EVO: An Ontology for the Field of Electric Vehicles

Ioannis Kivrakidis[1][0009-0000-3029-8678], Emmanouil S. Rigas[2][0000-0002-8042-9135], and Nick Bassiliades[1][0000-0001-6035-1038]

[1] School of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece. `{ikivrakid,nbassili}@csd.auth.gr`
[2] School of Medicine, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece. `erigas@auth.gr`

Abstract. The global automotive industry, a thriving economic powerhouse and epicenter of research and development, is increasingly focused on advancing safety for both vehicle occupants and pedestrians. While the rising number of vehicles offers unprecedented mobility and convenience, it has also led to alarming urban air pollution levels. To combat this, many countries are actively promoting the adoption of Electric Vehicles (EVs). EVs are hailed for their remarkable efficiency in reducing oil consumption and gas emissions, positioning them as a promising solution for future road transportation. However, the EV landscape is marred by substantial diversity among manufacturers, particularly concerning battery capacities and types, range of vehicles and charger types. This lack of standardization has created a pressing need for advanced data and knowledge management methods. In response, this paper focuses on the development of an ontology tailored specifically to the domain of EVs. This ontology aims to enhance reusability and interoperability, fostering seamless information, integration and collaboration across various EV-related applications and systems. By doing so, it propels the EV industry toward greater improvement in the years ahead. The framework that was utilized for the development of the EV Ontology (EVO) was Protégé, employing the Web Ontology Language (OWL) for standardized knowledge representation. The competence of the EVO is evaluated through SPARQL queries. Furthermore, the ontology is enriched by a Python script that scrapes data from web-based EV databases. Finally, a user interaction Python script is implemented that lets users interact with the ontology with personalized SPARQL queries.

Keywords: Electric Vehicle · Ontology · Semantic Web · EV ontology · Knowledge Representation · EVO.

1 Introduction

The rising prevalence of internal combustion vehicles using non-renewable traditional fuels has led to concerns related to both energy resources and the environment. Consequently, many nations have introduced new energy vehicles

(NEVs) as substitutes for conventional automobiles to decrease reliance on oil and mitigate air pollution associated with traditional vehicles. NEVs utilize non-conventional energy sources to power vehicles encompass electric vehicles, hydrogen vehicles, natural gas vehicles, methanol vehicles, and ethanol vehicles. Among them, Electric Vehicles -EVs- are widely regarded as the most effective in achieving environmental and socioeconomic benefits [4].

As a technology that emerged after the industrial revolution [13], EVs have been in existence for over a century, with the first practical EV to be invented by Thomas Parker in 1884. In the 1920s, before Henry Ford introduced the Model T with his revolutionary mass production process, EV manufacturers enjoyed a degree of success. In fact, EVs made up 28% of the total vehicles produced in the US during that period [14]. However, the promotion of EVs slowed down due to their high cost and the rapid advancements in conventional vehicles. It wasn't until the beginning of the 21st century, driven by concerns about environmental pollution and energy-related issues, that research and development of EVs regained momentum. With the involvement of both government and industry, there has been significant improvement in EV infrastructure and technology [5]. Indeed, the number of EVs in Europe [2] is expected to increase from about three-quarters of a million in 2019 to more than four million in 2025 [12], [8].

Renowned automakers such as Volkswagen, Mercedes, and Ford have also expressed their commitment to promoting EVs and new companies focusing solely to EVs such as Tesla have emerged. EVs are primarily categorized into five types: battery electric vehicles (BEVs), hybrid electric vehicles (HEVs), plug-in hybrid electric vehicles (PHEV), extended-range electric vehicles (ER-EVs) and fuel cell electric vehicles (FCEVs) [10].

Given the profound implications of EVs in addressing energy sustainability and environmental conservation, the need for advanced data and knowledge management methods has never been more pronounced. Consequently, ontologies [3] have become a potent tool for establishing mutually agreed-upon definitions that encapsulate the key concepts within a domain of knowledge, facilitating their reuse. This paper, therefore, aims to develop a dedicated ontology for the domain of EVs. This ontology was carefully crafted to improve the ability to use and share data and knowledge within the EV sector. This, in turn, promotes smooth integration and cooperation among different EV-related applications and systems. To realize this goal, OWL has been employed for defining the ontology itself, while SPARQL serves as the key tool for information retrieval. The development and maintenance of the ontology are facilitated by the utilization of Protégé framework.

