Towards Multilingual Haikus: Representing Accentuation to Build Poems

Fernando Bobillo[1,2], Maxim Ionov[1], Eduardo Mena[1,2], Carlos Bobed[1,2]
[1]University of Zaragoza, Zaragoza, Spain,
[2]Aragon Institute of Engineering Research (I3A), Zaragoza, Spain
Correspondence: {fbobillo,mionov,emena,cbobed}@unizar.es

Abstract

The paradigm of neuro-symbolic Artificial Intelligence is receiving an increasing attention in the last years to improve the results of intelligent systems by combining symbolic and sub-symbolic methods. For example, existing Large Language Models (LLMs) could be enriched by taking into account background knowledge encoded using semantic technologies, such as Linguistic Linked Data (LLD). In this paper, we claim that LLD can aid Large Language Models by providing the necessary information to compute the number of poetic syllables, which would help LLMs to correctly generate poems with a valid metric. To do so, we propose an encoding for syllabic structure based on an extension of RDF vocabularies widely used in the field: POSTDATA and OntoLex-Lemon.

1 Introduction

Neuro-symbolic Artificial Intelligence is a hybrid paradigm which combines both symbolic (e.g., semantic technologies such as ontologies and knowledge graphs) and sub-symbolic (e.g. neural networks and transformer-based Large Language models or LLMs) methods, trying to leverage the advantages of both of them (Hitzler et al., 2020).

To develop Neuro-symbolic Artificial Intelligence applications, we have proposed an architecture (illustrated in Figure 1) for hybrid intelligent systems called *HAIKU (Hybrid Artificial Intelligence on Knowledge and lingUistics)* (Bobillo et al., 2025). The main knowledge of the system would be stored using ontologies (as schema) populated in knowledge graphs and Linked Data (possibly including Linguistic Linked Data, LLD), but the system would also leverage transformer-based NLP models offering various services, such as user communication. The communication is bidirectional: NLP services can improve the semantic knowledge base, and semantic knowledge can improve NLP services.

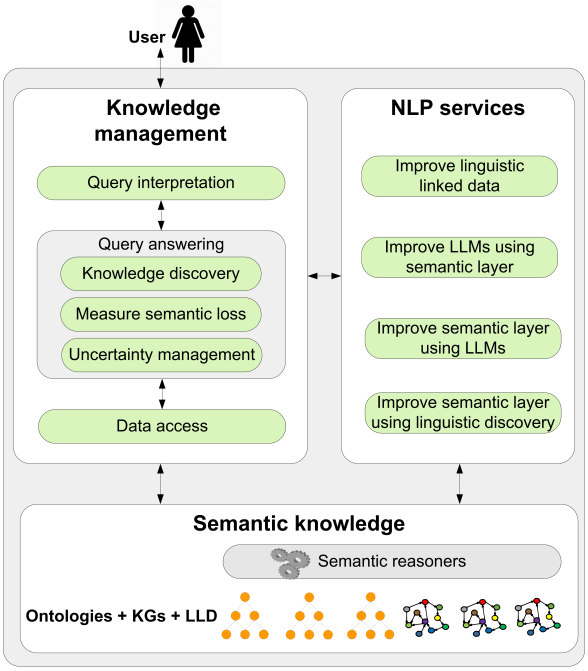

Figure 1: HAIKU architecture (Bobillo et al., 2025).

The motivations behind the HAIKU architecture are the limitations of both Natural Language Processing (NLP) systems and semantic-based systems when they do not cooperate. For example, although LLMs are a useful tool in many cases, they typically perform better on English than on other languages, and the situation worsens even further for dialects or minority languages (Kantharuban et al., 2023).

