KDE Project Report

Hugo Moura - 2283808, Panagiotis Andrikopoulos - 1780743,
Stylianos Psara - 2140527 Tomás Carrilho - 4324773, Ziv Hochman - 8454434

January 23rd, 2024

1 Introduction

In the domain of data engineering, the Semantic Web stands as a model that provides data integration and conceptualization. This report delves into a project that combines the world of Formula 1 (F1) and MotoGP, two big motorsports, with possible Semantic Web technologies applications. The primary objective is developing practical experience, using existing datasets from F1 and MotoGP. By doing so gaining understanding of the possibilities and limitations that this approach present, all within the Python programming environment is expected. Starting with building an integrated RDF Knowledge Base, with Blazegraph, graphs combination, providing statistics on the number of triples, resources, properties, axioms. Then, Leveraging RDF Schema and Web Ontology Language (OWL), ontological axioms will be presented defining distinct members within our motorsports knowledge graph. Linking knowledge base with prominent semantic repositories like DBpedia and Wikidata will be done exploring the challenges of semantic web, thereby enriching the depth of the project.To extract meaningful insights, SPARQL queries will be created, answering important questions related to motorsports organisation and statistics, incorporating advanced constructs such as property paths, negation, filters and ordering. Lastly, embedding and project limitations will be presented and challenges for the semantic web and future of generative AI idea will be covered.

2 Motivation

The motivation behind undertaking this project lies at the intersection of two captivating domains - the world of motorsports, represented by Formula 1 (F1) and MotoGP, and the world of Semantic Web technologies. The motorsports ecosystem generates a large data information from various sources such as race results, driver statistics, team dynamics and constructors statistics. Semantic web technologies offer clear linked data methods since the information is connected through standardized relationships. Also, it enables a shared understanding of data, making it easier for different platforms to communicate effectively. Learning and interpreting this diverse and volume data is an exciting challenge. RDF, RDF Schema, and OWL, provide a powerful means to define relationships, hierarchies, and constraints within the data. Exploring how these technologies can elevate representation providing comprehensive insights is captivating.Beyond theoretical exploration, this project provides an opportunity for hands-on. Bridging the gap between academic knowledge and practical application, giving a deeper understanding of the challenges and benefits associated with implementing semantic principles in real-world scenarios.

3 Datasets

3.1 F1 dataset

The Formula 1 data repository contains the sport statistics information from 1950 until 2023. The folder comprises organized CSV files, obtained from a public dataset accessible in Kaggle covering circuits, drivers, constructors, races, and seasons. The dataset includes race results (qualifying, sprints, main races), lap times, pit stops, race status, and standings. This structured information facilitates efficient report generation and insightful analyses of Formula 1 statistics.

3.2 Motogp dataset

So as to expand our graph, we decided to include MotoGP data obtained from a public dataset available in Kaggle. This dataset consists of a total of six csv files with MotoGP information from 1949 through 2022. From the data we can observe a variety of information about the drivers, emphasized in their positions through the years, the constructors and their years of contribution to the sport and finally the outcome of the grand prix events and the places that have been held on. Unfortunately, this dataset contains some inconsistencies.

4 Building an Integrated RDF Knowledge Base

4.1 Transforming the F1 dataset into RDF

In the process of transforming the Formula 1 dataset into RDF, the creation of namespaces is a fundamental step to uniquely identify and organize elements within the ontology. For F1, the integration of datasets, such as races with both seasons and circuits, status with sprint results and results (separately), is achieved by merging them based on common identifiers and shared attributes, such as racesID, circuitID and statusID, using the Python library pandas' functionalities. This method facilitates relationships among different entities and enables a representation of interconnected knowledge.

Then namespaces were created for each one of the dataset files, except for circuits, seasons and status given that these were merged into other files like mentioned previously. This and the fact that most of the dataset files have unique IDs, enables the creation of the desired structure for the RDF using a considerably streamlined approach. However, files such as Pitstops and Laptimes required a different approach, considering that such unique IDs were not directly obtained but instead created by concatenating the raceID with the driverID, and then respective data was assigned through two different processes:

- **Pitstops:** given that the maximum number of stops observed throughout the whole file was 6, it seemed to be more adequate to create predicates that explicitly stated which stop the corresponding values related to, for each one of the columns *lap*, *time* and *duration.*
- **Laptimes:** sequences were created in order to preserve the order of the values corresponding to each one of the laps, one for each one of the columns *position* and *milliseconds*, achieving a more compacted and straightforward data structure.

