lossy version of that intended by the poet.

Nonetheless, the length of one single verse is not enough to fix the type of stanza. In a stanza, verse lengths tend to have a pattern, so all verse lengths are needed to build the stanza length pattern.

2.2 | Rhyme type

There are two types of rhymes in Spanish poetry: assonant and consonant. The starting point for detecting assonance or consonance is the last stressed vowel of the last word. If all letters starting from this vowel coincide along with two or more verses, a consonant rhyme occurs. Alternatively, if only the vowels coincide, the rhyme becomes assonant. Furthermore, a consonant rhyme included assonant rhyme by definition, but the consonance supersedes the assonance. Table 2 shows an example of consonant rhyme where all the letters coincide from the last stressed vowel and an example of assonant rhyme where only the last vowels that follow the last stressed vowel of the verse are the same (Domínguez Caparrós, 2014).

2.3 | Rhyme pattern identification

Finally, it is necessary to identify the rhyming pattern of a stanza. Most stanza types are defined by one or several rhyme patterns. Depending on the rhyme type detected, researchers have to focus on all vowels from the last stressed vowel of the ending word of each verse when the type of rhyme is assonant or on all the letters from the last stressed vowel of the ending word of each verse in a consonant rhyme type. The rhyme pattern is represented as a string that matches rhyming verses. For each verse in the stanza, a letter of the alphabet is assigned, associating the same letters to the verses that rhyme with each other (i.e., same vowel or letters), and marking with a hyphen the verses that are left without a rhyme (see Table 3).

Once these three aspects are identified (verse lengths, rhyme types, and rhyme patterns), the expert has to match their characteristics against the different definitions of stanzas given by the poetic tradition and make a decision about the best match. Therefore, any automatic stanza classification tool needs at least this information, either as input or encoded in a representation of the text, to decide on the stanza type of a stanza. These tasks can be implemented using different computational paradigms. In this study, we offer three different implementations for stanza classification.

TABLE 4 Rule-based algorithm

```
Input: stanza
Output: A list of stanzas candidates' types
Stanza Classification (stanza)
    Calculate numberofverses
    Calculate stanza_verse_length_pattern[i](stanza)
    Calculate rhyme_pattern(stanza)
    Calculate rhyme_type(stanza)
    While stanza_verse_length_pattern[i]
Result = Check stanza_rules (stanza_verse_length_pattern[i],
  rhyme_type, rhyme_pattern)
Candidate_stanza = Candidate_stanza + Result
End
Return candidate_stanza
```

3 | MODELS DESCRIPTION AND EVALUATION

In this work, stanza classification in the Spanish poetic tradition task is afforded using different computational approaches. Therefore, we apply traditional computational paradigms based on the knowledge of experts and causal reasoning, (i.e., ruled-based) following with new approaches sit on classical machine learning approaches as well as statistical language models (i.e., neural networks approach).

3.1 | Ruled-based approaches

3.1.1 | Baseline

The first approach is a classical solution based on extracting the information needed and then composing a knowledge base curated by experts with the proper rules that identify the different stanza types. Verse length patterns, rhyme patterns, and rhyme types were extracted using the automatic Spanish scansion tool Rantanplan by De La Rosa (2020). This tool claims an accuracy of 96.23% when extracting metrical patterns and its related information. After the information extraction step, the algorithm of a basic expert system is outlined in Table 4. In this approach, the algorithm analyzes the stanza to gather information about the verse lengths, its rhyme pattern and its rhyme type. With this information, it will check the rules obtained from the knowledge of experts and their priority and will propose a stanza.

The system must deal with different situations that might impact the final result. For example, verse lengths may vary depending on the application of the literary devices aforementioned, which is represented in the algorithm as a degree of uncertainty in the predictions. When applied to a verse, this uncertainty is modeled as the range of possible verse lengths obtained when applying all possible rhetorical devices (upper bound) and none of them (lower bound) (e.g., in Table 1 the verse interval length is [11, 13]). This degree of uncertainty has to be accounted for in all the steps of the stanza classification process, thus producing different verse length patterns when the devices are present. In Table 5, the uncertainty intervals for each verse are [8, 10] for the first one, and [8, 9] for the second one. With this uncertainty the stanza could be both a "pareado de arte menor" (i.e., all verse lengths lower than 9) or a "pareado de arte mayor" (i.e., all verse lengths greater than 8).