2 Related Work

The EV industry has emerged as a significant area of research interest, leading to numerous studies aimed at addressing various challenges associated with the electrification of transportation. Given the complex and interconnected nature of the EV sector, researchers have recognized the need for interdisciplinary approaches and the integration of diverse data sources from various domains. In response to this challenge, ontologies have been developed to facilitate semantic interoperability, allowing seamless communication and data sharing between diverse sources in the EV sector. One of the works done in this domain is the research by Qi et al. [11], in which the authors developed an EV-centric Knowledge Graph (EVKG) as a comprehensive and extensible knowledge management system to effectively handle the complex and diverse EV industry. The EVKG integrates essential EV-related knowledge, such as EV adoption, charging infrastructure and electricity transmission networks, from various sources, and it employs ontologies for semantic interoperability with other knowledge graphs. In addition to the EVKG, researchers have also recognized the value of an existing ontology designed to capture concepts related to vehicles, such as the Vehicle Ontology.[3] This ontology distincts from the EVKG, as it offers a specialized understanding of vehicle-related concepts. Nouicer et al. [1] introduced another ontology within the EV domain, aiming to support E-Mobility applications. Initially, they applied this ontology in a trip planning application, intending to aid EV drivers in mapping out their journeys by pre-booking charging points (CPs) to alleviate range anxiety. Their ontology reuses and extends existing vocabularies like the Smart Applications REFerence ontology (SAREF), encompassing key concepts related to EVs (e.g., EV, Battery, EVSE, Charging Station).

In this paper, these foundational ontologies formed the basis for developing an advanced Electric Vehicle Ontology, abbreviated as EVO. The main differentiation of EVO is that it focuses extensively into the concepts of EVs, offering detailed taxonomies, specifications, and technical attributes. Furthermore, it expands into areas such as Charging Stations, Batteries, Engines, Connectors and Charging Modes offering a more comprehensive view of the EV landscape. This, in turn, makes it a valuable resource for researchers, policymakers, and industry stakeholders e.g., EV owners, Charging Stations (CS), grid suppliers, payment and third-party services providers seeking in-depth knowledge and insights into the EV domain. To the best of our knowledge, EVO is the first ontology fully focusing on all the main aspects of the EV domain. This advancement is a significant step in enhancing our understanding of the intricate and evolving field of electric vehicles.

3 Electric Vehicle Ontology

In this section, the Electric Vehicle Ontology - EVO,[4] as well as a web scraping tool for automatic creation of instances are described. The proposed EV ontology serves as a structured and systematic framework for organizing and categorizing various concepts, properties and relationships related to EVs. It also provides a foundation for understanding and classifying information considering EV components, charging infrastructure, energy sources, regulatory and

---

[3] https://enterpriseintegrationlab.github.io/icity/Vehicle/Vehicle_1.2/doc/index-en.html

[4] https://w3id.org/evo

propulsion aspects, as well as several environmental impacts. By offering a standardized structure and common language, this ontology enables data integration, semantic web applications, and knowledge sharing in EVs. More specifically, this paper provides an in-depth exploration of all the components and structure of the proposed EV ontology.

3.1 Description Of The EVO

An ontology is a formal description of knowledge as a set of concepts within a domain and the relationships that hold between them. To enable such a description, it is necessary to formally specify components such as individuals (instances of objects), classes, attributes, and relations[7]. For the domain of EVs, we introduced EVO. EVO consists of 20 classes, 17 object properties and 54 datatype properties. Figure 1, depicts the class hierarchy of this ontology. Additionally, Table 1 contains all the classes that consist the EVO, as well as their detailed description. The ontology has a set of properties. The EV class details aspects such as manufacturer, model type, battery specifications, model year, CO2 emissions, electric motor details, electric range and performance specs. BEV properties focus on attributes specific to battery electric vehicles, including connector type for charging interoperability. HEV properties encompass characteristics like internal combustion engine (ICE), fuel type, and fuel tank specifications. EREV class focuses on EVs with a range-extending ICE. FCEV properties include fuel cell output and number of fuel tanks. EVSE properties describe charging infrastructure features like bidirectional support and compatibility number of charging ports and charging rate. Manufacturer properties relate to EV producers' metrics. ChargingStation properties define attributes of charging stations, while ChargerCollection properties detail charger groupings. ConnectorType properties describe charging connector attributes, and ChargingModes properties specify charging mode details. The list of properties for each class is descerибed in detail in Table 2.