As a concrete example, by playing with the name of our architecture, let us consider the problem of obtaining a haiku in Spanish, which requires a correct use of syllables in poetry. A haiku is a Japanese poem with 3 lines of 5, 7, and 5 poetic syllables, respectively. We asked ChatGPT-4 (the most recent free version, at the time of writing) to write a haiku about heavy metal music, in English. The answer

> “Riffs shake earth and sky,
> thunder roars, hearts pounding wild
> metal gods arise”

correctly respected the metric of haikus, but when repeating the same query in Spanish, the answer

> “Hierro y trueno va,
> el grito rompe la noche,
> fuego en el altar”

was made up of 3 verses of 6, 8 and 6 poetic syllables, respectively, which is incorrect as a haiku[1]

We also tried to refine the answer of the LLM. Firstly, we asked it to explain how the metric syllables were computed. Since its answer was incorrect, we pointed it out some mistakes (the word “hierro”, iron in Spanish, has two syllables and words ending in “l” do not form a synalepha[2]) and asked it to provide a new answer, which was again incorrect, having 4, 8 and 5 poetic syllables. Even after a third iteration where we pointed out some errors, the answer was still incorrect as a haiku, having again 4, 8 and 5 poetic syllables. Details about the experiment can be found in Appendix A.

Of course, it is not desirable to expect the user to provide the division of a verse into poetic syllables, neither as part of the initial prompt nor by asking the LLM to refine the answers. Our aim is not to criticize existing LLMs (indeed, we only considered a single LLM to illustrate our claim), but to point out that future intelligent systems must be able to know things like the correct number of poetic syllables in a verse. This could be implemented directly in the system or it could use an external service. In any case, we claim that the use of Linguistic Linked Data (Cimiano et al., 2020) (LLD) would help to solve this problem: LLD would provide the system the necessary information to compute the number of poetic syllables correctly. The objective of this short paper is to present a possible data representation based on LLD that can express all the relevant information needed to compute the number of poetic syllables in Spanish.

The remainder of this paper is structured as follows. Section 2 provides some background on poetic syllables in Spanish. Then, Section 3 discussed our modelling. Finally, Section 4 sets out some conclusions and ideas for future work.

[1]The first verse is also grammatically incorrect: the subject is plural but the verb (“va”) is singular.

[2]The merging of two syllables into one, especially when it causes two words to be pronounced as one.

2 Poetic syllables in Spanish

Computing the poetic syllables of a verse in Spanish[3] requires the following steps:

- Firstly, the total number of grammatical syllables in the verse is computed.
- Secondly, the last word of the verse is considered: if it is **oxytone**, i.e. the stress in that word falls on the last syllable, a poetic syllable is added, whereas if it is **proparoxytone**, i.e. the stress falls on the third to last syllable, a poetic syllable is subtracted. In **paroxytone** words, i.e. with stress on the penultimate syllable, the number of syllables does not change.

  If a word is written with an accent, by separating it into syllables we can easily see whether it is proparoxytone, paroxytone, or oxytone. If the word is written without an accent, it will be oxytone if it ends in a vowel, “n” or “s”, otherwise it will be paroxytone.
- Finally, **synalephas** are considered: if any word in the verse ends in a vowel, “h”, or “y”, and the next word is “y” or begins with a vowel or “h”, both syllables count as one.

The procedure is similar in other Romance languages (e.g., Italian and Galician). In some languages (e.g., French and Catalan), there is a notable difference: only the number of metrical syllables until the last stressed syllable is taking into account, but one still needs to know whether the last word is oxytone, paroxytone, or proparoxytone.

In principle, in order to both calculate the number of syllables in a verse and to check whether a word with a written accent is proparoxytone, paroxytone, or oxytone, a syllabification algorithm would be needed, that is, one that separates a word into syllables. Implementing these algorithms is not easy and requires deep domain knowledge, due to the large number of possible exceptions.

An alternative is to use a linguistic knowledge base where each word is already separated by syllables. Additionally, the knowledge base could also indicate for each word whether it is proparoxytone, paroxytone, or oxytone: without being strictly obligatory, it would increase the efficiency of the system by avoiding having to calculate it. For example, we could use a pronunciation dictionary such as (Quilis et al., 1999), which represents the

[3]Here, we do not take into account poetic licenses.

Spanish word “uva” (grape) as “oo’-vah”, where the hyphen separates the syllables (“u” and “va”) and the apostrophe indicates the stressed syllable (“u“). That dictionary exists only in paper form, but even if a version of it were available in digital format, it would most likely be only in a human- but not machine-readable format, such as HTML or PDF. In order to make it useful for LLMs and other non-human consumers, it would be desirable to have a representation in a more machine-friendly format, such as RDF and have it accessible via a SPARQL endpoint.

