Rantanplan, Fast and Accurate Syllabification and Scansion of Spanish Poetry

Rantanplan, silabación y escansión rápidas de poesía española

Javier de la Rosa[1], **Álvaro Pérez**[1],
Laura Hernández[1], **Salvador Ros**[1], **Elena González-Blanco**[2]
[1]Digital Humanities Innovation Lab, UNED, Madrid, Spain
[2]School of Human Sciences and Technology, IE University, Madrid, Spain
{versae, alvaro.perez, laura.hernandez, sros}@scc.uned.es,
egonzalezblanco@faculty.ie.edu

Abstract: Automated analysis of Spanish poetry corpora lacks the richness of tools available for English. The existing options suffer from a number of issues: are limited to fixed-metre hendecasyllabic verses, are not publicly available, the syllabification procedure underneath is not thoroughly tested, and their speed is questionable. Within the context of the ERC POSTDATA Project, this paper introduces new methods to alleviate these concerns. For syllabification, we contribute with our own method and manually crafted corpus. For scansion, our approach is based on a heuristic for the application of rhetorical figures that alter metrical length. Experimental evaluation shows that both fixed-metre and mixed-metre poetry can be successfully analyzed, producing metrical patterns more accurately (increasing accuracy by 2% and 15%, respectively), and at a fraction of the time other methods need (running at least 100 times faster).
Keywords: stress, metrical patterns, scansion

Resumen: El análisis automatizado de la poesía en corpus españoles carece de la riqueza de las herramientas disponibles para el inglés. Las opciones existentes adolecen de una serie de problemas: se limitan a versos endecasílabos de métrica fija, no están disponibles públicamente, el procedimiento de silabación no está probado a fondo, y su velocidad es mejorable. En el contexto del Proyecto ERC POSTDATA, este artículo presenta nuevos métodos para contrarrestar estos problemas. Para la silabación, contribuimos con nuestro propio método, así como un corpus elaborado manualmente. Para la escansión, nuestro enfoque se basa en una heurística para la aplicación de figuras retóricas que alteran la longitud métrica. La evaluación experimental demuestra que tanto la poesía de métrica fija como la de métrica mixta se analizan con éxito, obteniéndose patrones métricos con mayor precisión (mejoras de un 2% y un 15%, respectivamente), y en una fracción del tiempo que otros métodos necesitan (ejecutándose al menos 100 veces más rápido).
Palabras clave: acentuación, patrones métricos, escansión

1 Introduction

Although different in nature, syllabification and scansion are loosely coupled by the underlying functioning of the prosody of a language. Syllabification is the splitting of words into their constituent units, syllables. Unlike English, where there is a weak correspondence between sounds and letters, spoken syllables in Spanish are the basis of the orthographic units of its words. These building blocks shape the stress patterns and rhythm of a language, as well as the poetic metre of its poetry. Once a word is split into syllables, Spanish orthography establishes somewhat rigid rules to assign stress and classifies the words according to the position of the last stressed syllable (there is generally only one stressed syllable per word, with few exceptions (RAE, 2010)[1]). Depend-

[1]When represented, syllables are usually separated by an hyphen (e.g., 'a-mo-ro-so') or an interpunct character ('a·mo·ro·so'), although the former is the preferred way for written syllables. In this work we use hyphens as the syllabic separator for representa-

ing on the stress, there are three categories of words:

- *oxytone* words, when the stressed syllable is the last syllable of the word: 'tam-**bor**'.
- *paroxytone* words, when the stressed syllable is the one before the last syllable of the word: '**plan**-ta'.
- *proparoxytone* words, when the stressed syllable lies two syllables from the end of the word: '**plá**-ta-no'.

Some word functions, such as prepositions, conjunctions, articles, and even some pronouns and determiners, are usually left unstressed for metrical purposes despite having stress assigned by orthographic rules (Caparrós, 1993).