Class	**Description**
EV	An EV provides a means of transportation within the urban system. EVs are powered by electric motors and use rechargeable batteries as their primary source of propulsion.
BEV	A BEV (Battery Electric Vehicle) is an EV solely powered by an electric battery and does not rely on any ICE or fossil fuels for propulsion. The vehicle's wheels are driven by one or more electric motors, drawing power from the battery.
HEV	An HEV (Hybrid Electric Vehicle) is a type of vehicle that combines an ICE with an electric propulsion system. Unlike BEVs, which rely solely on electric power, HEVs use both the ICE and the electric motor to drive the vehicle. HEVs do not have a charging port for external charging like BEVs or Plug-in Hybrid Electric Vehicles (PHEVs)

PHEV	A PHEV (Plug-in Hybrid Electric Vehicle) is a type of HEV that combines both an ICE and an electric propulsion system with a larger battery pack. The key feature that distinguishes PHEVs from regular HEVs is their ability to be charged externally by plugging them into an electric power source.
EREV	An EREV (Extended Range Electric Vehicle) is a type of EV that uses an electric propulsion system, including a battery and an electric motor. The distinguishing feature of EREVs is the fact that they have an ICE on board solely for electricity generation when the battery is depleted. EREVs are designed to offer extended electric driving ranges, with the ICE acting as a generator to recharge the battery, making them more versatile than traditional EVs.
FCEV	An FCEV (Fuel Cell Electric Vehicle) is a type of EV that uses a fuel cell to generate electricity onboard by combining hydrogen (or other fuel) with oxygen from the air. This electricity powers an electric motor, which propels the vehicle. The only byproduct of this process is water vapor, making FCEVs a potentially environmentally friendly option.
BatteryTypes	BatteryTypes is the class that refers to the various kinds or categories of batteries that are used to store and supply electrical energy to EVs. There are various battery technologies, each with its own characteristics, advantages, and limitations.
BatteryRelation	BatteryRelation is the class that serves as a connector or intermediary that links the EV (e.g., Tesla Model_X) to its battery (e.g., Lithium Ion) and allows for the representation of additional information about this pairing.
EVSE	EVSE is the class representing the various types of chargers available, each with specific characteristics and capabilities. These types of chargers are used to charge BEVs and PHEVs.
ModelType	ModelType is the class that represents individual vehicle models (e.g., Model_X).
Manufacturer	Manufacturer is the class that represents the entities that produce vehicles, batteries, engines etc. These are organizations or companies responsible for designing, producing, and distributing vehicles to the market.
ChargingStation	ChargingStation is the class that represents a facility where EVs can be charged. Charging stations are essential for recharging the batteries of EVs and are a crucial part of the electric vehicle ecosystem.
ChargerCollection	ChargerCollection is the class that represents a group of chargers or EVSEs that are hosted by a charging station.

EngineTypes	EngineTypes is the class that represents various categories or types of engines used in vehicles.
EngineRelation	EngineRelation is the class that serves as a connector or intermediary that links the EV (e.g., Tesla Model_X) to its engine (e.g., permanent magnet synchronous) and allows for the representation of additional information about this pairing.
ElectricEngine	ElectricEngine is the class that serves as a connector or intermediary that links the EV (e.g., Tesla Model_X) to its electric engine (e.g., permanent magnet synchronous) and allows for the representation of additional information about this pairing.
ICE	ICE is the class that serves as a connector or intermediary that links the EV (e.g., BMW X5 G05 45e) to its internal combustion engine and allows for the representation of additional information about this pairing.
RangeExtenderICE	RangeExtenderICE is the class that serves as a connector or intermediary that links the EV to its range extender or auxiliary engine and allows for the representation of additional information about this pairing.
ConnectorType	ConnectorType is the class that represents the various types of connectors used for charging EVs. Each instance of this class corresponds to a specific connector type, such as CCS, Type 1, Type 2, etc.
ChargingModes	ChargingModes is the class that represents the various charging modes that are available. These modes differ in terms of safety, speed and the connector. This class is designed in accordance with the IEC 62196-2 standard.

Table 1: Classes of Electric Vehicle Ontology.

3.2 Data Import in the EVO

Web scraping and the Cellfie plugin were used in order to automate the process of importing the data into the proposed EVO. More specifically, web scraping [6] entails the extraction of information from the internet, employing a web scraper as the designated tool for this purpose. Python is known for its scripting simplicity, as it offers an extensive range of libraries tailored for web scraping. This feature makes Python an optimal programming language for such a task, which only requires a limited amount of code. Conversely, the Cellfie[5] plugin is a Protege plugin designed for importing spreadsheet data into OWL ontologies. The goal is to seamlessly integrate these technologies to establish a standardized