3 LLD-based solutions

In this section, we will discuss LLD-based solutions to represent the syllables and the stressed syllable of a given word. Typically, LLD use RDF[4], a W3C standard framework for representing information. Although RDF can be serialized into different formats, we will use Turtle syntax, which is more easily understandable by humans. For example, the triple ‘s p o .” states that a subject s is related with an object o via a property p.

POSTDATA. As a basis for our modelling we reuse the set of ontologies created in the POSTDATA (Poetry Standardization and Linked Open Data) project (Bermúdez-Sabel et al., 2022). In particular, the *postdata-structural* (pdstruct)[5] and the *poetic-analysis* (pdp)[6] ontologies to represent the basic structure and the literary analysis properties, respectively.

Using POSTDATA, we can represent words as instances of the class pdstruct:Word and grammatical syllables as instances of the class pdstruct:Syllable. Each word is connected to the first and the last syllables that form it via the object properties pdstruct:hasFirstSyllable and pdstruct:hasLastSyllable, respectively. Furthermore, each syllable is related to the next and the previous one via the data properties pdstruct:nextSyllable and pdstruct:previousSyllable, respectively, which makes it possible to navigate through all the syllables of a word. For a given syllable, pdp:positionInWord is a functional data property with an xsd:integer value to represent the position of the syllable from the end of the word (e.g., 1 for the stressed syllable of an oxytone word, 2 for the stressed syllable of a paroxytone word, 3 for the stressed syllable of a proparoxytone word, etc.), as illustrated in Figure 2.

However, it is not possible to represent whether a pdstruct:previousSyllable is stressed or not[7]. To do so, one must use metrical syllables. The class pdstruct:Line makes it possible to represent a line of a poem and, for a given line, pd:hasMetricalSyllableList retrieves a list of metrical syllables. Given a metrical syllable, pdp:isStressed is a functional data property indicating whether the syllable is stressed or not using an xsd:boolean value, whereas pdp:metricalSyllableNumber is a functional data property using an xsd:integer value to represent the position of the stressed syllable from the end of the line. This is illustrated in Figure 3.

While using lines is suitable for poems, it is not possible in our case since we want to encode a list of words with their syllable structures. It is possible to represent each word in the dictionary (e.g., “uva”) as a line (pdstruct:Line), buth this is highly undesirable.

Furthermore, to encode the stressed syllable, grammatical syllables would have to be represented as metric syllables, which is also not the ideal situation. Note indeed that the division of a word into metrical syllables might not be unique, as authors could use poetic licenses such as synaeresis, diaeresis, or hiatus.

Two new properties. As a solution to the previously mentioned limitations of POSTDATA for our use-case, we propose two new properties haiku:tonicSyllable and haiku:hasSyllables, where haiku is a new vocabulary:

- haiku:tonicSyllable is a data property intended to link a word to a numeric value representing the stressed syllable, starting from the end of the word. Thus, “1” corresponds to an oxytone word, “2” to a paroxytone word, and a value strictly greater than two to a proparoxytone or over-proparoxytones word.

- haiku:hasSyllables is an object property intended to link a word to an (ordered) list of strings, each of which represents one of the grammatical syllables of the word. haiku:hasSyllables is somehow similar to pdstruct:hasMetricalSyllableList, but with different domais and range (recall that

[4] https://www.w3.org/TR/rdf11-primer/

[5] https://postdata.linhd.uned.es/ontology/postdata-structuralElements/documentation/index-en.html

