Assignment 1 – Animal Scouter

University of Aveiro

Web Semântica

2022/2023

Group nº 2:

* João Bernardo Coelho Leite - 115041
* João Pedro dos Reis - 115513
* Luís Miguel Gomes Batista - 115279



Index

[Introduction 2](#_Toc135953454)

[Data Operations (SPARQL) 3](#_Toc135953455)

[Transformations 3](#_Toc135953456)

[Ontology definition 4](#_Toc135953457)

[Ontology (RDFS & OWL) 4](#_Toc135953458)

[SPIN Inferences 5](#_Toc135953459)

[Use of Wikidata & DBpedia 6](#_Toc135953460)

[Application Features 6](#_Toc135953461)

[Conclusions 6](#_Toc135953462)

[Application Configuration 7](#_Toc135953463)

[Requirements 7](#_Toc135953464)

[Creating the database 7](#_Toc135953465)

[Running with PyCharm 7](#_Toc135953466)

[Running with command line 7](#_Toc135953467)

[References 8](#_Toc135953468)

Introduction

In current times, websites serve as powerful platforms for information dissemination, interaction, and engagement. As technologies continue to evolve, web developers are constantly seeking innovative ways to enhance user experiences and improve the functionality of their websites. One such avenue of exploration involves leveraging the capabilities of inferences and SPIN (SPARQL Inferencing Notation) to extract meaningful insights from data and enable advanced functionalities.

This report explores the potential of incorporating inferences and SPIN (SPARQL Inferencing Notation) into a website. It focuses on creating a new tab on an existing website, using a transformed dataset of animal features in N3 format OWL. The ontology editor Protégé is utilized to define rules, axioms, and constraints for managing inferences. Additionally, the integration of data from Wikidata and DBpedia is considered to enrich the dataset and enhance the website's functionality.

We aim to demonstrate the benefits of inferences and SPIN in improving the website's user experience and providing comprehensive information. By integrating external knowledge bases, the website can tap into a broader range of structured data, establish semantic connections, and offer a more interconnected web experience.

In summary, this paper showcases the possibilities and advantages of incorporating inferences and SPIN into website design.

Data Operations (SPARQL)

## Transformations

The initial dataset, represented in N-Triples format, contained statements in the form of subject-predicate-object triples, such as:

A picture containing text, screenshot, font, line

Description automatically generated

Figure 1 Example of original dataset N-Triples formatting

To maximize the inferential capabilities of the dataset and enable more sophisticated reasoning, a transformation process was undertaken to convert the dataset into N3 format using OWL (Web Ontology Language) techniques. This transformation enhanced the dataset's expressiveness and provided a more structured representation of the information.

The transformed dataset, now in N3 format, follows a different syntax that offers improved readability and organization. Each entity in the dataset is assigned a prefixed name to enhance clarity and facilitate efficient referencing.

In the N3 representation, the subject <http://zoo.org/animal/id/abelha> has been transformed into the prefixed name ‘zooi:abelha’, improving readability and providing a more concise identification. Similarly, the predicates and objects have bee n transformed into prefixed names and values, respectively, to ensure clarity and maintain consistency throughout the dataset.

A picture containing text, font, screenshot, receipt

Description automatically generated

Figure 2 Example of N3 formatted entry of the dataset

By employing OWL techniques and transforming the dataset into N3 format, we have enhanced the expressiveness and readability of the dataset. This transformation sets the foundation for more sophisticated reasoning and enables the utilization of advanced inference techniques and SPIN.

In addition, the attributes “Land”, “Backbone” and “Predator” were added to the dataset to increase the possibility and accuracy of inferences in the future.

Ontology definition

## Ontology (RDFS & OWL)

The ontology presented in this report defines a comprehensive taxonomy and relationships within the domain of animals. It serves as a structured knowledge representation that enables reasoning and inference about different types of animals, their attributes, and classifications.

The ontology is built using the RDF/OWL syntax. It begins by establishing the necessary prefixes for namespaces used in the ontology, such as ‘owl’, ‘rdf’, ‘rdfs’, and custom prefixes like ‘zooc’, ‘zooi’, ‘zoon’, and ‘zoop’, as can be seen in Figure 3.

A picture containing text, screenshot, font, number

Description automatically generated

Figure 3 Ontology Prefixes

It defines a hierarchy of classes that represent different types of animals. Classes such as ‘animalc:Amphibian’, ‘animalc:Bird’, ‘animalc:Fish’, ‘animalc:Mammal’, and ‘animalc:Reptile’ provide a structured taxonomy of animals based on their biological characteristics and behaviours. It also incorporates logical constructs, such as equivalent classes and restrictions, to capture relationships and constraints among classes.