This division of words into stressed and unstressed syllables is the basis for scansion, the process of determining the rhythmic structure or metrical pattern of a line or verse. As defined, it depends entirely on a correct assignment of stress to the syllables of the words of a verse. However, scansion is also affected by some rhetorical devices that might alter the counting of stresses and even syllables present in a verse, thus differentiating between metrical length and syllabic or orthographic length. We can talk about phonological groups for the syllables in a metre, which may be affected by metrical phenomena. Possibly, the two most common of these figures in Spanish are synaloepha and synaeresis. While both imply the union of separate phonological groups, the former acts between the last syllable of a word and the first of the next, for example in 'la amaba', 'la' and 'a' would be joined together. For the latter, the union occurs between the adjacent vowels within a word, 'son-re-ír' can be then split as 'son-reír' after a synaeresis. After applying these alterations to the sounds of words, the number of syllables effectively shrinks for metrical purposes. Diaeresis, on the other hand, is the metric phenomenon in which two vowels within the same syllable forming diphtongs are separated into different syllables, increasing the syllable count. Diaereses tend to be graphically marked with a diacritical sign (¨) (Caparrós, 2014), although its use in modern poetry is becoming less common (Torre, 2000).

Following the definition and representation of Spanish metre given by Navarro-Colorado (Navarro-Colorado, 2017), we consider the metre of a Spanish verse as a sequence of stressed and unstressed syllables (Quilis, 1969; Navarro Tomás, 1991; Caparrós, 1993; Merino, Sánchez, and Pou, 2005), where stressed syllables are marked with the plus symbol '+' and unstressed ones use the minus '−'. An extra unstressed symbol is added to the metrical representation of a verse when its last word is an oxytone, removed if a proparoxytone, or left unchanged if a paroxytone. Example 1 shows a verse of eight syllables and the resulting metrical pattern after applying the pertinent synaloepha (denoted by '‿') and considering the stress of the last word.

(1) *Cuando el alba me despierta*
*Cuan-do‿el-**al**-ba-me-des-**pier**-ta*
− − + − − − +− 8
(Miguel de Unamuno)

It is the extraction of these metrical patterns of stressed and unstressed syllables that we are interested in automating and enhancing. The application of automated techniques would enable corpus linguistic approaches over poetry corpora, that would otherwise need to be annotated manually. At the pedagogical level, it would also allow for the generation of didactic resources for the teaching of poetry and its scansion procedures, as our method produces not only a single output but all the information it relies upon to making its decisions.

2 Related Work

Manuals for metrical analysis of Spanish poetry exist at least since the 18th century (Caparrós, 1975), although the foundational work and subsequent refined guides for modern analysis would take another century to appear (Bello, 1859; Navarro Tomás, 1991; Caparrós, 1993). Despite such a long and rich tradition, not many computational tools have been created to assist scholars in the annotation and analysis of Spanish poetry. With ever increasing corpora sizes and the popularization of distant reading techniques (Moretti, 2013; Jockers, 2013), the possibility of automating part of the analysis be-

tion purposes, marking in bold the stressed syllable (e.g., 'a-mo-**ro**-so').

came very appealing. Although solutions exist, they are either incomplete, not suitable for Spanish (Hartman, 2005; Agirrezabal et al., 2016), or not reproducible. The first of such methods was introduced by Gervás in 2000 as part of a larger system for the automatic generation of metrical poetry (Gervas, 2000). In his work, Gervás uses Definite Clause Grammars in the logic programming language Prolog to model the division of a word into its constituents syllables, adding additional predicates to handle synaloepha and synaeresis. Once a metrical pattern is calculated, is matched against a repository of metrical templates and the best match is returned. There are two issues with this approach: first, all words are assigned their correct orthographic stress regardless of the part of speech. Secondly, all synaloephas are applied indiscriminately since the actual metrical pattern calculated is never returned. How this repository is built is not entirely clear. He reported 88.73% per-line accuracy on a corpus of poems from the Spanish Golden Age period. We could not reproduce the figure since neither the code nor the dataset are publicly available at the moment.

