Towards Enriching the Electric Vehicle Knowledge Graph by Linking it to DBpedia

Ioannis Kivrakidis
School of Informatics,
Aristotle University of Thessaloniki
Thessaloniki, Greece
ikivrakid@csd.auth.gr

Emmanouil S. Rigas
School of Medicine,
Aristotle University of Thessaloniki
Thessaloniki, Greece
erigas@auth.gr

Nick Bassiliades
School of Informatics,
Aristotle University of Thessaloniki
Thessaloniki, Greece
nbassili@csd.auth.gr

ABSTRACT

The automotive industry is focusing on Electric Vehicles (EVs) for their efficiency in reducing oil consumption and emissions. However, the EV market's diversity in battery capacities, classifications, and connector types creates a lack of standardization. Researchers are exploring advanced data and knowledge management methods, with Knowledge Graphs (KGs) emerging as a promising solution. KGs represent data in a way that reflects human understanding, promoting natural human-machine interactions and enhancing AI's insights. Their structure and constraints are defined by a vocabulary or ontology. This paper presents ongoing work towards enriching an existing Electric Vehicle Knowledge Graph (EVKG), which is specified by the Electric Vehicle Ontology (EVO). To achieve this, we utilized the Python programming language to create labels for each resource in the EVKG. These labels were then used to match and link these resources to their corresponding entries in DBpedia through the use of the owl:sameAs and rdfs:seeAlso properties. To ensure the matching was as accurate as possible, we developed two algorithms: one employing string matching and the other using word vectorization and distance techniques.

CCS CONCEPTS

• **Information systems** → *World Wide Web*; **Web Ontology Language (OWL)**; **Ontologies**.

KEYWORDS

Linked Data, Ontology, Knowledge Graph, EVO Ontology, Electric Vehicles

ACM Reference Format:
Ioannis Kivrakidis, Emmanouil S. Rigas, and Nick Bassiliades. 2024. Towards Enriching the Electric Vehicle Knowledge Graph by Linking it to DBpedia. In *13th Conference on Artificial Intelligence (SETN 2024), September 11–13, 2024, Piraeus, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3688671.3688734

1 INTRODUCTION

The growing number of internal combustion vehicles using non-renewable fuels has raised concerns about energy resources and the environment. As a result, many countries have introduced new energy vehicles (NEVs) to reduce oil dependency and mitigate air pollution. NEVs, which include EVs, hydrogen vehicles, natural gas vehicles, methanol vehicles, and ethanol vehicles, use alternative energy sources. Among these, EVs are considered the most effective for environmental and socioeconomic benefits [8]. As a technology that emerged after the industrial revolution [15], EVs have existed for over a century, with Thomas Parker inventing the first practical EV in 1884. They accounted for 28% of US vehicle production in the 1920s [17], but their adoption slowed due to high costs and 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 [11]. Indeed, the number of EVs in Europe [4] is expected to increase from about three-quarters of a million in 2019 to more than four million in 2025. Major automakers such as Volkswagen, Mercedes, and Ford are committed to promoting EVs, and new companies dedicated to EVs have emerged. EVs are primarily categorized into five types: battery electric vehicles (BEVs), hybrid electric vehicles (HEVs), plug-in hybrid electric vehicles (PHEVs), extended-range electric vehicles (ER-EVs), and fuel cell electric vehicles (FCEVs) [14].

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 definite. KGs stand out as the primary choice for organizing and representing this kind of knowledge because, as highlighted by [13], they can effectively represent complex information. This paper therefore aims to enrich such a KG to make it completer and more useful. The KG in question is defined by the EVO [9], a domain-specific ontology designed for the EV sector. EVO encompasses EV-related concepts, including EV taxonomies, connectors, charging infrastructure, and EV specifications. To this end, we developed two Python scripts that, among other functions, aim to map the entries of the EVKG to DBpedia's[1] by using the owl:sameAs and rdfs:seeAlso properties. Ontology mapping is used to align entities from different ontologies that have similar semantic properties, creating connections between these matched entities, with each connection typically classified as either equivalence or subsumption. The owl:sameAs built-in Web Ontology Language (OWL) property is frequently used to support Linked Data integration by declaratively interconnecting "equivalent" resources or individuals with the same identity across distributed datasets [6] [18]. It is observed

[1]https://www.dbpedia.org/

SETN 2024, September 11–13, 2024, Piraeus, Greece

ACM ISBN 979-8-4007-0982-1/24/09
https://doi.org/10.1145/3688671.3688734

that within most Linked Dataset categories, the owl:sameAs predicate is the most significant for linking, followed by rdfs:seeAlso [5]. The rdfs:seeAlso relates a resource to another resource "that might provide additional information about the subject resource."[19].