By using these constructs, the ontology ensures consistency and enables inferencing. For example, ‘animalc:Land’ in the ontology, representing land-dwelling animals states that this is equivalent to the intersection of two classes. One is defined as the complement of animals that have the property ‘zoop:is’ with the value "Airborne", which excludes airborne animals. The other is defined as the complement of animals that have the same property with the value "Aquatic", excluding aquatic animals. This definition ensures that ‘animalc:Land’ includes only animals that are neither airborne nor aquatic, implying they live on land.

A screenshot of a computer code

Description automatically generated with low confidence

Figure 4 animalc:Land construct

## SPIN Inferences

In this project, we used RDF and SPIN (SARQL Inference Notation) to create a knowledge base of animal classifications. Through seven queries, we made significant inferences about different aspects of the animal kingdom. We identified "Arthropod animals" by classifying certain animals as arthropods based on their membership in the "Insect" class.

Regarding thermoregulation, we categorized animals as "Warm-Blood" if they had "Feathers" or "Hair," a "Backbone," and belonged to a specific nurturing class. Similarly, "Cold-Blood" animals were identified based on the presence of "Fins" or their classification as "Reptile," "Amphibian," or a specific nurturing class.

We also created the inference of "Backbone animals" by classifying animals with "Feathers" or "Hair," a "Backbone," or "Fins," and belonging to relevant classes. Furthermore, we inferred "Land animals" by identifying animals in the "Animal" class without "Airborne" or "Aquatic" properties, categorizing them as land-dwelling. The inference of "No-Backbone animals" was created by identifying animals in the "Animal" class without a "Backbone," denoting those lacking a backbone.

Lastly, animals associated with the specific class of mammals were classified as "Mammal animals" if they belonged to the "Animal" class.

A screenshot of a computer code

Description automatically generated with low confidence

Figure 5 Warm-Blood animals query

Use of Wikidata & DBpedia

The integration of Wikidata and DBpedia has greatly enhanced the functionality and information presented on our website. Through our connection to DBpedia, we are able to enrich the data presented in the 'Inferences' tab by incorporating additional details and context from DBpedia's extensive knowledge base. This allows us to provide users with a more comprehensive understanding of the inferred classifications and characteristics of various animals. Additionally, our query to Wikidata enables us to retrieve relevant images associated with specific animal entries. By displaying these images alongside the textual information, we enhance the visual appeal of the website and provide users with a more engaging and immersive experience.

Application Features

Conclusions

The development of the new tab for our website has proven to be a resounding success. By incorporating inferences and leveraging data from external sources such as Wikidata and DBpedia, we have significantly enhanced the information presented to our users.

One of the key benefits of this new implementation is the utilization of SPIN, which has notably improved the speed of our queries. This enhancement has resulted in a slightly better user experience, allowing for faster retrieval and presentation of relevant data. While the impact may not be immediately noticeable due to the relatively small dataset we currently possess, it holds great potential for even greater efficiency and performance gains when applied to larger datasets.

By inferring additional knowledge based on existing data, we have been able to provide users with a more comprehensive understanding of the subject matter. This has enabled us to present a wider range of information, ensuring that our users can delve deeper into the topics they explore on our website.

In conclusion, the development of the new tab and implementation of inferences has been a success. Through the incorporation of inferences, integration of external data sources, and the utilization of SPIN, we have created a more informative and engaging experience for the users. While the immediate impact may be more pronounced with larger datasets, the foundations established lay the groundwork for even greater advancements in the future.

Application Configuration

## Requirements

Installation requirements to set the developed application up and running:

* Python (preferably 3.8.10 or higher)
* GraphDB
* s4api (pip install s4api)

## Creating the database

In GrapDB control panel, it is needed to create a database named “zoo” and import the provided “zoo.nt” N-Triples file.

Optionally, for the base url, [*http://zoo.org/*](http://zoo.org/)may be used.

## Running with PyCharm

To run the application with PyCharm, simply open the wsproject folder and press the run button. Then, a localhost link should appear in the console which needs to be opened with a web browser.

## Running with command line

For running the application using the command line, open a new command line in the “/wsproject/” directory and type the command “py manage.py runserver”. A localhost link should appear in the console which needs to be opened with a web browser.

References

Dataset

<https://www.kaggle.com/datasets/agajorte/zoo-animals-extended-dataset>

Slides

Representação do Conhecimento, Standards da Web Semântica, WS, DETI, UA

Representação do Conhecimento, A Linguagem SPARQL, WS, DETI, UA

Representação do Conhecimento, A *Triplestore* GraphDB, WS, DETI, UA

Web references

<https://docs.djangoproject.com/en/4.1/ref/forms/widgets/>

<https://docs.djangoproject.com/en/4.1/ref/forms/fields/>