4.2 Transforming the MotoGP dataset into RDF

While handling MotoGP data, some entries were straightforward allowing us to directly add them as triple from the csv files. However, some csv proved more complicated which made the transformation of the data necessary. Take, for instance, the data in the Same_nation_podium_lockouts csv , which consists of information without an identifier (key) to use as subjects, leading us to create a new class MotoSportsEvents combining the track name with the class of the race. Additionally, since each event was assosiated with both the rider's nationality and the year, we chose to connect all three information using blank nodes. Moreover, in order to model the temporal information like the years that each constructor in MotoGP contributed to the world championship we decided to consider them as range and not as individual values. The csv consists of a total of 258 rows and 3 columns before the preprocessing. By adding a range of the years and a new column with the total number of the years that a constructor contributed in the world championship we manage to reduce significantly the size of the data into 36 rows and 4 columns.

4.3 Graphs combination

In order to connect both F1 and MotoGP graphs, we created the namespace http://example.org/ motor-sports/ as the unified prefix for both graphs. Moreover, to combine the information for both datasets, we employed OWL axioms to create classes and subclasses relationships for MotorSportsEvents, Constructors and Racers as follows:

- **MotorSportsEvents:** We unified the classes F1RaceEvents and MotoRaceEvents under the common class MotorSportsEvents.
- **Constructors:** The Classes F1Constructor and MotoGPConstructor were placed under the class Constructors which is presented as a subclass of dbo:Company. To capture the relationship between F1Constructor and MotoGPConstructor, we employ the OWL axiom 'unionOf'. This decision was made based on the fact that there are some constructors that contributes in both MotoGP and F1.
- **Racers:** The Driver and Rider were placed under the class Racer which is presented as a subclass of dbo:Person. To ensure clear distinction between the members of those two classes, we employed the OWL axiom "disjointUnionOf" since we didn't observe driver that are riders and vice versa.

4.4 Blazegraph

Given that our RDF graphs consist of large amounts of data, we stored our knowledge base in Blazegraph in order to perform more efficient and faster searches using SPARQL queries. After creating the TTL files that were originated from our code, these files were collected by a script (Store_graph_to_BlazeG.py) that executed a POST request and stored the files in Blazegraph. The reason we did not combine our TTL files into a single file is that storing a single large file in Blazegraph was time-consuming.

4.5 Statistics

Statistics	**Results**
Total number of Triples	1962632
Number of distinct resources created by us	116719
Total number of properties created by us	122
Total number of OWL axioms	156
Total number of RDFS axioms	5181
MotoGPConstructor triples after modeling temporal data	36 (from 258)
Total number of Classes	20

Table 1: Statistics

4.6 General Axioms

The subsequent step involves the creation of classes, subclasses, properties, predicates, and resources, which collectively define the structure and semantics of the ontology. OWL facilitates a more coherent and interconnected data. By utilizing OWL, ontologies can be constructed providing clarity, directness and understanding. Furthermore, the declaration of distinct members within a class, such as DriverID for F1 or Rider for MotoGP. In the context of RDF graphs, a more organized and meaningful representation of data is represented and is very important when dealing with big datasets. In table 2 and 3 is presented the OWL axioms and RDFS axioms. The result is then presented in turtle format in a ttl type file.

OWL AXIOMS
ms:Constructor owl:unionOf (re:MotoGPConstructor co:F1Constructor).
ms:Racers owl:disjointUnionOf (re:Rider d:Driver).
re:Rider owl:distinctMembers (re:GIACOMO_AGOSTINI re:VALENTINO_ROSSI ...).
ns1:MotorSportsEvents owl:equivalentClass [owl:disjointUnionOf (f1:F1RaceEvents, ns1:MotoRaceEvents)].
dbo:Person owl:equivalentClass dbo:Human.
ns1:MotoRaceEvents owl:disjointWith f1:F1RaceEvents.
ns1:Rider owl:disjointWith f1:Driver.
ns1:MotoRaceEvents owl:equivalentClass dbr:Grand_Prix_motorcycle_racing.
rc:hasCircuitName a owl:IrreflexiveProperty.
f1:Driver owl:distinctMembers [rdf:first d:1, d:10, d:100...].