---

[5] https://github.com/protegeproject/cellfie-plugin

Class	**Properties**
EV	hasManufacturer, hasModelType, hasBattery, hasModelYear, hasCO2Emissions, hasElectricMotor, hasElectricRange, hasSystemHorsePower, hasSystemTorque, hasZeroToHundred, hasMaxSpeed
BEV	hasMatchableConnectorType
HEV	hasICE, hasFuelType, hasFuelTank
PHEV	
EREV	hasREICE
FCEV	fuelCellMaxOutput, hasMaxFillingPressure, hasNumberOfTanks
BatteryTypes	hasSpecificEnergy, hasEnergyDensity, hasSpecificPower, hasLifeSpan, hasCostPerKWH
BatteryRelation	hasBatteryType, hasCapacity, hasNumberOfCells, hasVoltage
EVSE	supportsBidirectionalCharging, hasMaxChargingRate, hasEVSEPorts, hasMode, hasEVSEPorts, hasCompatibilityWith
ModelType	hasLevelOfAutomation, hasDriveWheel
Manufacturer	produces, hasTotalEVSales, onlyProducesEVs
ChargingStation	hosts, renewableSources, hasZipCode, hasStreetNumber, hasStreet, hasLongitute, hasLatitude, hasCountry, hasCity
ChargerCollection	isHostedBy, hasEVSE
EngineTypes	
EngineRelation	electricEngine, internalCombustionEngine
ElectricEngine	hasElectricPower
ICE	hasDisplacement, hasNumberOfCylinders, hasHorsePower, hasCompressionRatio
RangeExtenderICE	hasRangeExtensionCapability, providesPropulsion
ConnectorType	isUsedInCountry, hasOutputCurrentType, hasNumberOfPins, hasMaximumPowerOutput, hasMaximumOutputVoltage, hasMaximumOutputCurrent
ChargingModes	hasPowerRating, hasMaxCurrent, hasPhase, hasMaxVoltage, hasSpecificConnector

Table 2. properties for each class of Electric Vehicle Ontology.

Fig. 1. Class hierarchy of the EVO.

method for importing data from various web pages related to EVs into EVO. Each component is presented in one of the two following parts. The first part focuses on describing the Web Scraping algorithm that utilizes Python libraries and Xpath to extract data from the web. Data is then processed, cleared, and organized into a structured spreadsheet. The second part is dedicated to the Cellfie plugin, which takes the spreadsheet as input. This plugin employs transformation rules written in Manchester syntax to map the data from the spreadsheet onto the ontology. Finally, once the rules are established, the generated data is imported into the ontology, completing the technical process of integrating web-scraped information into the specified ontology.

Web Scraping: The Python script begins by importing necessary libraries, including html from lxml, requests, and pandas as pd. It then makes an HTTP request to retrieve the page content from the EV Database website[6]. Using the lxml library, the script parses the HTML content into a tree structure. XPath queries are employed to extract multiple URLs related to EVs from the website. The script ensures a careful approach by looping through a subset of the collected EV URLs in each execution to avoid triggering anomaly detection algorithms on the website. This precautionary measure reduces the risk of facing a

---

[6] https://ev-database.org/

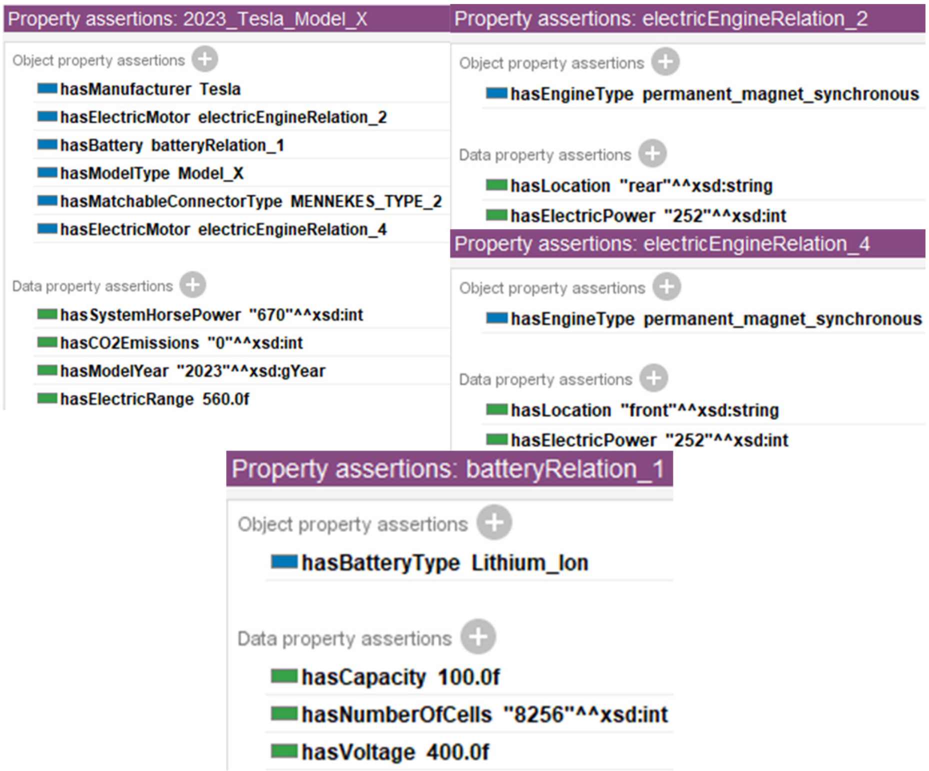
permanent denial of service from the server due to algorithm triggers. Within the loop, additional HTTP requests are made for individual EVs, and XPath queries are utilized to extract performance, battery, and charging data from each EV's page. The script employs a function to convert the extracted data, initially in list format, into separate DataFrames for performance, battery, and charging information. These DataFrames are then concatenated along with appropriate keys ('Performance', 'Battery', 'Charging') and transposed for better organization. The aggregated data from each EV is appended to a final DataFrame. Finally, the script names the columns of the final DataFrame based on the headers of the original data. The aggregated data is saved to an Excel file. This script effectively automates the extraction and organization of EV data from the website into a structured Excel file.

