

# TOWARDS MULTILINGUAL HAIKUS: REPRESENTING ACCENTUATION TO BUILD POEMS

Fernando Bobillo<sup>1,2</sup>, Maxim Ionov<sup>1</sup>, Eduardo Mena<sup>1,2</sup>, Carlos Bobed<sup>1,2</sup>

<sup>1</sup>University of Zaragoza, Zaragoza, Spain  
<sup>2</sup>Aragon Institute of Engineering Research (I3A), Zaragoza, Spain

## Introduction

- **Neuro-symbolic AI** is a **hybrid paradigm** which combines both **symbolic** (e.g., semantic technologies, knowledge graphs) and **sub-symbolic** (e.g. LLMs) approaches.
- This paper introduces an **architecture for combining** both, **HAIKU** (Hybrid Artificial Intelligence on Knowledge and linguistics).
- We claim that **Linguistic Linked Data (LLD)** can **aid Large Language Models (LLMs)**

## The HAIKU Architecture

- We propose **HAIKU** to combine symbolic and sub-symbolic methods. This system:
- Leverages semantic knowledge from ontologies and knowledge graphs / Linguistic Linked Data (LLD).
  - Uses ontologies (as schema) populated in knowledge graphs and Linked Data to store its knowledge.
  - Establishes communication between the KG and Transformer-based NLP models.
  - The communication is **bidirectional**: NLP services can improve the semantic knowledge base, and semantic knowledge can improve NLP services.