The DBpedia knowledge base is extensively used as a testing platform in the research community, with many applications, algorithms, and tools developed around, or applied to it. Available as Linked Data on the Web, DBpedia covers a wide range of topics and includes RDF links to various external data sources. Consequently, numerous Linked Data publishers have chosen to create RDF links pointing to DBpedia from their datasets[10]. Our approach ensures that the enriched EVKG not only benefits from the structured and comprehensive information within EVO but also integrates seamlessly with the broader web of data.

This paper is organized into six sections which are made to address the enrichment of the EVKG by linking it to DBpedia. Section 2 reviews existing literature on the use of ontology linking and the uses of owl:sameAS and rdfs:seeAlso properties. Section 3 details the EVO, outlining its structure and the various classes and properties it encompasses. Section 4 describes the Python scripts developed to enrich the EVKG, which includes pseudocode for better understanding. Section 5 presents the outcomes of the enrichment process, focusing on the success rate and accuracy of the links created. Finally, in Section 6, the main findings and conclusions were summarized and the implications for future research and applications were discussed.

2 RELATED WORK

Several key scientific papers have explored the application of the owl:sameAs and rdfs:seeAlso properties for linking ontologies to datasets on the web. The seminal article from Berners-Lee, Hendler and Lassila's on the Semantic Web [2] lays the foundational principles for understanding the importance of linking data. Moreover, the authors of [1] describe DBpedia as a pivotal project in the Linked Data community that serves as a central hub by utilizing the owl:sameAs property to interlink diverse datasets, thus enhancing data accessibility and interoperability. Moreover, Heath and Bizer [7] provide an in-depth examination of Linked Data principles and the practical use of the owl:sameAs property, emphasizing its role in evolving the web into a global data space. Additionally, Nazarian and Bassiliades [12] discuss the effort towards creating alternative links for DBpedia's bibliographic references using a combination of key-based and similarity-based approaches. Scharffe and Euzenat [16] explore the intersection of Linked Data and ontology matching, demonstrating how ontology alignments can enhance data linking by providing more accurate and meaningful connections between datasets. Ding et al. [3] provide an empirical study on the use of owl:sameAs within Linked Data, analyzing its implications and highlighting the importance of careful usage to avoid semantic ambiguities. These works collectively highlight the crucial role of of linking ontologies to web datasets, underscoring its significance in the development of the Semantic Web and Linked Data.

3 DESCRIPTION OF EVO

In this section, we describe the EVO.[2] As previously mentioned, the EVKG is defined by EVO. This ontology serves as a structured and systematic framework for organizing and categorizing various concepts, properties, and relationships related to EVs. It provides a foundation for understanding and classifying information concerning EV taxonomies, EV components, charging infrastructure, connector types, energy sources, regulatory and propulsion aspects, and environmental impacts.

Specifically, EVO [9] consists of 20 classes, 17 object properties, and 54 datatype properties. Each class is designed to capture a particular aspect of the EV landscape. For example, the EV class details aspects such as manufacturer, model type, battery specifications, model year, CO2 emissions, electric motor details, electric range, and performance specifications. The EV class has five subclasses: Battery Electric Vehicle (BEV), Hybrid Electric Vehicle (HEV), Fuel Cell Electric Vehicle (FCEV), Extended Range Electric Vehicle (EREV), and Plug-in Hybrid Electric Vehicle (PHEV), each with properties corresponding to their specific taxonomies. The entire class hierarchy of EVO is shown in Figure 1.

The EVO, combined with its instances, constitutes the EVKG. The purpose of this paper is to enrich the EVKG, expanding its scope and depth to provide a more comprehensive resource for the Electric Vehicles' ecosystem.