TABLE 5 Stanza classification uncertainty

Stanza	Uncertainty interval
Verse 1: Yo soy parael secreto	[8, 10]
Verse 2: lo mismo que es el abeto	[8, 9]
	Author: Federico García Lorca

Another important aspect to consider is related to the construction of the rhyme pattern itself. Many verses may share the same proposed letter regardless of the position in the poem they appear, thus forming rhyming pairs. In practice, it is rare to find rhyming pairs separated apart by more than a few verses since the authors play with the rhyme patterns to build musical effect with the patterns. If the verses with the same rhyme pair are too far apart, it seems reasonable to assume that the author did not want them to be considered a rhyme pair. To deal with this situation, verses that are too far from each other are not considered as a rhyming pair even if their endings match. The distance between these verses is somewhat arbitrary since it is usually the author who "feels" when verses are too distant from each other for them to be considered a rhyming pair. The tests and manual revision of stanzas suggested that separation of four verses is optimal to detect verse with the same endings.

Finally, the algorithm must deal with rhyme type detection, which poses some challenges since the rhymes are expected to have a sound effect. Therefore, the pronunciation of the words may influence rhyme detection. Two special cases are considered: homophonic groups and diphthongs. The homophonic groups have different spellings but are pronounced the same way. In the case of diphthongs, the difficulty lies in the selection of the rhyming vowel.

To alleviate these issues, a set of relaxation rules is applied, allowing the replacement of certain phonetical

TABLE 6 Sexta Rima stanza definition

Sexta Rima stanza
Type of rhyme: Consonant rhyme Rhyme pattern: "ababcc" or "aacbbc," Verse length pattern [11, 11, 11, 11, 11, 11]

groups by their equivalent ones for the purpose of the calculation of the rhyme. In the case of homophonic groups, the rules are specific for the consonant rhyme. In consonant rhyme, all the letters from the last stressed vowel of the ending word of each verse have to match. However, for rhyming purposes, letters than sound the same should also match. These homophonic groups in Spanish are v/b, zi/ci, qui/ki, que/ke, gi/ji, ge/je, ce/ze and ll and y (syllabic). If a diphthong is found on any of the last syllables, the general rule for detecting rhyme is to only consider the strong vowels (a, e, o) and leave out the weak ones (e.g., i, u) (Quilis, 2000). But if there is a diphthong with two weak vowels, only the first one is considered (e.g., "cementerio," "cementero").

Once the algorithm has extracted all the information, it must check the rules to obtain the stanza type candidates. Each stanza is defined according to the three properties discussed (see Table 6), which are encoded as rules using regular expressions as pattern matching mechanism.

In the corpus, only the most frequent and prominent named stanza types in the Spanish tradition were included. Moreover, stanzas of more than 14 verses become were not include since they start to lose their expressive power and it is usually preferable to break them down using multi-stanza structures (e.g., soneto) (Jauralde, 2020). Thus, the catalogue includes 46 stanza types based on Spanish meter (Domínguez Caparrós, 2014; Jauralde, 2020; Quilis, 2000; Torre, 2000).

To optimize the matching process, the stanza catalogue was structured in four groups of stanzas types based on their pattern lengths: structured, semi-structured, *arte mayor* ("major art," longer than eight syllables) and *arte menor* ("minor art," shorter than eight syllables).

1. A structured length pattern is defined when each verse length is fixed (see Table 7) (e.g., *Manriqueña stanza* is defined as a stanza with the pattern: [8, 8, 4, 8, 8, 4]).
2. A stanza is defined as semi-structured because part of its verse length pattern is known (see Table 8) (e.g., Cuarteto Lira, is a stanza with verses of 11 and 7 syllables in any position).

TABLE 7 Manriqueña stanza

Stanza	Length
Vir-gen-del-cie-lo-re-í-na,	8
e-del-mun-do-me-le-ci-na,	8
quié-ras-meo-ír,	4
que-de-tus-go-zos-a-i-na	8
es-cri-ba-yo-pro-sa-dig-na	8
por-te-ser-vir.	4
	Author: Jorge Manrique

TABLE 8 Cuarteto Lira

Stanza	Length
¡Cuán-so-li-ta-ria-la-na-ción-queun-dí-a	11
po-bla-rain-men-sa-gen-te,	7
la-na-ción-cu-yoim-pe-rio-seex-ten-dí-a	11
del-O-ca-soal-O-rien-te!	7
	Author: José de Espronceda

Finally, in some stanza types, the verse length does not have a fixed value but a maximum or minimum value for all the metrical lengths in a stanza.