A more modern approach was introduced in 2017 by Navarro-Colorado (Navarro-Colorado, 2017). He built a rule-based system leveraging the morphological analyzer in Freeling (Navarro-Colorado, 2017; Padró and Stanilovsky, 2012) and focused on resolving metrical ambiguities. In his method, after splitting words into syllables and assigning stress according to their PoS, the possible synaloephas and diaeresis are marked and applied, ignoring synaereses. This happens according to a knowledge base with probabilities for the different metrical patterns. The knowledge base is built offline from a large corpus[2] and fed to the system, thus assuming a relationship between high probabilities and metricality. The system was evaluated on more than 1000 lines extracted from a corpus of 100 manually annotated sonnets from the Spanish Golden Age period as well. A considerable increase in per-line accuracy is reported at 95%, contributing further with the first human annotated baseline reporting an inter-annotator agreement of 96%. However, and setting aside the dependence of the system on a correct PoS tagging, as much as 20% of the errors in the evaluation are due to problems related to the use of synaloephas and diaereses, mostly when combined. Moreover, there is no evidence nor evaluation of the ability of Navarro-Colorado's approach to properly assign metrical patterns for lines of verses other than hendecasyllables.

Shortly thereafter, Agirrezabal experimented with the idea of applying neural networks to predict the metrical pattern of lines of verses (Agirrezabal, Alegria, and Hulden, 2017). He designed a character-based bidirectional long short term (BiLSTM) neural network with conditional random fields and trained it on an similar corpus. A prior process of feature engineering added to the syllabification transformed each line of verse into a feature vector that kept the syllabic split, the surroundings of each syllable, PoS tags, and even stresses. He reported a per-line accuracy of 90.84%. Unfortunately, his approach is solely focused on predicting a metrical pattern from a very rich transformation of a verse, loosing in the process all information about phonological groups, individual syllabic stress, and synaloephas, diaereses, and synaereses if any.

Although all approaches rely on a syllabification algorithm, Gervás' system was not made public, and there is no evaluation of Navarro-Colorado's although all his code was made publicly available to experiment with. To the best of our knowledge, the only published syllabification algorithm for Spanish was introduced by Agirrezabal as an extension of his work in the English language. It used a finite state machine to split words into syllables and assign stress following the sonority hierarchy and maximum onset principle (Agirrezabal et al., 2014). However, we found some issues in the syllables of words present in the syllabification corpus employed for evaluation. Based on Ríos Mestre (Mestre, 1998), we disagree in the form some of the words are split into syllables, which could bias the accuracy of his method.

3 Fast Scansion

The aforementioned limitations guided the design of our own syllabification and scansion system, Rantanplan[3], which is comprised of four modules that work together to perform

[2]It is not exactly clear how large this corpus must be for his system to work.

[3]See `https://github.com/linhd-postdata/rantanplan/`

scansion of both fixed-metre as well as mixed-metre poetry: PoS tagger, syllabification, stress assignment, and metrical adjustment. The general algorithm, described in algorithm 1, operates at the line level with a sequence of words. First, for each word in a line of verse the PoS information is extracted and the word split into syllables (lines 2-3 in algorithm 1). Combining the PoS information and the syllabified word, the stress for each syllable is assigned according to the rules for oxytone, paroxytone, and proparoxytone words, plus a few exceptions detailed below (line 4). In the process, all possible synaloephas and synaeresis are marked at the syllable level. With the enriched syllabic data, a new sequence of phonological groups is created by applying all possible synaloephas and synaeresis and keeping the information about the stress positions (line 6). This sequence of phonological groups is translated directly into a metrical pattern (line 7), since each phonological group represents a prosodic unit of pronunciation. The only consideration to factor in is the stress of the ending word, so an extra symbol could be added or subtracted accordingly when necessary. From here, two situations can occur:

1. The expected metrical length is not known, in which case the calculated pattern is returned (line 14).
2. The expected metrical length is known and its value greater than the length of the calculated pattern (lines 8-13). This means some of the applied synaloephas and synaeresis must be undone until both lengths match. The metrical adjustment module will try every option iteratively giving priority based on a heuristic. For each attempt, a new metrical pattern and its corresponding length is calculated and checked against the expected metrical length. If no match is found, the last pattern calculated is returned.