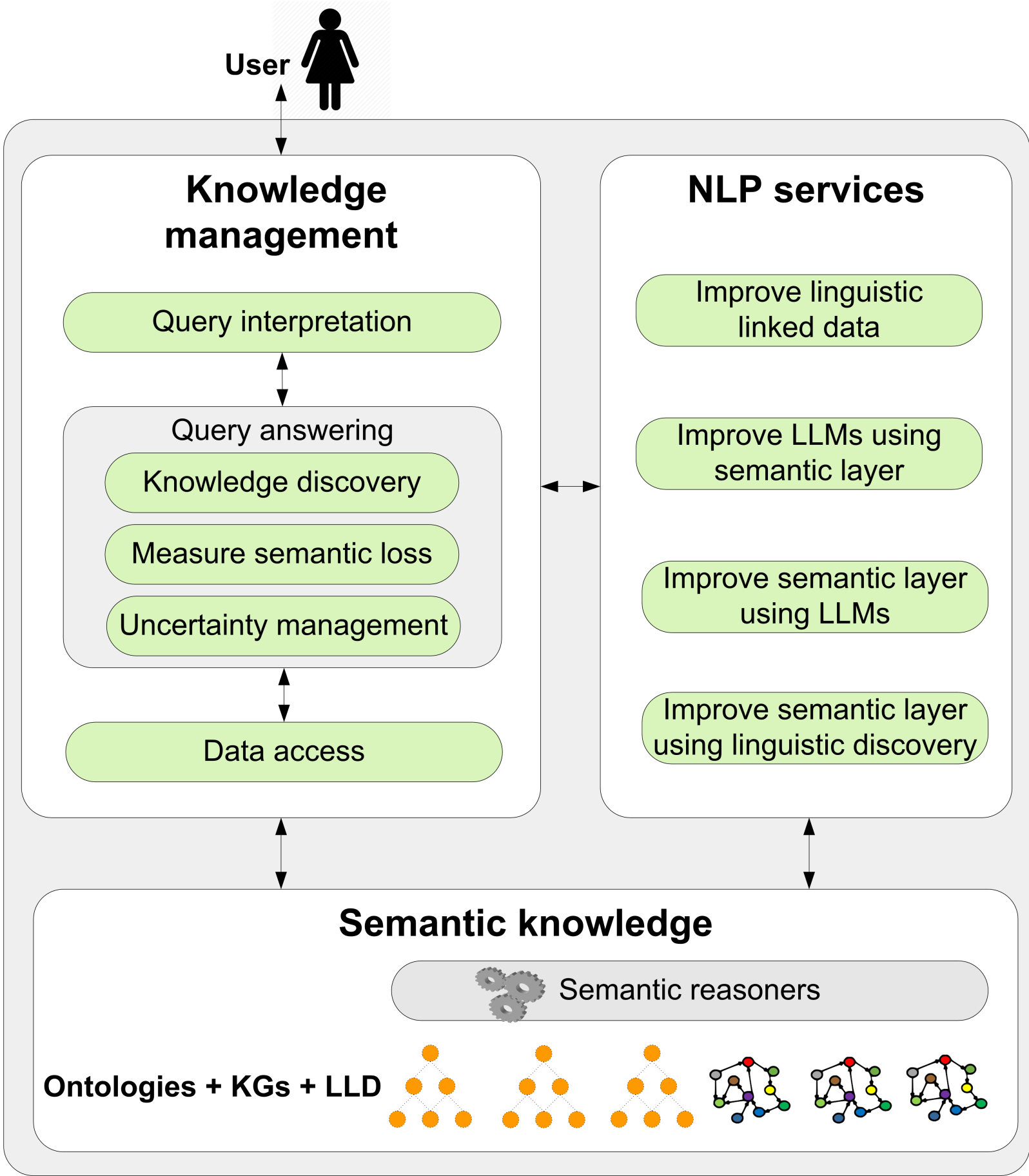


Figure 1: HAIKU architecture.

## Naive experiments: LLM Prompting

- **Task:** Prompting ChatGPT-4 (most recent free version at the time) to write a haiku about heavy metal.
- **English result:**  
*Riffs shake earth and sky, // thunder roars, hearts pounding wild // metal gods arise*  
→ Correct 5–7–5 structure.
- **Spanish result:**  
*Hierro y trueno va, // el grito rompe la noche, // fuego en el altar*  
→ Incorrect: 6–8–6 syllables; also grammatical error (plural subject with singular verb).
- **Replication:** Repeated the query several times; obtained similar incorrect results in Spanish.
- **Solution:**
  - Intelligent systems must be able to count the correct number of poetic syllables in a verse.
  - Could be implemented directly or with an external service.
  - We claim that the use of LLD would help to solve this: LLD would provide the system the necessary information to compute the number of poetic syllables correctly.

## Poetic Syllables in Spanish

- Computing poetic syllables in Spanish** involves three steps:
1. Count the total number of grammatical syllables in the verse.
  2. Adjust based on the last word:
    - **Oxytone** (stress on last syllable): add one syllable.
    - **Paroxytone** (stress on penultimate syllable): no change.
    - **Proparoxytone** (stress on third-to-last syllable): subtract one syllable.If the word has a written accent, its stress is explicit. If not: → ends in vowel, “n” or “s” ⇒ oxytone; otherwise ⇒ paroxytone.
  3. Apply **synalephas**: if a word ends in a vowel, “h” or “y”, and the next word is “y” or begins with a vowel/“h”, both syllables merge into one.

## LLD-based Encoding

Most of the data can be **represented with POSTDATA** (Poetry Standardization and Linked Open Data)

```
@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 .
```

**Problem:** there is no direct way to mark the stressed (tonic) syllable.

**Solution:** Two new properties (haiku vocabulary):

1. **‘haiku:tonicSyllable’** → numeric value indicating stressed syllable: 1 = oxytone, 2 = paroxytone, >2 = proparoxytone.
  2. **‘haiku:hasSyllables’** → ordered list of strings representing grammatical syllables.
- Provides direct, unambiguous stress and syllable information.
  - Simplifies NLP tasks like haiku validation and syllable counting.

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

## Example: Haiku Generation

- In this pilot study, we explore how LLMs fail to *consistently* generate valid haikus (3 lines of 5, 7, and 5 poetic syllables);
- And explore the encoding of haikus for their syllabic structure based on an extension of widely used RDF vocabularies: OntoLex-Lemon and POSTDATA (designed for poetry encoding);
- In the future, using this structured knowledge should improve LLM-based haiku generation.

## Conclusion and Future Work

- This work showed that LLD provides crucial symbolic knowledge and that LLMs need to correctly handle tasks like generating poetry with valid metric rules.
- **Future Work:**
  - Experiment with combining LLD and LLMs
  - Extend this approach to more poetic licenses.
  - Apply the methods to languages beyond Spanish.