[6] https://postdata.linhd.uned.es/OntoPoetry/Poetic/documentation/index-en.html

[7] Note that Bermúdez-Sabel et al. (2022, Figure 3) uses an old version of the ontology, where it was possible.

```
@base <http://www.example.org/lexicon> .
@prefix pdstruct: <http://postdata.linhd.uned.es/ontology/
                                                postdata-structuralElements#> .
@prefix pdp: <http://postdata.linhd.uned.es/ontology/postdata-poeticAnalysis#> .

:word_uva a pdstruct:Word ;
            pdstruct:content "uva"@es ;
            pdstruct:hasFirstSyllable :syllable_u;
            pdstruct:hasLastSyllable :syllable_va .

:syllable_u a pdstruct:Syllable ;
              pdstruct:content "u" ;
              pdstruct:nextSyllable :syllable_va ;
              pdstruct:positionInWord 1 .

:syllable_va a pdstruct:Syllable ;
               pdstruct:content "va" ;
               pdstruct:previousSyllable :syllable_u ;
               pdstruct:positionInWord 2 .
```

Figure 2: Example of representation of the grammatical syllables of the Spanish word "uva" in POSTDATA.

```
@base <http://www.example.org/lexicon> .
@prefix pdstruct: <http://postdata.linhd.uned.es/ontology/
                                                postdata-structuralElements#> .
@prefix pdp: <http://postdata.linhd.uned.es/ontology/postdata-poeticAnalysis#> .

:word_uva a pdstruct:Line ;
            pdstruct:content "uva" ;
            pdp:hasMetricalSyllableList :syllable_list_uva .

:syllable_list_uva a pdp:MetricalSyllableList ;
                     pdp:firstMetricalSyllable :syllable_u .

:syllable_u a pdp:MetricalSyllable ;
              pdp:content "u" ;
              pdp:metricalSyllableNumber 1 ;
              pdp:isStressed true ;
              pdp:nextMetricalSyllable :syllable_va  .

:syllable_va a pdp:MetricalSyllable ;
               pdp:content "va" ;
               pdp:metricalSyllableNumber 2 ;
               pdp:isStressed false .
```

Figure 3: Example of representation of the metrical syllables of "uva" in POSTDATA.

pdstruct:hasMetricalSyllableList links a line with a list of metrical syllables).

OntoLex Lemon. So far, we have proposed to use two novel properties that state some information about a word, but the representation of the word has not been discussed in detail. One option would be to use OntoLex-Lemon model, a W3C vocabulary that provides rich linguistic grounding for ontologies and is a *de facto* standard to represent lexical resources such as dictionaries as RDF data (McCrae et al., 2017).

In particular, the domain of the novel properties can be ontolex:Form, a class that represents a surface form of a lexical entry. In our case, this can be used to point to the canonical form of a word.[8] Therefore, haiku:tonicSyllable property links (the canonical form of) a word to a numeric value representing the stressed syllable, whereas haiku:hasSyllables property links (the canonical form of) a word to an (ordered) list of grammatical syllables represented as strings. Figure 4 shows how to associate the hyphenation and accentuation to a word encoded in OntoLex Lemon.

[8]The logic is preserved if we want to represent syllabic structures of inflected forms, but in our case this situation does not arise since we deal with dictionary entries.

```
@base <http://www.example.org/lexicon> .
@prefix lime: <http://www.w3.org/ns/lemon/lime#> .
@prefix haiku: <https://sid.cps.unizar.es/vocab#> .
@prefix ontolex: <http://www.w3.org/ns/lemon/ontolex#> .
:lexicon_en a lime:Lexicon ;
              lime:language "es" ;
              lime:entry :uva.
:uva rdfs:label "uva"@es ;
     ontolex:canonicalForm [
            ontolex:writtenRep "uva"@es ;
            haiku:stressedSyllable 2 ;
            haiku:hasSyllables ("u" "va")
     ]
```

Figure 4: Possible representation of the stressed syllable and grammatical syllables of "uva" in OntoLex Lemon (the novel vocabulary is highlighted in blue).

Finally, it is worth to note that OntoLex Lemon is not intended to be generalized by other authors, so rather than proposing two novel properties within OntoLex Lemon (e.g., in the lime module), we chose to use a novel vocabulary.

4 Conclusions and future work

In this paper, we showed how to use Linguistic Linked Data to represent the necessary information to compute the number of poetic syllables in Spanish. While existing vocabularies such as POSTDATA are appropriate to represent the metric syllables of an existing poem, they had to be extended for our purposes. Therefore, we proposed two novel properties to represent the hyphenation and accentuation of a word, which could be represented using OntoLex Lemon vocabulary.