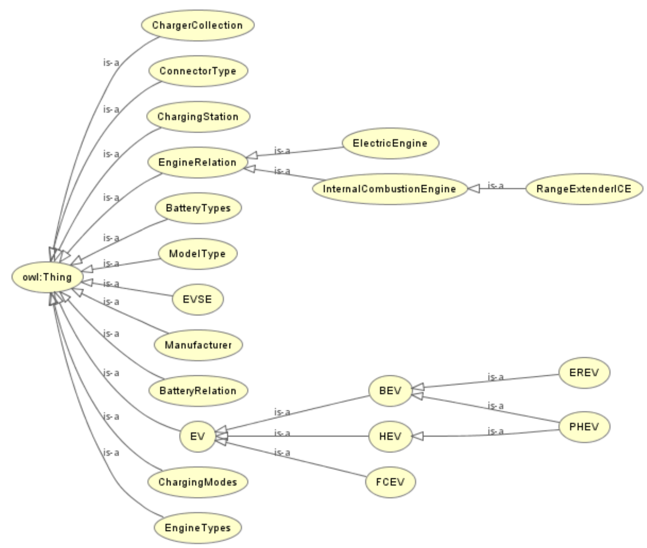

Figure 1: Class hierarchy of the EVO.

4 EVKG ENRICHMENT ALGORITHMS

4.1 Algorithm 1

In this section the first algorithm for enriching the EVKG by linking its resources to DBpedia is presented. This algorithm is implemented using a Python script that leverages libraries such as 'rdflib'[3] for RDF data manipulation and 'fuzzywuzzy'[4] for string

[2]https://w3id.org/evo
[3]https://rdflib.readthedocs.io/en/stable/
[4]https://pypi.org/project/fuzzywuzzy/

matching. Initially, the script loads the existing EVKG from an OWL file using rdflib. The data, defined by the EVO, is parsed into a graph object, and a namespace for the EVKG is established. Relevant classes within EVO, including various types of EVs (BEV, EREV, PHEV, FCEV, HEV) and related concepts like connector types, battery types, manufacturers, charging modes, and engine types, are then defined to facilitate iteration over each class's instances.

For each instance of the defined classes, the script generates an rdfs:label using the local name of the instance, modifying it for readability by replacing hyphens and underscores with spaces. These labels are added to the instances within the EVKG. To link these instances to DBpedia, the script sends search queries to the DBpedia Lookup API using the generated labels. The API responses provide potential matches from DBpedia, and the script retrieves the best match's URI. The fuzzywuzzy library calculates the similarity between the EVKG label and the DBpedia label to ensure sufficient match accuracy. If the similarity score exceeds a predefined threshold (0.6), an owl:sameAs link is added between the EVKG instance and the DBpedia URI, indicating equivalence and enriching the EVKG with data from DBpedia. If the similarity score falls within a lower range (between 0.5 and 0.6), an rdfs:seeAlso link is added instead. Finally, the enriched EVKG, now containing owl:sameAs and rdfs:seeAlso links to DBpedia, is serialized to a new OWL file.

4.2 Algorithm 2

In this section, we present the second algorithm for enriching the EVKG by linking its resources to DBpedia. This algorithm also utilizes a Python script that leverages libraries such as ‘rdflib‘ for RDF data manipulation and spaCy[5] for natural language proccesing (NLP). Up to the point of matching the generated labels to DBpedia, the algorithms are exactly the same.

In this algorithm, in order to determine the best match, the script uses the spaCy library to vectorize the EVKG labels and each candidate DBpedia label. It then calculates the cosine similarity between these vectors to ensure sufficient match accuracy. The script tracks the best match for each label, ensuring that only the closest match above a predefined similarity threshold (0.75) is considered. If such a match is found, an owl:sameAs link is added between the EVKG instance and the DBpedia URI. If the similarity score falls within a lower range (between 0.6 and 0.75), an rdfs:seeAlso link is added instead. Finally, the enriched EVKG, now containing owl:sameAs links to DBpedia, is similarly to the first algorithm serialized to a new Turtle file. Both methodologies enhance the EVKG by integrating external data from DBpedia, thereby increasing its completeness and utility.

The use of the owl:sameAs and rdfs:seeAlso properties enables integration into the broader Linked Data ecosystem. By linking the EVKG to DBpedia, the script capitalizes on DBpedia's extensive and structured information, resulting in a richer and more comprehensive Knowledge Bases (KBs) for EVs. An example instance written in Turtle after the enrichment process, along with the corresponding subgraph, can be seen in Figures 2 and 3, respectively. More specifically, Figure 3 presents a part of the EVKG, including two instances of EV and BEV types (Tesla Model Y, Tesla Model 3) linking to DBpedia. A pseudocode that summarizes how both algorithms work can be viewed at Algorithm 1.