Cellfie plugin: Cellfie, is a Protégé plugin, that facilitates the integration of spreadsheet data into OWL ontologies. To initiate this process, an Excel file is generated from the Web Scraper and subsequently imported into Protégé, following the loading of the EVO. The integration begins from the Tools menu, specifically by selecting "Create axioms from Excel workbook". Within the Open File window, an Editing panel for transformation rules becomes available. This panel empowers users to add, modify, or remove rules, offering a flexible environment for rule management. Additionally, the panel allows the user to save and load their current set of rules, ensuring reusability and adaptability. In the context of a set of instances, a set of rules is crafted to transform the imported Excel file into instances within the ontology. Once the transformation rules are finalized, the "Generate Axioms" button, positioned at the bottom of the window, becomes active. Clicking this button triggers Cellfie to autonomously generate OWL axioms and Individuals based on the defined rules, providing users with a preview of the outcome. Users are presented with two options for importing the newly generated axioms: they can either import them into a new ontology or integrate them directly into the currently open ontology.

3.3 Named Individuals

In the previous subsection, a detailed process of importing data or individual instances of classes into the ontology was presented. These instances, when combined with the EVO, collectively form the EV knowledge base. To illustrate the accuracy of the modeling an example instance of the BEV class is shown in Figure 2.

The first panel, labeled "Property assertions: 2023_Tesla_Model_X", serves as a good example of a BEV instance. This instance comprises two types of properties: Object Properties (highlighted in blue) and Data Type Properties (highlighted in green). Data Type Properties establish relationships between individuals and literal data, such as strings, numbers, and datetimes. In this case, the Data Type Properties link the individual "2023_Tesla_Model_X" to data related to its performance, CO_2 emissions, range, etc. On the other hand, Object Properties establish relationships between individuals and other individuals. In this instance, these properties link the individual "2023_Tesla_Model_X" with other individuals, such as “electricEngineRelation_2” (signifying the rear engine), “electricEngineRelation_4” (signifying the front engine), “batteryRelation_1” (detailing specifications of its battery), “MENNEKES_TYPE_2” (indicating its connector type), “Tesla” (manufacturer), and “Model_X” (its model designation).



Property assertions: 2023_Tesla_Model_X

Object property assertions

hasManufacturer Tesla
hasElectricMotor electricEngineRelation_2
hasBattery batteryRelation_1
hasModelType Model_X
hasMatchableConnectorType MENNEKES_TYPE_2
hasElectricMotor electricEngineRelation_4

Data property assertions

hasSystemHorsePower "670"^^xsd:int
hasCO2Emissions "0"^^xsd:int
hasModelYear "2023"^^xsd:gYear
hasElectricRange 560.0f