3. A stanza is defined as *Arte Mayor* because all its verses are greater or equal than nine syllables (Domínguez Caparrós, 2014) (see Table 9).
4. A stanza is defined as *Arte Menor* because all its verses are of less than nine syllables (Domínguez Caparrós, 2014) (see Table 10).

Therefore, the algorithm first will check the structured patterns looking for possible matches in the set of stanza verse length patterns. If it does not match will continue through the rest of groups of stanzas until it succeeds or finishes. This process must be carried out for all the possible combinations obtained by the intervals of verse lengths. The result could be then more than one kind of stanza. This uncertainty is even complicated for scholars since as we said above the decision of used the literary devices depend on the original author.

3.1.2 | Evaluation

In the evaluation process of the model, an annotated corpus of 5,005 stanzas was built (https://github.com/linhd-postdata/stanza-detection-evaluation). The corpus is composed of stanzas ranging from the early 15th century

TABLE 9 Arte Mayor stanza

Stanza	Arte Mayor
Y-de-pron-toen-ho-rren-does-tam-pi-do	10
Des-qui-ciar-se-laes-tan-cia-sin-tió,	10
Yal-tre-men-do-tar-tá-reo-rü-i-do	10
Cien-es-pec-tros-al-zar-se-mi-ró.	10
	Author G. A. Bécquer

TABLE 10 Arte Menor stanza

Stanza	Arte Menor
Es-cri-bíen-el-a-re-nal	8
Los-tres-nom-bres-de-la-vi-da:	8
vi-da-mü-er-te-a-mor.	8
U-na-rá-fa-ga-de-mar,	8
tan-tas-cla-ras-ve-ces-da,	8
vi-no-y-nos-bo-rró.	8
	Author: Miguel Hernández

to contemporary poems and was manually reviewed by three experts to ensure its quality. The manually annotated stanza dataset was compared with the output given by the system. In the end, accuracy is computed (see Table 11). The results and code to replicate the evaluation are available in the repository (Pérez, 2020).

Despite all the rules extracted from the handbooks and all the recommendations given by the experts, the accuracy of the system is improved still further what gives an idea of the complexity of the problem. Based on these results, it is easy to infer that the heterogeneity of interpretations of the three features opens a wide range of possibilities that the algorithm has to consider. This characteristic makes that the solution could not be robust enough since the final results depend on parameters difficult of implementing computationally and therefore, it would of interest to find a solution more general and at least with the same accuracy.

3.2 | Decision trees and rule sets

A different approach using machine learning is to consider this problem a pure multiclass classification task, where each sample is the text of a stanza and the model must predict what kind of stanza type is the text structured like. In this setting, and based on the knowledge base we already defined, decision trees come as a natural

TABLE 11 Evaluation results

Accuracy (%)
78.63

TABLE 12 Decision tree and decision rule set evaluation results expressed in accuracy percentages

Method	Accuracy@1 (%)	Accuracy@3 (%)
Decision Tree	88.21	N/A
Random Forests	88.51	N/A
Decision Rule Set	81.11	79.22

fit to the posed problem for their capacity to encode rules and their priorities, and their ability to produce explainable and interpretable models (see Safavian & Landgrebe, 1991 for a survey on the topic). To carry out the training, we split the 5,005 stanzas in the corpus into stratified training and test sets of 4,004 stanzas (80%) and 1,001 stanzas (20%), respectively. However, a common issue with two methods (i.e., Decision Tree and Random Forest), in general, is their inability to generalize well and their tendency to overfit on the training sets. Table 12 shows the results obtained by decision tree and random forest classifiers on the test set, which improve the knowledge base baseline by 10%. Reducing the number of leaves and general complexity of the decision tree or changing the number of estimators in the random forest method did not improve the accuracy.