Algorithm 1: Scansion procedure

Input: A sequence $\mathcal{W}$ of words $\langle w_1, w_2, \ldots, w_n \rangle$
Input: A value *length* for the metrical length expected (optional)
Output: A sequence $\langle s_1, s_2, \ldots, s_{\mathcal{L}} \rangle$ of booleans expressing the metrical pattern

1 **for** $w_i \in \mathcal{W}$ **do**
2 $\quad tag_i \leftarrow \text{pos}(w_i)$
3 $\quad syllables_i \leftarrow \text{syllabify}(w_i)$
4 $\quad stresses_i \leftarrow \text{stress}(syllables_i, tag_i)$
5 **end**
6 $groups \leftarrow \text{phonological}(syllables, stresses)$
7 $pattern \leftarrow \text{transform}(groups)$
8 **if** *length* **then**
9 $\quad$ **while** $|pattern| < length$ **do**
10 $\quad\quad g \leftarrow \text{generate\_phonological}(\mathcal{W})$
11 $\quad\quad pattern \leftarrow \text{transform}(g)$
12 $\quad$ **end**
13 **end**
14 **return** *pattern*

3.1 PoS tagger

We built Rantanplan on top of the industrial-strength natural language processing (NLP) framework spaCy for speed (Honnibal and Montani, 2017). As mentioned previously, in Spanish some words are stressed depending on their function in the sentence, hence the need for a proper part of speech tagger. AnCora (Taulé, Martí, and Recasens, 2008), the gold standard corpus many modern statistical language models are trained on for PoS tagging of Spanish texts, splits most affixes thus causing some failures in the tags it assigns on prediction. To circumvent this limitation and to ensure clitics[4] were handled properly, we integrated Freeling's affixes rules via a custom built pipeline for spaCy. The resulting package, spacy-affixes[5], splits words with affixes before assigning PoS, and can be plugged in to a regular spaCy pipeline loading one of the statistical models for Spanish. In our approach, only suffixes on verbs are enabled in spacy-affixes to guarantee clitics are handled adequately by spaCy and PoS tags are assigned correctly.

[4] Syntactically independent but phonologically dependent morphemes that appear together in a word, e.g., in 'cógemelo', both 'me' and 'lo' are pronouns written together with the verb 'coge'

[5] See `https://github.com/linhd-postdata/spacy-affixes/`

3.2 Syllabification

Our method then follows a rule-based algorithm inspired by Ríos Mestre (Mestre, 1998), Caparrós (Caparrós, 1993) and Navarro Tomás (Navarro Tomás, 1991) to split words into syllables. The procedure relies heavily on regular expressions to extract the letter groups that form the syllables. It is comprised of three steps.

1. Pre-syllabification rules are applied, which include the detection of consonant groups other than double 'l', as in 'aislar', and the handling of the prefixes 'sin-' and 'des-' when followed by consonants, as in 'deshielo'.
2. Regular letter clusters are identified and separated from the rest.
3. Post-syllabification exceptions for consonant clusters and diphthongs are applied.

Apart from the official rules for syllabification (RAE, 2010), there are cases with more than one correct way to proceed. The first of these cases was the 'tl' group. Let's take the word 'atlántico' for example, its syllabification changes according to the territory[6]. We decided not to split the group 'tl' since most of the Spanish speakers around the world do not separate it. In the case of words of Nahuatl origin this separation should not be made either. Compounds words and words with an 'h' in between were also challenging. As an example of the former let's take the word 'reutilizar'. Although intuitively it may seem that the prefix 're-' should be separated from the rest of the word, the Fundéu[7] recommends not to do it this way, splitting instead as 'reu-ti-li-zar'. Similarly, the 'h' in a middle position does not split diphthongs, so 'desahijar' would be syllabified as 'de-sahi-jar', which might feel odd at a first pass but it actually agrees with the pronunciation of the word. Moreover, we also included possible diaereses as part of our alternative syllabification exceptions. One such word is 'hiato'[8] which can be split either as 'hia-to' or 'hi-a-to'. As noted by Navarro-Colorado (Navarro-Colorado, 2017), another common case is the word 'suave', which poets tend to apply diaeresis to thus resulting in 'sua-ve' instead of the default split as 'su-a-ve'. Therefore, our method relies on a list of words with alternative syllabifications compiled from Ríos Mestre's work. These alternatives are only taken into account by the metrical adjustment module.