Table 2: OWL axioms

RDFS AXIOMS
re:MotoGPConstructor rdfs:subClassOf ms:Constructors .
re:Rider rdfs:subClassOf ms:Racers.
ms:Racers rdfs:subClassOf dbo:Person.
ns1:HadFirstPlaces rdfs:domain ns1:Rider.
ns1:Held_On rdfs:range ns1:Track.
ms:Constructors rdfs:subClassOf dbo:Company.
ns1:MotoRaceEvents rdfs:subClassOf ms:MotorSportsEvents .
f1:F1Constructor rdfs:subClassOf ms:Constructor.
f1:F1RaceEvents rdfs:subClassOf ms:MotorSportsEvents .
f1:Driver rdfs:subClassOf dbo:Person.

Table 3: RDFS axioms

5 Linking knowledge with DBpedia and Wikidata

To expand our knowledge base, we connected the MotoGP Tracks with the DBpedia and the MotoGP Riders with both DBpedia and WikiData using the property rdfs:seeAlso. The connection with DBpedia was relatively straightforward since the DBpedia URI for each rider is in the form dbo:riderName_riderSurname (e.g., dbo:Valentino_Rossi). Therefore, we did not need a specific query to obtain the riders' names as they were already present in our data. On the other hand, connecting the MotoGP Riders with Wikidata was more challenging because we needed the QID for each rider, as the URI for each rider in Wikidata is of the form http://www.wikidata.org/entity/QID, for example, the URI for the rider Valentino Rossi is http://www.wikidata.org/entity/Q169814. To obtain the QID and create the URIs needed for each rider, we developed a script (Retrieve_WikiData_Links.py) that executes SPARQL queries on Wikidata, searching for the QID based on the name and surname of each rider. After retrieving the required information, the script generates the URIs for each rider and stores them along with their names in a CSV file (wikidata_results.csv). This CSV file can be utilized later to link this data with our graph. Unfortunately, information was available for only 233 out of the 395 riders in Wikidata.

6 SPARQL queries

By adopting Semantic Web principles, we aim to turning our data the most accessible as possible, creating queries. In order to achieve this goal SPARQL will be pivotal, our objective is to

formulate queries that explain the complexities of F1 data.
One of many queries zoom in on the winners of qualifying sessions in Formula 1. These queries help us figure out how many times a driver nailed the top spot in qualifying during a specific season. Furthermore, we also delve into the world of sprint race winners throughout all Formula 1 seasons, not forgetting to specify the exact times the drivers completed those sprints. For the elaboration of these queries we deal with prefixes, filters, orders, concatenation and counting functions to achieve the best results as possible, also presenting them effectively.
Shifting our focus, we extracted information about the drivers who represented a constructor from their own country in Formula 1, along with the corresponding years of their collaboration. Here we demonstrate an exploration of the relationships between drivers and constructors within the Formula 1 ecosystem using four different sources or prefixes.

7 Embedding

In this project, where the analysis and processing of the organized data are the primary goals, the utilization of embedding becomes crucial. This provides valuable insights into how well the model captures relevant information and makes accurate predictions. To achieve this, the TransE class was implemented with the TorchE library, employing a neural network as the machine learning model.
The workflow initiates by converting the TTL file into triples, which are usable data for Python. These triples represent the relations between entities in the data, enabling the computation of the number of unique entities and relations (specifically, in the Formula 1 dataset, where there are 272,041 unique entities and 190 unique relations). Given that the majority of the data pertains to Formula 1, the focus is primarily on this domain. Nevertheless, to facilitate the training and evaluation processes within the TransE class, the need to convert these triples into tensor format arose. Subsequently, training the model with the desired format and evaluating it using two useful metrics—AP (Average Precision, where a higher AP indicates a better trade-off between recall and precision) and Hits@k (the proportion of correctly predicted items in the top k ranked list, where a higher Hits@k indicates better performance)—reveals an accuracy of 72.9 percent for the model. Furthermore, we employed custom queries to observe the predictions generated by our model. Initially, we provided the query (co:27, s_res:status, ?) to determine its predictions then, we examined the anticipated output, and the highest-scoring result we obtained was ”Turbo.”

Evaluation	Results
Average Precision (AP)	0.7295
Hits@1	0.1111
Hits@3	0.3333
Hits@5	0.5556
Hits@10	1.0000

Table 4: Evaluation results

8 Limitations

8.1 Limitations of the RDF data model and semantic web technologies

The RDF data model and semantic web technologies encounter some limitations related to modeling relationships, human interpretability, and the evolution of data. Representing relationships in RDF triples, especially those involving complex structures, can be challenging and may require some abstraction. Human interpretability is another concern, as the semantics web envolves a machine understanding that may not be evident for humans language and thinking. Additionaly, adapting to the everyday progress in data and accommodating changes in ontologies can become a huge challenge, requiring careful management of updates to ensure the continued accuracy of the semantic representations. In the context of our project, the information remains accurate only up to the MotoGP and Formula 1 seasons of 2022/2023.