Property assertions: electricEngineRelation_2

Object property assertions

hasEngineType permanent_magnet_synchronous

Data property assertions

hasLocation "rear"^^xsd:string
hasElectricPower "252"^^xsd:int

Property assertions: electricEngineRelation_4

Object property assertions

hasEngineType permanent_magnet_synchronous

Data property assertions

hasLocation "front"^^xsd:string
hasElectricPower "252"^^xsd:int

Property assertions: batteryRelation_1

Object property assertions

hasBatteryType Lithium_Ion

Data property assertions

hasCapacity 100.0f
hasNumberOfCells "8256"^^xsd:int
hasVoltage 400.0f

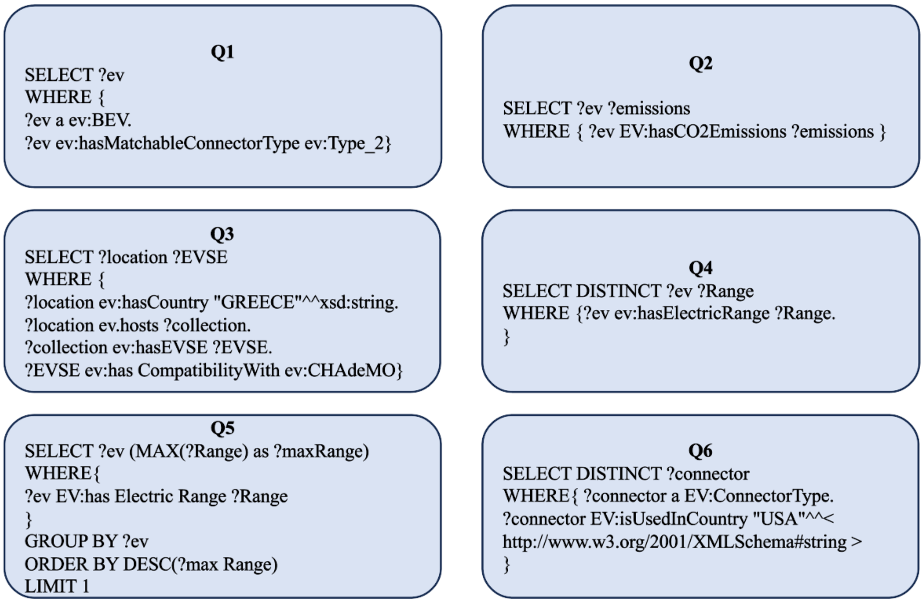
Fig. 2. Example instance of the BEV class.

4 SPARQL Queries

This Section focuses on the process of crafting SPARQL queries [9] from user stories. User stories capture the “what” from the user’s viewpoint, SPARQL queries focus on the “how” in terms of navigating and querying the relationships and properties defined within the EVO. Throughout this chapter, the goal is to guide through the process of translating user stories into effective SPARQL queries. By doing so, the aim is to harness the potential of the EVO in representing and querying complex knowledge structures related to EVs and assessing the competence of the proposed ontology. The user stories implemented in our study are the following.

- Q1: As a user looking for a BEV, I want to find BEVs that are compatible with the Type of connector, i.e., Type 2.
- Q2: As a consumer, I want to understand the environmental impact of EV.
- Q3: As an EV owner, I want to find charging stations in my country, i.e., Greece, that have compatibility with the type of connectors, i.e., CHAdeMO.
- Q4: As a potential EV costumer, I want to understand the driving range capabilities of various EV models.
- Q5: As a potential EV costumer, I want to know which specific EV model has the maximum electric range.
- Q6: As a potential EV costumer, I want to know which specific connector type is used in a country.

The equivalent Sparql queries that have been developed to cover each of these user stories are depicted in Figure 3



Q1

```
SELECT ?ev
WHERE {
?ev a ev:BEV.
?ev ev:hasMatchableConnectorType ev:Type_2}
```

Q2

```
SELECT ?ev ?emissions
WHERE { ?ev EV:hasCO2Emissions ?emissions }
```

Q3

```
SELECT ?location ?EVSE
WHERE {
?location ev:hasCountry "GREECE"^^xsd:string.
?location ev.hosts ?collection.
?collection ev:hasEVSE ?EVSE.
?EVSE ev:has CompatibilityWith ev:CHAdeMO}
```

Q4

```
SELECT DISTINCT ?ev ?Range
WHERE {?ev ev:hasElectricRange ?Range.
}
```

Q5

```
SELECT ?ev (MAX(?Range) as ?maxRange)
WHERE{
?ev EV:has Electric Range ?Range
}
GROUP BY ?ev
ORDER BY DESC(?max Range)
LIMIT 1
```

Q6

```
SELECT DISTINCT ?connector
WHERE{ ?connector a EV:ConnectorType.
?connector EV:isUsedInCountry "USA"^^<
http://www.w3.org/2001/XMLSchema#string >
}
```

Fig. 3. SPARQL queries derived from user stories.

5 Menu Interface For Querying The EVO

This Section focuses on the creation of a Python script that implements a user interface. In this script the user can interact with an RDF graph containing information about EVs. The RDF graph is parsed from an RDF file (EVO.ttl[4]) which is the EVO using the rdflib library. After the parsing of the ontology into a graph the script provides a menu-driven interface for the user to perform various queries on the EV dataset. Overall, the script serves as an interactive tool for users to explore and retrieve information from the EVO, ranging from environmental impact to technical specifications, using various SPARQL queries that were presented in the previous Section. The menu-driven approach makes it easy for users to select and obtain the specific information they are interested in. A detailed explanation of how the algorithm works follows.