One approach to mitigate the generalization problem of decision trees while maintaining the interpretability of the results is to use associative classification methods. Bayesian rule sets and decision rule sets fit somewhat in this category with the added advantage of not greedily growing the model (Domingos, 2000; Wang et al., 2017). For this experiment, rules are extracted for each stanza type from an ensemble of trees, and a weighted combination of these rules was built by solving an L1-regularized optimization problem over the weights. Since this approach works well for binary classification problems, the best set of rules for each stanza type is generated and the results are evaluated as a multilabel multiclass problem. Each stanza was vectorized and transformed into a feature vector binary vector containing information on its assonance or consonance rhyme as well as which of the regular expressions in the knowledge-based were triggered on its rhyme pattern (Table 12). The accuracy of the overall model at first prediction (accuracy of the first prediction) and the accuracy of the model when predicting more than one stanza type (accuracy of the first three predictions) is also reported.

3.3 | Neural networks

Finally, solutions based on neural networks are explored. Neural networks have the advantage of capturing knowledge about datasets without having to specify the rules the govern them. When applied to the stanza classification problem, the aim was that the networks would learn most of the rules necessary to classify the different stanza types without encoding the proposed rules crafted by the experts. In this scenario, classifying stanzas is again a classical classification problem where the stanzas are the inputs and need to be classified as one of the 46 proposed stanza types in the annotated corpus. Since the models relied on word embeddings and language models to extract feature vectors, it was not necessary to calculate any of the previous features of a stanza (i.e., stanza length pattern, rhyme, rhyme pattern). Specifically, it is considered as a multiclass classification problem (Zhou et al., 2016) and used a modification of the three-layer model proposed by Basaldella et al. (2018).

The first step in this kind of architectures is the extraction of the semantic information of the text in the stanza using a layer of embeddings. To build this layer, a fine-tuning approach is normally used in which knowledge acquired in previous tasks is used to solve new ones, thus reducing the number of weights the neural network must learn while enabling the use of much smaller datasets. Embeddings have demonstrated to be highly effective in a number of NLP tasks since they represent textual inputs (words, sentences, or entire passages) as dense vectors that encode structural and semantic information (Chen et al., 2013). In the stanza classification task, the embedding layer works as a look up table where each row represents the embedding of a word, and the concatenation of these dense vectors represent the text of a stanza. The two most important parameters of the embedding layer are the number of unique words used as vocabulary, and the number of dimensions for representing a word. There are several embedding representation techniques. The ones most commonly used due to its good performance and accuracy are word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and BERT (Devlin et al., 2019). In this work, the models are limited to use GloVe and BERT embeddings.

The output of this first layer is fed into a second layer with three stacked bi-directional LSTM layers. The LSTM layers can preserve data long-term memory, which of special interest in NLP, time series and sequential tasks (e.g., predicting information in a context). Since word embeddings encode semantic information of the words of a stanza, Bi-LSTM layer is used to reinforce the structural knowledge that characterizes a stanza (e.g., verses patterns, rhyme patterns). Finally, a fully connected linear layer is used to reduce the dimensionality of the second layer output to 46 dimensions, one for each of the 46 stanza types. As recommended in the literature, the parameters of the network were tuned to better perform when classifying text (Probst et al., 2018): learning rate set to 3e-05, size of the dimension of the Bi-LSTM layer was set to 100, the dropout rate of the LSTM layers was 0.1, number of training epochs was 3, and number of dense layers was 1. For this process a cross-validation procedure and state the art information has been used (Bischl et al., 2012; Devlin et al., 2019; Ge et al., 2019; Guyon et al., 2010) (see Table 13).

3.3.1 | Evaluation

For evaluation purposes, the model was trained using the annotated stanza corpus. The corpus was split into three sub-sets for testing, training, and validation tasks. First, the rule 80/20 was applied, splitting the corpus into two subsets for training and testing (i.e., 80% training, 20% testing). In the training set was applied the same 80/20 building the final training and evaluation sets. During the whole process, data were split in a stratified fashion according to the percentage of examples of each stanza in the corpus. In Table 13, the results obtained for the two models (i.e., BERT and GloVe) are presented.

4 | DISCUSSION

On the bases of the evaluation results, Table 14, although the stanza classification may look simple on paper, it is in fact extremely complicated to solve. The first approach, a classical solution based on extracting the information needed and then composing a knowledge base curated by experts with the proper rules that identify the different stanza types obtains an accuracy of 78.63%, a result far away of what would be desirable for an approach that uses a scansion tool of 96.23% of accuracy in the tasks involved in the classification of stanzas. Despite all the rules extracted from the handbooks and all the recommendations given by the experts, the accuracy of the system is improved still further. This situation has at least two explanations: not all the expert knowledge is feasible to translate to machine language, so the computer systems are not able to reproduce it and the knowledge needed is extent enough that it is impossible to acquire and apply it in a computer system.