[6]See `https://www.fundeu.es/consulta/at-lan-ti-co-o-a-tlan-ti-co-12213/`

[7]The Fundéu is a foundation created from the Department of Urgent Spanish of the EFE Agency. See `https://twitter.com/Fundeu/status/1182226555457724416`

[8]Several examples can be found in Ríos Mestre (Mestre, 1998), see `http://elies.rediris.es/elies4/Fon8.htm`

3.3 Stress assignment and phonological groups

Once syllables and part of speech of a word are extracted, stress can be assigned. The assignment of stress follows very closely the rules defined in (RAE, 2010), adding exceptions for certain parts of speech, consonant groups, and words that are usually stressed but are not for metrical reasons. The sequence of phonological groups that will be used to extract the metrical pattern is calculated by applying all possible synaereses and synaloephas to the list of syllables of words per line, and propagating the stress information when needed. For example, the words 'me ama' are split into the syllables 'me-**a**-ma', and after applying synaloepha the resulting phonological groups, '**me͡a**-ma', keep the stress in place. Word ends are also marked since they are needed to adjust the length of the metrical pattern according to the position of the stress of the last word. Phonological groups are then transformed into a metrical pattern representation and returned if the expected metrical length of the verse is not known beforehand.

3.4 Metrical adjustment

However, there are situations where the expected metrical length is known, such as processing a corpus of sonnets which tend to be hendecasyllables. In cases like this, verses with applied synaloephas or synaereses but a metrical length lower than the expected would trigger the adjustment module. In example 2, the expected metrical length is 11 but our system returns 9, thus triggering the metrical adjustment module.

(2) *loor a mi autor, y al que leyere*
***loo͡r**-a-mi͡au-**tor**-ya͡l-que-le-**ye**-re*
$+ - - + - - - + - \; 9 < 11$
(Juan de Timoneda)

This means that $11 - 9 = 2$ of the applied synaloephas and synaereses must be undone until both lengths match. The metrical adjustment module tries every possible metrical pattern combining synaereses, synaloephas, and alternative syllabifications. Priority is given to keep the synaloephas since they are rarely broken, and synaeresis are usually undone. The same happens for the alternative syllabifications, which deals with diaeresis and adds more combinations to check for. A special case adding possibilities to the search space is the handling of synaloephas between words with an initial 'h' and vowel ending words. Up to the 16th century, the initial 'h' in words was aspired instead of silent. This depends on the etymology of some words. For example, in the verse '*cubra de nieve la hermosa cumbre*' (see example 3) there should not be synaloepha between 'la' and 'hermosa' since 'hermosa' evolved from the Latin 'fermosa' and as such a synaloepha was not possible at all. To this day, this remains an option to the author, who can decide whether or not to apply a synaloepha in such cases.

(3) *cubra de nieve la hermosa cumbre*
***cu**-bra-de-**nie**-ve-la-her-**mo**-sa-**cum**-bre*
$+ - - + - - - + - + -$ 11
(Garcilaso de la Vega)

For each attempt, a new metrical pattern and length is calculated and checked against the expected metrical length. If no match is found, the last pattern calculated is returned. For the verse in example 2, the generated possible metrical patterns are shown in example 4. Pattern 4 a, with no synaeresis and one synaloepha between 'y' and 'al' would be generated first and returned afterwards. Since the metrical pattern has the correct length it is returned as such and the generation stops, saving the time it takes to generate any other possible pattern. This is also a limitation of our approach since more than one correct metrical pattern can be generated that matches the desired length.