8.2 Overall Limitations

Query performance and the complexity of the triple structure, with subject-predicate-object, are important aspects when working with RDF data. The triples offer a flexible approach to data representation however, this flexibility can introduce challenges related to query performance using Python. To reduce the number of triples generated during transformation, several optimization tricks and methods were employed such as filtering selective information in the dataset, aggregating and merging data when possible. As the volume of RDF triples grows, the complexity of queries increase, leading to longer processing times. The solution for that problem was the use of the Blazegraph environment, resulting in process times much lower than the ones we encounter using Python. In the process of formulating queries, we encountered some limitations from the nature of the available data in the Turtle format and files. The queries necessitated an exploration of the data, and in certain instances, the provided information proved insufficient for addressing specific inquiries related to race outcomes. Notably, the constructor data lacked specificity, particularly concerning season winners. Addressing this gap, with more time available for the dedication of this project, the creation of a new file focused on season winners could provide valuable insights into the performance of constructor teams. Additionally, challenges arose when integrating information from two distinct sources, MotoGP and Formula 1. Specifically, differencies in the representation of nationalities posed issues, with MotoGP using abbreviations (e.g., IT for Italy) and Formula 1 employing the full names (e.g., Italian or Italy). This divergence in data representation presented challenges for queries attempting to reconcile these variations, however, given more time, we could incorporate an additional preprocessing step to combine these two representations into a unified format.

9 Conclusion

9.1 What are the challenges for the semantic web? Future in the era of generative AI?

In the course of discussions with a professional in the field, Tiago Mota, Tech manager at Deloitte, we can affirm that the Semantic Web holds the promise of enhancing data value by fostering a more interconnected and comprehensible digital landscape. This, in turn, empowers machines to derive meaning from data, elevating the efficiency and accuracy of search engines. However, challenges persist, including the need for standardized data representations, interoperability across diverse sources and formats, particularly unstructured data. Overcoming resistance to adopting new technologies and transforming work methodologies presents a substantial obstacle, requiring time and effort. Critical aspects include the development of robust data management mechanisms and achieving (near) real-time updates of semantic data to advance knowledge and ensure timely representations. Looking ahead, the Semantic Web's future intersects with the era of generative AI, offering opportunities for even more sophisticated and intelligent information processing.

9.2 Gitlab repository

Link: https://git.science.uu.nl/group-7-kde/csv_files_analysis/

10 Tasks distribution

Hugo Moura

- Exploration of F1 dataset and sharing motorsport knowledge with the group members.
- Transforming F1 dataset into triples, starting stage, creation of Namespaces, classes, literals and properties for some F1 csv files and OWL axioms creation.
- SPARQL queries creation for F1.
- Reporting, project limitations and interview with a professional working on SWT.

Stylianos Psara (KDE_MOTO_GP_S_)

- Focused mostly on the MotoGP dataset and information.
- Transforming some of the MotoGP datasets into triples and creating a complete rdf graph, and apply RDFS and OWL axioms.
- Adding axioms in order to connect information between FormulaOne and MotoGP.
- SPARQL queries creation for F1 and MotoGP.
- Statistics
- Reporting

Panagiotis Andrikopoulos

- Focused mostly on the MotoGP dataset and information.
- Transforming some of the MotoGP datasets into triples and creating a complete rdf graph, and creating RDFS and OWL axioms.
- Adding axioms in order to connect information between FormulaOne and MotoGP.
- SPARQL queries creation both for F1 and MotoGP.
- Link Motogp graph with DB pedia and Wikidata (Tracks and Riders)
- Store our knowledge base to Blazegraph
- Reporting

Tomás Carrilho

- Exploration of F1 dataset and information.
- Transforming F1 dataset into RDF, by assembling the needed structure, triples, properties, classes and axioms
- Reporting

Ziv Hochman

- Implementing and evaluating the Embedding graph
- SPARQL queries creation for F1
- Reporting

The group project had a very dynamic collaboration, with organised tasks and efficiently distributed among team members. The coordination and well-structured organization of responsibilities within the group contributed to the project's success. We had an inclusive and collaborative environment, ensuring that each member's strengths were leveraged. Effective communication and positive environment were key throughout all our project.