Starting with the impossibility of applying all the knowledge in a computer system, the problem of the use of different variants of a language is founded. For

TABLE 13 Model hyperparameters and evaluation result (GLOVE and BERT embeddings)

Embeddings	Learning rate	Epochs	LSTM layers	LSTM dropout	Dense layers	Accuracy (%)
GloVe	0.000125	325	3	0.2	1	66.72
BERT	0.0001	5	3	0.2	2	42.12

TABLE 14 Summary of evaluation results

Computational approach	Accuracy (%)
Ruled-based	78.63
Decision tree	88.21
Random forest	88.51
Bayesian rules	81.11
GLOVE	67.72
BERT	42.12

TABLE 15 Different Spanish variants

Old Spanish poem (original)	
Querié, peroque malo, bien a Sancta María,	-ía
udié los sus miráculos, dávais acogía;	-ía
saludávala siempre, diciela cada día:	-ía
"Ave gratïa plena, que pariste a Messía."	-ía
Contemporary Spanish modernization	
Quería, aunque era malo, mucho a Santa María,	-ía
oía sus milagros, siempre los acogía.	-ía
La saludaba siempre diciendo cada día:	-ía
"Ave, llena de gracia que pariste al Mesías"	-ías

TABLE 16 Relaxation in the verse length of a soleá stanza

Stanza	Length
E-se-tu-Nar-ci-so	6 (8)
Ya-no -se-veen-el-es-pe-jo	8
Por-quees-el-es-pe-jo-mis-mo.	8
	Author: Antonio Machado

example, when the system is facing poems in Old Spanish, the scansion tool has to consider these variants. Of course, this situation is unfordable computationally for all the existing variants. In the case of Spanish, it is possible to find errors when the spelling of an Old Spanish word differs greatly from contemporary spelling, as in the case of the word "ojos," which in Old Spanish would be "oios," as found on the "Cantar del mío Cid": "*De los sos oios tan fuertemientre llorando*." Because it does not distinguish between i and j, the model would detect a diphthong, which directly affects the metric.

To solve this kind of problems, it would be necessary to create a syllabifier for each variant of the language or to use modernized versions. However, this last option will pose new problems. Many times, words are lost or no longer exist, presented archaic forms where a word of two syllables becomes one of three (e.g., "u-dié" (2) vs. "o-í-a" (3)), or cause the consonant rhyme to be lost as in Table 15, wherein the original version a consonant rhyme in -ía is found, but when modernized this rhyme is lost in the last verse because the contemporary form of "messia" has a final "-s" ("mesías" in Spanish) that transforms the rhyme of this verse from consonant to assonant.

Regarding the impossibility of building an algorithm to solve some problems, two examples are found. First, the relaxation in the rules related to verse length described in the poetry handbooks allowing a small fluctuation in the fixed length of verses. This situation is shown in Table 16 where a popular stanza called *soleá* allows a length verse of six when it is defined as a stanza where all the verses have a metrical length of eight syllables.

Second, the treatment of verses formed by hemistichs. In poetry a hemistich is a half-verse of a verse, that is followed and preceded by a caesura (a small pause), that makes up a single verse. One feature of hemistichs in Spanish poetry is that the metric of each one of them is affected by the stress of the last word of the hemistich (Quilis, 2000) and thus the final length of the two hemistichs is different than the length of the verse as a whole. This situation depends on the realization of the reader or author (see Table 17).

If the last verse is analyzed isolated from the rest of the stanza, it has a metrical length of 15 (e.g., en-la-ci-ma-deun-á-la-mo-so-llo-za-ba-un-jil-gue-ro), but this stanza is a *cuaderna vía*, and it is defined as having all verses with a length of 14 syllables. If the hemistichs are considered (i.e., according to Domínguez Caparrós, 2014), hemistichs usually appear if the verse length is 14 syllables), the first one has a metrical length of seven syllables since ends with a proparoxytone word and the second has a length of seven syllables, making the total length of the verse 14 syllables.