(4) *loor a mi autor, y al que leyere*

(a) *lo-**or**-a-mi-au-**tor**-y‿al-que-le-**ye**-re*
$- + - - - + - - - + -$ 11

(b) *lo-**or**-a-mi‿au-**tor**-y-al-que-le-**ye**-re*
$- + - - + - - - - + -$ 11

(c) ***loor**-a-mi-au-**tor**-y-al-que-le-**ye**-re*
$+ - - - + - - - - + -$ 11

4 Evaluation

One notably difficult aspect of benchmarking automated analysis of Spanish poetry is the lack of a gold standard reference corpus. In recent years, the Corpus of Spanish Golden-Age Sonnets (Navarro-Colorado, Lafoz, and Sánchez, 2016) is being used as the baseline. For syllabification, the best option is the limited corpus by (Agirrezabal et al., 2014)[9]. Unfortunately, it contains some errors thus making it a not reliable source of truth. All evaluations were run on an computer with an Intel® Core™ i7-8550U CPU @ 1.80GHz and 16GiB of DDR4 RAM memory. When reporting figures, accuracy is expressed in percentages and time in seconds.

4.1 Syllabification

Since the only resource for syllabification in Spanish contains errors, we were forced to build our own corpus for the evaluation of the syllabification algorithm. We collected more than 100k words using a combination of online resources[10] into a corpus we named EDFU, and are releasing it under a Creative Commons license[11]. All entries are manually reviewed for correction and compliance with Ríos Mestre and Fundéu recommendations. Table 1 shows the accuracy of the methods by Agirrezabal, Navarro-Colorado, and ours when run against EDFU. Our method performs almost perfectly, more than one percentual point of gain over the others. No time comparison is made since all times are fairly similar.

Method	Accuracy
Navarro-Colorado	98.35
Agirrezabal	98.06
Rantanplan (ours)	**99.99**

Table 1: Scores on EDFU syllabification corpus. Best scores in bold.

[9] See `https://bitbucket.org/manexagirrezabal/syllabification_gold_standard/src/master/`

[10] Namely, `https://educalingo.com`, `https://dirae.es/`, and `https://www.fundeu.es/`

[11] See `https://github.com/linhd-postdata/edfu`

4.2 Scansion

In his original work describing his scansion approach, Navarro-Colorado uses a set of 100 poems (1,400 verses) extracted from the Corpus of Spanish Golden-Age Sonnets (Navarro-Colorado, Lafoz, and Sánchez, 2016) for the evaluation of his system. While the list of the exact 100 poems selected was not made public, the author of the paper kindly provided us with a copy[12]. Since the corpus is comprised entirely of hendecasyllable sonnets, we used it for the evaluation of fixed-metre poetry and compared our results against Agirrezabal's neural network approach, and Navarro-Colorado's rule-based algorithm. Gervás' logic programming method was not available but he kindly agreed to run its system against the fixed-metre corpora and report back the raw outputs. Table 2 summarizes the results of per-line accuracy (evaluated as binary accuracy, entire metrical pattern matches divided by total number of lines of verse), showing that Rantaplan scores better than all other methods. The increase in accuracy is rather small but significant, while our method executes about 150 times faster than Navarro-Colorado's. We are marking the execution times for Gervás and Agirrezabal methods as not available.

Method	Accuracy	Time
Gervás[13]	70.88	N/A
Navarro-Colorado	94.45	2,356s
Agirrezabal	90.84	N/A
Rantanplan (ours)	**96.23**	**21s**

Table 2: Scores on Navarro-Colorado's fixed-metre 1,400 verses corpus. Best scores in bold.

When compared against the entire manually checked part of (Navarro-Colorado, Lafoz, and Sánchez, 2016) (around 10,000 verses from 730 poems), the difference in per-line accuracy increases. Execution time is also added to the comparison. Table 3 shows per-line accuracy of our approach and Navarro-Colorado's system, showing a similar increment in accuracy for our method, around 2% better in metrical pattern calculation, and more than 300 times faster in terms of execution time.