Algorithm 1 Enhance EVKG with links to DBpedia

```
Require: RDF graph file, DBpedia lookup API URL, upper similarity threshold, lower similarity threshold
Ensure: Updated RDF graph with DBpedia links
Load RDF graph from file and bind namespace EX
Define class URIs and list of classes
for each class in classes do
    for each instance in class subjects do
        Add label to instance in graph after processing local name
    end for
end for
Set DBpedia lookup API URL and similarity thresholds
for each labeled instance in the graph do
    Create API request with label string and perform request
    Parse the response
    if response contains documents then
        Initialize variables for best similarity (sim) and DBpedia URI
        for each document in response do
            Extract DBpedia URI and label, calculate sim (string and semantic) between the instance and the DBpedia labels
            if sim is higher than current best sim then
                Update best sim and best DBpedia URI
            end if
        end for
        if best DBpedia URI is found then
            if best sim is above upper threshold then
                Link instance to DBpedia URI with owl:sameAs
            else if best sim is between lower and upper then
                Link instance to DBpedia URI with rdfs:seeAlso
            end if
            Document label details, DBpedia URI, and similarity
        else
            Document no suitable match found
        end if
    end if
end for
Serialize updated graph to output file
```

5 RESULTS

The enrichment process was evaluated based on the number of links established between EVKG and DBpedia entries. Considering the number of the resources in the EVKG, the 20% of them were linked through the owl:sameAs property (exact match), while the 5% of them were linked through the rdfs:seeAlso property (additional information). The remaining 75% were not linked to DBpedia, because they did not meet the lower similarity threshold. Moreover, a detailed investigation was made, in order to examine whether the links of the owl:sameAs property were correct or not. The examination showed that a high percentage of 90% of them were

[5]https://spacy.io/

correctly linked to the corresponding DBpedia entry. Although both algorithms performed well, the string-matching algorithm demonstrated slightly better average similarity results compared to the word vectorization algorithm (Table 1).

Table 1: Average Similarity Scores for Different Methods

Method	Average Similarity	Comments
String Matching	0.88	Algorithm 1
Word Vectorization	0.79	Algorithm 2

6 CONCLUSIONS

In this study, we successfully enriched an existing EVKG defined by the EVO by linking its resources to corresponding entries in DBpedia. We automated the creation of labels for each resource in the EVKG and employed string matching and vector similarity to identify and establish connections with DBpedia entries. This process leveraged the owl:sameAs property to integrate equivalent resources across the datasets and rdfs:seeAlso to provide additional information about the subject resource, thereby enhancing the EVKG with additional structured information. To determine which property to use, we set an upper and a lower bound: if the similarity of a given resource was above the upper bound, owl:sameAs was used, if the similarity fell between the upper and lower bounds, rdfs:seeAlso was used, and lastly if the similarity was below the lower bound, no link was created. The result of this process was an enriched EVKG which not only benefits from the comprehensive information contained within EVO, but also achieves greater interoperability with the broader web of data. This process demonstrates the potential of combining domain-specific ontologies with external Knowledge Bases to create more complete and useful KGs, ultimately contributing to better data and knowledge management in the field of EVs. Future work could explore further integration with other relevant datasets, such as Wikidata.

```
###  http://www.semanticweb.org/ev#Tesla-Model-Y
evo:Tesla-Model-Y rdf:type owl:NamedIndividual , evo:BEV ;
  owl:sameAs <http://dbpedia.org/resource/Tesla_Model_Y> ;
  evo:hasBattery evo:batteryRelation_12 ;
  evo:hasElectricRange "350 km" ;
  evo:hasMaxSpeed "217 km/h" ;
  evo:hasSystemHorsePower "220 kW (299 PS)" ;
  evo:hasSystemTorque "420 Nm" ;
  rdfs:label "Tesla Model Y"@en ;
  rdfs:seeAlso evo:Tesla .
```