The results for the second model (decision-rules) show that decision rule sets perform slightly below baseline while generating rules automatically that both learn the right priority of application and the necessary

TABLE 17 Stanza with hemistichs

Stanza ("/" are hemistiches)
Con sayal de amarguras, / de la vida romero,
topé, tras luenga andanza, / con la paz de un sendero.
Fenecía del día / el resplandor postrero.
En la cima de un álamo / sollozaba un jilguero.
Author: Ramón Pérez de Ayala

conditions to perform their matches. Random forests produce the best accuracy results, but the gain of 0.3% over traditional decision trees is hardly justifiable as decision trees are fully explainable.

The third model is based on neural networks. The election of this approximation was based on the main characteristic of this kind of supervised model: does not need any more knowledge than knowing what kind of stanza is and its capacity of generalization. From the results, it is clear that the structural and semantic information extracted from the embedding layer and by the Bi-LSTM does not seem to be enough to solve the problem with some precision so on the contrary to what may expect the traditional approaches based on expert knowledge obtained better results. These results could be improved if the corpus were of a larger size, but this is very difficult to achieve since it is a very technical corpus with a limited construction base compared to the millions of lines of text produced today. Another cause of this low performance can be related to how the stanza is coded at the input. In these models an embeddings layer is used, and it seems that the structural information it carries is not suitable for this task, and the semantic information does not improve its performance. Similarly, the Bi-LSTM layer does not seem to be able to capture the structural information needed to solve the task. Therefore, it seems that this task is certainly a bridge too far for artificial intelligence at this time and more research on the models and ways to encode the verses and their structural information.

5 | CONCLUSIONS

In this work, the stanza classification problem has been posed. To solve the problem, three different approaches have been implemented. First, an expert system based on rules and expert knowledge was modeled. This model considers the three main features required for stanza detection (verses length, rhyme pattern, and rhyme type) for Spanish poetry. The results showed that this approach has two main problems: not all the expert knowledge is feasible to translate to machine language. The knowledge needed is extent enough that it is impossible to acquire and apply it in a computer system. Besides these problems, it is reported accuracy of 78.76%. The second kinds of models based on decision trees and rule set had a better behavior (i.e., accuracy range from 81.11 to 88.51%) but still has a lack of generalization and depends on the expert's knowledge. Finally, neural network models have been implemented, expecting that these systems' properties were an advantage to solve the problem.

Contrary to what expected, these systems obtained the worse accuracy (i.e., 42.12 and 67.72%). These results confirm that the stanza classification problem is complex and nontrivial and that neural network methods are far away to be applied in this context, so more experiments and more research is needed, especially in the way to code the input of the networks since the semantic information of the embeddings looks not rich enough to help in the solution. In the same way, the capacity to remember the information of the Bi-LSTM neurons looks not enough to reinforce the structural information of the stanza.

ACKNOWLEDGMENTS

This work was supported by Starting Grant research project "Poetry Standardization and Linked Open Data: POSTDATA" (ERC-2015-STG-679528), funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program. All the source code and corpus are available at GitHub repository: https://github.com/linhd-postdata/stanza-detection-evaluation.

ORCID

Álvaro Pérez Pozo https://orcid.org/0000-0001-5897-1246

Salvador Ros https://orcid.org/0000-0001-6330-4958

REFERENCES

Agirrezabal, M. (2016). ZeuScansion: A tool for scansion of English poetry. *Journal of Language Modelling*, *4*(1), 3. https://doi.org/10.15398/jlm.v4i1.102

Araújo P., & Mamede N. (2002). Classificador de Poemas. In *CCTE conference, Lisbon, Portugal*. CCTE.

Basaldella, M., Antolli, E., Serra, G., & Tasso, C. (2018). Bidirectional LSTM recurrent neural network for Keyphrase extraction. In G. Serra & C. Tasso (Eds.), *Digital libraries and multimedia archives* (pp. 180–187). Springer. https://doi.org/10.1007/978-3-319-73165-0_18

Bischl, B., Mersmann, O., Trautmann, H., & Preuß, M. (2012). Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In *Proceedings of the fourteenth international conference on genetic and evolutionary computation conference – GECCO '12* (p. 313). ACM. https://doi.org/10.1145/2330163.2330209

Chen, Y., Perozzi, B., Al-Rfou, R., & Skiena, S. (2013). The expressive power of word embeddings. ArXiv:1301.3226 [Cs, Stat]

De La Rosa, J. (2020). Rantanplan: Fast and accurate syllabification and scansion of Spanish poetry. In *SEPLN annual congress, virtual conference*. SEPLN. https://doi.org/10.5281/zenodo.4301485

De Sisto, M. (2020). The interaction between phonology and metre. In *Approaches to romance and West-Germanic metre*. LOT.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. ArXiv:1810.04805 [Cs].