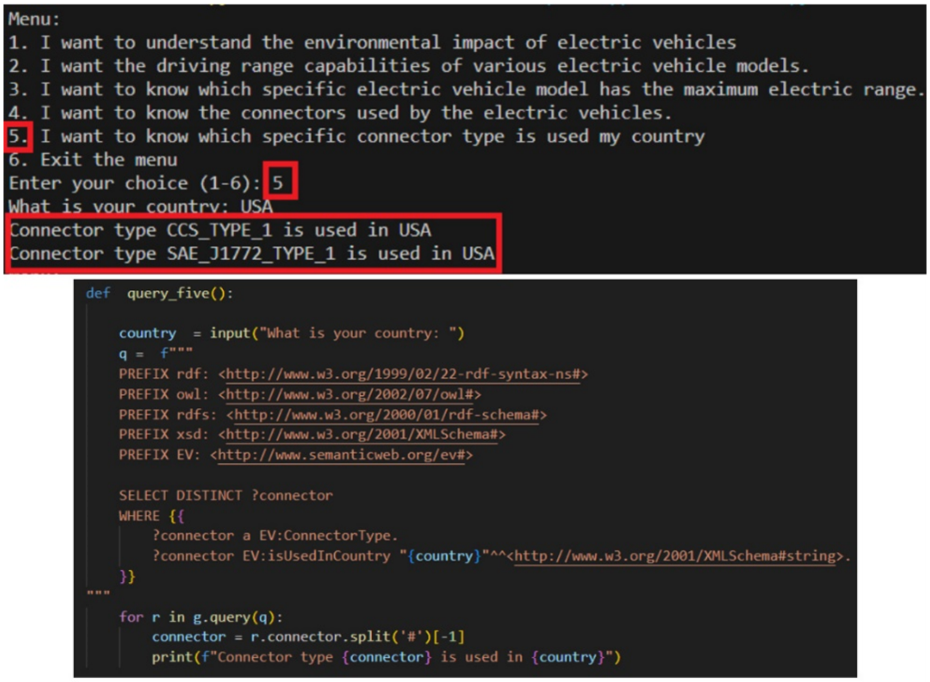
The algorithm begins by importing the necessary libraries and modules, including the rdflib library for working with RDF data. Subsequently, a graph is created, and RDF data from the EVO is parsed into the graph using the parse method. The script's menu feature is implemented as a while loop, offering users a range of options from 1 to 6. Each option corresponds to a specific query that the user desires to retrieve answers for. The user interacts with the menu by entering a number, triggering the execution of the corresponding query function (Figure 4). All query functions follow a consistent pattern. They store a SPARQL query in a variable and use the query() function on the graph to iterate through the results. The fifth query function, specifically designed for user customization, accepts an additional input field. This input allows users to modify the existing query according to their specifications, accomplished through formatted strings. During a demo run, the user chooses option 5, focusing on specific connectors used in their country of interest. After inputting the country as "USA", the query retrieves all types of connectors used in the USA. Subsequently, the user inputs the number 6 to exit the menu and terminate the program. This script provides a user-friendly interface for querying information from the EVO. The flexibility of query customization enhances the script's usability, making it a powerful tool for ontology exploration.

6 Ontology Evaluation

To assess EVO, we utilized the OOPS! scanner [7] along with the Pellet reasoner. OOPS! serves as an online evaluation tool for ontologies, pinpointing potential pitfalls across three categories: Critical, Important, and Minor. Following the evaluation, minor issues and one important were highlighted. The minor issues encompassed the absence of human-readable annotations (e.g., rdfs:label), missing inverseof relationships, and the usage of varied naming conventions (e.g., CamelCase and delimiters). The absence of readable annotations is mitigated by comprehensive documentation accompanying the ontology, detailed within this paper. We've also taken into account the tool's suggestions regarding missing inverseof relationships, in response we intend to incorporate them in a subsequent version. Furthermore, instances of different naming conventions will be rectified. The sole important pitfall was about the lack of disjoint axioms, although not affecting ontology functionality. A few axioms addressing this concern will be