The next step is to support intelligent systems in the automatic generation of poems with a valid metric. For this, Linguistic Linked Data could be used for knowledge injection in existing Large Language Models, improving them. This illustrates the usefulness of our approach within the field of Neurosymbolic Artificial Intelligence.

In future work, apart from the actual application of our vocabulary for knowledge injection in LLMs, we could generalize the vocabulary to support different languages. While our novel properties are enough to infer the number of poetic syllables in Spanish, other languages might require different information. Furthermore, other types of metrics apart from haikus might be considered. Thus, our novel vocabulary is intended to be generalized with more properties, if needed.

Limitations

Our solution focuses on Spanish and other languages sharing a similar way to compute the number of metric syllables, but not for other languages.

Acknowledgments

We were partially supported by the I+D+i projects PID2020-113903RB-I00, PID2024-159530OB-I00 (funded by MCIN/AEI/10.13039/501100011033) and T42_23R (Gobierno de Aragón).

References

Helena Bermúdez-Sabel, María Luisa Díez Platas, Salvador Ros, and Elena González-Blanco. 2022. Towards a common model for european poetry: Challenges and solutions. *Digital Scholarship in the Humanities*, 37(4):921—-933.

Fernando Bobillo, Eduardo Mena, Jorge Gracia, and Carlos Bobed. 2025. HAIKU: Hybrid artificial intelligence on knowledge and linguistics. In *Actas de las XXIX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2025)*.

Philipp Cimiano, Christian Chiarcos, John P. McCrae, and Jorge Gracia. 2020. *Linguistic Linked Data - Representation, Generation and Applications*. Springer.

Pascal Hitzler, Federico Bianchi, Monireh Ebrahimi, and Md. Kamruzzaman Sarker. 2020. Neural-symbolic integration and the semantic web. *Semantic Web*, 11(1):3–11.

Anjali Kantharuban, Ivan Vulić, and Anna Korhonen. 2023. Quantifying the dialect gap and its correlates across languages. In *Proc. of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pages 7226–7245. ACL.

John P. McCrae, Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. 2017. The Ontolex-Lemon model: development and applications. In *Proc. of the eLex 2017*, pages 19–21.

Antonio Quilis, Celia Casado-Fresnillo, and Rafael Marc. 1999. Dos diccionarios de pronunciación [in Spanish]. *Revista Española de Lingüística*, 29(2):437–453.

Appendix A Prompts used

To create the haiku in English, we simply used:

> `Write a haiku about heavy metal`

To obtain the Spanish haiku, we firstly asked:

> `Escribe un haiku sobre el heavy metal`

Since the answer was unsatisfactory (having 6, 8 and 6 poetic syllables), we specifically asked to consider the the metric rules in Spanish:

> `Ten en cuenta al escribirlo las normas de las métricas en español, por favor. Tienes que tener en cuenta que si la última palabra es aguda, se suma una sílaba métrica; si es llana, se deja igual, y si es esdrújula, se resta una. Además, también tienes que cuidar la sinalefa, según la cual dos sonidos vocálicos al principio y fin de dos palabras consecutivas hacen que solo se cuente una sílaba (por ejemplo, "ruge el" sólo sería una sílaba métrica)`

The LLM claimed that the answer was correct (“Este haiku también sigue correctamente la métrica 5-7-5, respetando tanto la sinalefa como la acentuación de las palabras”), which was not the case, so we pointed out a specific error with a synalepha:

> `Hierro son dos sílabas, y las palabras que terminan en l no hacen sinalefa`

Since the new haiku provided as an aswer was still incorrect (having 4, 8 and 5 poetic syllables), we tried again to point out specific mistakes: a synalepha and two wrong numbers of syllables:

> `Te has saltado la sinalefa de "rro y", el segundo párrafo suma 8 sílabas y la última también suma 6 según lo que dices (aunque es verdad que hay una sinalefa)`

Unfortunately, the new answer still had 4, 8 and 5 poetic syllables.