Domingos, P. (2000). Bayesian averaging of classifiers and the overfitting problem. In *Proceedings of the seventeenth international conference on machine learning* (pp. 223–230). ACM.

Domínguez Caparrós, J. (2014). *Métrica española*. Editorial UNED.

Fisher D., & Frey N. (2013). Close reading as part of a comprehensive literacy framework. *Colorado Reading Journal*, *24*, 30–34.

Ge, R., Kakade, S. M., Kidambi, R., & Netrapalli, P. (2019). The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. ArXiv:1904.12838

Gervás, P. (2000). A logic programming application for the analysis of Spanish verse. In J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.-K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, & P. J. Stuckey (Eds.), *Computational logic—CL 2000* (Vol. 1861, pp. 1330–1344). Springer. https://doi.org/10.1007/3-540-44957-4_89

Guyon, I., Saffari, A., Dror, G., & Cawley, G. (2010). Model selection: Beyond the Bayesian/frequentist divide. *Journal of Machine Learning Research*, *11*, 61–87.

Heuser, R. & Anttila, A. (2010). Prosodic. Retrieved from http://prosodic.stanford.edu

Ibrahim, R., & Plecháč, P. (2011). Towards the automatic analysis of Czech verse. In *Formal methods in poetics* (pp. 295–305). RAM.

Jauralde, P. (2020). *Métrica española*. Cátedra.

Kirszner, L. G. (2013). *Literature: Reading, reacting, writing* (8th ed.). Wadsworth Cengage Learning.

McAleese, W. G. M. (2007). *Improving scansion with syntax: An investigation into the effectiveness of a syntactic analysis of poetry by computer using phonological scansion theory* (Technical report). Department of Computing Faculty of Mathematics, Computing and Technology, The Open University.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. ArXiv: 1301.3781 [Cs]

Moretti, F. (2013). *Distant reading*. Verso.

Onan, A. (2020a). Mining opinions from instructor evaluation reviews: A deep learning approach. *Computer Applications in Engineering Education*, *28*(1), 117–138. https://doi.org/10.1002/cae.22179

Onan, A. (2020b). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency and Computation: Practice and Experience*. e5909. https://doi.org/10.1002/cpe.5909

Onan, A., & Korukoğlu, S. (2017). A feature selection model based on genetic rank aggregation for text sentiment classification. *Journal of Information Science*, *43*(1), 25–38. https://doi.org/10.1177/0165551515613226

Onan, A., Korukoglu, S., & Bulut, H. (2016). LDA-based topic modelling in text sentiment classification: An empirical analysis. *International Journal of Linguistics and Computational Applications*, *7*, 101–119.

Onan, A., Korukoğlu, S., & Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, *57*, 232–247. https://doi.org/10.1016/j.eswa.2016.03.045

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Pérez, Á. (2020). *Corpus of public domain stanzas used for the evaluation of Rantanplan's stanza detection* (LINHD POSTDATA Project). LINHD. Retrieved from https://github.com/linhd-postdata/stanzas-evalution-public

Probst, P., Bischl, B., & Boulesteix, A.-L. (2018). Tunability: Importance of hyperparameters of machine learning algorithms. ArXiv:1802.09596 [Stat].

Quilis, A. (2000). *Métrica española*. Editorial Planeta.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(3), 660–674. https://doi.org/10.1109/21.97458

Torre, E. (2000). *Métrica española comparada*. Universidad de Sevilla.

Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., & MacNeille, P. (2017). A Bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning Research*, *18*(70), 1–37.

Zhou, X., Wan, X., & Xiao, J. (2016). Attention-based LSTM network for cross-lingual sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 247–256). Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1024

How to cite this article: Pérez Pozo, Á., de la Rosa, J., Ros, S., González-Blanco, E., Hernández, L., & de Sisto, M. (2021). A bridge too far for artificial intelligence?: Automatic classification of stanzas in Spanish poetry. *Journal of the Association for Information Science and Technology*, *73*(2), 258–267. https://doi.org/10.1002/asi.24532