Figure 2: Example Link to DBpedia

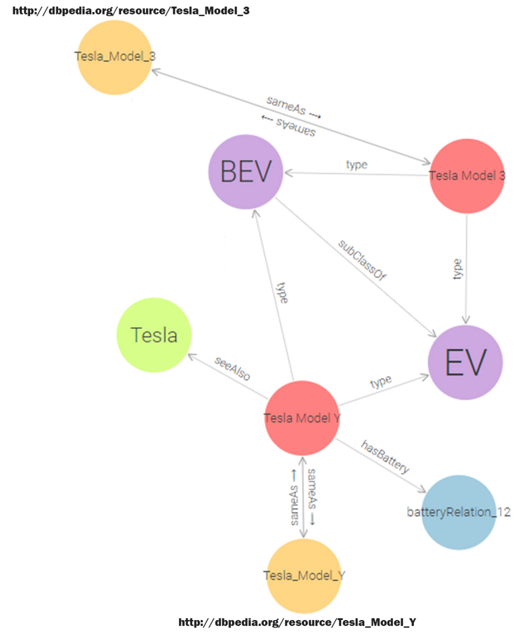

Figure 3: Example Subgraph

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*. Springer, 722–735.

[2] Tim Berners-Lee, James Hendler, and Olli Lassila. 2001. The Semantic Web" in Scientific American. *Scientific American Magazine* 284 (05 2001).

[3] Li Ding, Joshua Shinavier, Tim Finin, Deborah L McGuinness, et al. 2010. owl: sameAs and Linked Data: An empirical study. In *Proceedings of the Second Web Science Conference*.

[4] Emmanouil Gryparis, Perikles Papadopoulos, Hellen C Leligou, and Constantinos S Psomopoulos. 2020. Electricity demand and carbon emission in power generation under high penetration of electric vehicles. A European Union perspective. *Energy Reports* 6 (2020), 475–486.

[5] Armin Haller, Javier D Fernández, Maulik R Kamdar, and Axel Polleres. 2020. What are links in linked open data? A characterization and evaluation of links between knowledge graphs on the web. *Journal of Data and Information Quality (JDIQ)* 12, 2 (2020), 1–34.

[6] Harry Halpin, Patrick J Hayes, James P McCusker, Deborah L McGuinness, and Henry S Thompson. 2010. When owl: sameas isn't the same: An analysis of identity in linked data. In *The Semantic Web–ISWC 2010: 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I 9*. Springer, 305–320.

[7] Tom Heath and Christian Bizer. 2011. *Linked data: Evolving the web into a global data space*. Vol. 1. Morgan & Claypool Publishers.

[8] International Energy Agency. 2024. Global EV Outlook 2024. https://www.iea.org/reports/global-ev-outlook-2024 Licence: CC BY 4.0.

[9] Ioannis Kivrakidis, Emmanouil S. Rigas, and Nick Bassiliades. 2024. EVO: An Ontology for the Field of Electric Vehicles. In *Artificial Intelligence Applications and Innovations*. Springer Nature Switzerland, Cham, 335–348.

[10] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 6, 2 (2015), 167–195.

[11] Chengjiang Li, Michael Negnevitsky, Xiaolin Wang, Wen Long Yue, and Xin Zou. 2019. Multi-criteria analysis of policies for implementing clean energy vehicles in China. *Energy Policy* 129 (2019), 826–840.

[12] David Nazarian and Nick Bassiliades. 2017. Towards Linking DBpedia's Bibliographic References to Bibliographic Repositories. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 120–129.

[13] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review* 56, 11 (2023), 13071–13102.

[14] Andreas Poullikkas. 2015. Sustainable options for electric vehicle technologies. *Renewable and Sustainable Energy Reviews* 41 (2015), 1277–1287.

[15] DJ Santini. 2011. *Electric vehicle waves of history: lessons learned about market deployment of electric vehicles*. IntechOpen.

[16] François Scharffe and Jérôme Euzenat. 2011. Linked data meets ontology matching: enhancing data linking through ontology alignments. In *Proc. 3rd international conference on Knowledge engineering and ontology development (KEOD)*. No commercial editor., 279–284.

[17] Xiaoli Sun, Zhengguo Li, Xiaolin Wang, and Chengjiang Li. 2019. Technology development of electric vehicles: A review. *Energies* 13, 1 (2019), 90.

[18] W3C. 2004. OWL Web Ontology Language Reference. https://www.w3.org/TR/owl-ref/. Accessed: 2024-06-16.

[19] World Wide Web Consortium. 2023. *UsingSeeAlso*. https://www.w3.org/wiki/UsingSeeAlso Accessed: 2024-06-19.