Method	Accuracy	Time
Gervás[14]	67.56	N/A
Navarro-Colorado	90.89	16,787s
Rantanplan (ours)	**92.75**	**53s**

Table 3: Scores on Navarro-Colorado's fixed-metre 10,000 verses corpus. Best scores in bold.

Lastly, for the evaluation of mixed-metre poetry we are using our own corpus of over 4,300 verses obtained from Carjaval's annotated anthology (Fernández-Carvajal, 2003). Unfortunately, due to copyright issues we are unable to release our annotated corpus for mixed-metre poetry. Table 4 shows results comparing performance of our method against Navarro-Colorado's (Navarro-Colorado, 2017), showing that our approach is over 250 times faster and better suited to handle metrical stress that differ from a fixed value with a 15% increase in accuracy over Navarro-Colorado's system.

Method	Accuracy	Time
Navarro-Colorado	49.38	7,484s
Rantanplan (ours)	**65.02**	**27s**

Table 4: Scores on Carvajal's mixed-metre 4,300 verses corpus. Best scores in bold.

In addition to the improvements in accuracy for the different corpora, execution times seem to grow approximately linear with corpus size once we take into consideration that the loading time for the statistical model of Spanish in spaCy is 18 seconds, which gives execution times of 3 seconds for 1,400 verses, 9 seconds for 4,300 verses, and 35 seconds for 10,000 verses.

5 Limitations

Despite the good scores obtained by our method, it is still based on a heuristic. Although thoroughly tested against different corpora, it could be the case that the heuristic we developed cannot account for changes

[12] We are making this corpus available in our corpus downloader tool, Averell: `https://github.com/linhd-postdata/averell/`

[13] Only 1,291 verses of the 1,400 verses corpus were evaluated by Gervás' method.

[14] Similarly, Gervás' method was only evaluated on 9,643 verses of the 10,000 verses corpus.

in poetic production over time, thus making our system incapable of identifying accurately metrical patterns in modern expressions of poetry. We would need a more recent corpus to test this issue, but unfortunately most of these texts are still under copyright.

A second important limitation of our method is the use of a PoS tagger that relies on a statistical language model optimized for speed, which in some cases assigns incorrect part of speech tags, thus impacting the stress of the words and producing inaccurate metrical patterns.

6 Conclusions and Further Work

In this paper we have proposed methods for the automatic syllabification and scansion of Spanish poetry. Our syllabification method benefits from a carefully crafted new corpus, which we are releasing to the public. For scansion, two are the main advantages. First, we used a modern language model optimized for speed for the extraction of part of speech tags, improving execution times by a couple of orders of magnitude. Lastly, when extracting the actual metrical pattern we took the opposite approach to the previous state of the art and decided to apply all possible synaloephas and synaereses by default, only breaking them up when needed to match metrical length. This decision paid off well in terms of accuracy since our method outperformed the rest in both fixed-metre and mixed-metre poetry.

We plan to continue improving Rantanplan and explore alternatives, specially using statistical language models to produce end-to-end metrical patterns further improving speed. Moreover, the outpus produced by our method will eventually be machine readable, interoperable, and ready to be ingested into a triple store compliant with the POSTDATA Project network of ontologies. After all, syllabification and scansion are the necessary building blocks for achieving stanza and structure identification, a long-term goal of the project.

Acknowledgements

Research for this paper has been achieved thanks to the Starting Grant research project Poetry Standardization and Linked Open Data: POSTDATA (ERC-2015-STG-679528) obtained by Elena González-Blanco. This project is funded by the European Research Council (https://erc.europa.eu) (ERC) under the research and innovation program Horizon2020 of the European Union.

References

[Agirrezabal, Alegria, and Hulden2017] Agirrezabal, M., I. Alegria, and M. Hulden. 2017. A comparison of feature-based and neural scansion of poetry. *arXiv preprint arXiv:1711.00938*.

[Agirrezabal et al.2016] Agirrezabal, M., A. Astigarraga, B. Arrieta, and M. Hulden. 2016. Zeuscansion: a tool for scansion of english poetry. *Journal of Language Modelling*, 4.