---

[7] https://oops.linkeddata.es/



```
Menu:
1. I want to understand the environmental impact of electric vehicles
2. I want the driving range capabilities of various electric vehicle models.
3. I want to know which specific electric vehicle model has the maximum electric range.
4. I want to know the connectors used by the electric vehicles.
5. I want to know which specific connector type is used my country
6. Exit the menu
Enter your choice (1-6): 5
What is your country: USA
Connector type CCS_TYPE_1 is used in USA
Connector type SAE_J1772_TYPE_1 is used in USA
```

```
def  query_five():

    country  = input("What is your country: ")
    q =  f"""
    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX owl: <http://www.w3.org/2002/07/owl#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
    PREFIX EV: <http://www.semanticweb.org/ev#>

    SELECT DISTINCT ?connector
    WHERE {{
        ?connector a EV:ConnectorType.
        ?connector EV:isUsedInCountry "{country}"^^<http://www.w3.org/2001/XMLSchema#string>.
    }}
"""
    for r in g.query(q):
        connector = r.connector.split('#')[-1]
        print(f"Connector type {connector} is used in {country}")
```

Fig. 4. Menu interface with code demo.

integrated into the next version. Lastly, the Pellet reasoner revealed no inconsistencies and functioned seamlessly.

7 Conclusions

In this work, an examination of current EV technologies is undertaken. The analysis commences with a systematic classification of EVs into distinct categories, followed by an in-depth examination of EV battery characteristics and connector types. The exploration extends to EV infrastructure, covering charging stations and charging modes. This comprehensive examination of EV technologies serves to provide a cohesive understanding of the current EV landscape. Following the review of the EV landscape, the focus transitions to the ontology languages and tools used to model all this information into an ontology. The Protege framework is employed for the development of the proposed ontology employing the OWL for standardized knowledge representation. The construction of the proposed EVO was based on similar ontologies emerged from the literature. Next, SPARQL queries were utilized to illustrate how the ontology can answer EV-related questions, assessing the competence of the created EV ontology. In order for the SPARQL queries to be more user friendly and useful, a menu interface Python script has been developed using the 'rdflib' library, which parses RDF files into a graph. Looking ahead, the objective is to enrich the EV ontology with more EVs to enhance its utility and comprehensiveness. However, the current lack of a standardized way to import individuals necessitates the development of new import methods. The use of a web scraper developed in this work is only made for a specific website that only stores information about BEVs. Although the web scraper in this work utilizes the state-of-the-art libraries for web scraping, more web scrapers, or a more general one, need to be developed for the rest of the taxonomies of EVs. It's worth noting that one of the primary motivations behind ontology research is the facilitation of reusability. Consequently, the proposed ontology can be reused and extended to cater to specific domains of interest.

References

1. Dimeth Nouicer, Ikbal Chammakhi Msadaa, K.G.: Ontology for the context of e-mobility: Charging station recommendation based on the ev trip (2022)
2. Gryparis, E., Papadopoulos, P., Leligou, H.C., Psomopoulos, C.S.: Electricity demand and carbon emission in power generation under high penetration of electric vehicles. a european union perspective. Energy Reports **6**, 475–486 (2020), technologies and Materials for Renewable Energy, Environment and Sustainability
3. Guarino, N., Oberle, D., Staab, S.: What Is an Ontology?, pp. 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-92673-3_0
4. IEA: Global ev outlook. Tech. rep. (2023)
5. Li, Z., Khajepour, A., Song, J.: A comprehensive review of the key technologies for pure electric vehicles. Energy **182**, 824–839 (2019)
6. Mitchell, R.: Web scraping with python. " O'Reilly Media, Inc." (2024)
7. Natalya F. Noy, D.L.M.: Ontology development 101: A guide to creating your first ontology (2001)
8. Novotny, A., Szeberin, I., Kovács, S., Máté, D.: National culture and the market development of battery electric vehicles in 21 countries. Energies **15**(4) (2022)
9. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of sparql. ACM Trans. Database Syst. **34**(3) (sep 2009)
10. Poullikkas, A.: Sustainable options for electric vehicle technologies. Renewable and Sustainable Energy Reviews **41**, 1277–1287 (2015)
11. Qi, Y., Mai, G., Zhu, R., Zhang, M.: Evkg: An interlinked and interoperable electric vehicle knowledge graph for smart transportation system. Transactions in GIS (2023)
12. Razmjoo, A., Ghazanfari, A., Jahangiri, M., Franklin, E., Denai, M., Marzband, M., Astiaso Garcia, D., Maheri, A.: A comprehensive study on the expansion of electric vehicles in europe. Applied Sciences **12**(22) (2022)
13. Santini, D.: Electric vehicle waves of history: lessons learned about market deployment of electric vehicles. IntechOpen (2011)
14. Sun, X., Li, Z., Wang, X., Li, C.: Technology development of electric vehicles: A review. Energies **13**(1) (2020)