[Agirrezabal et al.2014] Agirrezabal, M., J. Heinz, M. Hulden, and B. Arrieta. 2014. Assigning stress to out-of-vocabulary words: three approaches. In *International Conference on Artificial Intelligence, Las Vegas, NV*, volume 27, pages 105–110.

[Bello1859] Bello, A. 1859. *Principios de la ortolojía i métrica de la lengua castellana..* la Opinión.

[Caparrós1975] Caparrós, J. D. 1975. *Contribución a la historia de las teorías métricas en los siglos XVIII y XIX*, volume 92. Editorial CSIC-CSIC Press.

[Caparrós1993] Caparrós, J. D. 1993. *Métrica española*. Síntesis Madrid.

[Caparrós2014] Caparrós, J. D. 2014. Teoría métrica del verso esdrújulo. *Rhythmica: revista española de métrica comparada*, 12:55–96.

[Fernández-Carvajal2003] Fernández-Carvajal, F. 2003. Antología de textos.

[Gervas2000] Gervas, P. 2000. A logic programming application for the analysis of spanish verse. In *International Conference on Computational Logic*, pages 1330–1344. Springer.

[Hartman2005] Hartman, C. O. 2005. The scandroid 1.1. *Software available at http://oak. conncoll. edu/cohar/Programs. htm.*

[Honnibal and Montani2017] Honnibal, M. and I. Montani. 2017. spacy 2: Natural

language understanding with bloom embeddings. *Convolutional Neural Networks and Incremental Parsing*, 7.

[Jockers2013] Jockers, M. L. 2013. *Macroanalysis: Digital methods and literary history*. University of Illinois Press.

[Merino, Sánchez, and Pou2005] Merino, E. V., P. M. Sánchez, and P. J. Pou. 2005. *Manual de métrica española*. Editorial Castalia.

[Mestre1998] Mestre, A. R. 1998. *La transcripcion fonetica automatica del diccionario electronico de formas simples flexivas del español: un estudio fonologico en el lexico*. Ph.D. thesis, Universitat Autònoma de Barcelona.

[Moretti2013] Moretti, F. 2013. *Distant reading*. Verso Books.

[Navarro-Colorado2017] Navarro-Colorado, B. 2017. A metrical scansion system for fixed-metre spanish poetry. *Digital Scholarship in the Humanities*, 33(1):112–127.

[Navarro-Colorado, Lafoz, and Sánchez2016] Navarro-Colorado, B., M. R. Lafoz, and N. Sánchez. 2016. Metrical annotation of a large corpus of spanish sonnets: representation, scansion and evaluation. In *International Conference on Language Resources and Evaluation*, pages 4360–4364.

[Navarro Tomás1991] Navarro Tomás, T. 1991. Métrica española. *Reseña histórica y descriptiva*, 50.

[Padró and Stanilovsky2012] Padró, L. and E. Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *International Conference on Language Resources and Evaluation*.

[Quilis1969] Quilis, A. 1969. *Métrica española*. Alcalá Madrid.

[RAE2010] RAE, R. A. E. 2010. *Ortografía de la lengua española*. Espasa.

[Taulé, Martí, and Recasens2008] Taulé, M., M. A. Martí, and M. Recasens. 2008. Ancora: Multilevel annotated corpora for catalan and spanish. In *International Conference on Language Resources and Evaluation*.

[Torre2000] Torre, E. 2000. *Métrica española comparada*, volume 48. Universidad de Sevilla.

7 Appendix: Reproducibility

To reproduce the results in this paper, please, refer to the next code repository: `https://github.com/linhd-postdata/rantanplan-evaluation`

8 Appendix: Availability

A demo of the scansion system can be found online at `http://postdata.uned.es/poetrylab/`. All source code is available under an Apache License 2.0 in a public code repository (`https://github.com/linhd-postdata/rantanplan/`) and as Python package in PyPI (`https://pypi.org/project/rantanplan/`).