Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.023063
Article

Tech Science Press

Ontology-Based Semantic Search Framework for Disparate Datasets

Paramjeet Kaur[1], Parma Nand[1], Salman Naseer[2], Akber Abid Gardezi[3], Fawaz Alassery[4], Habib Hamam[5], Omar Cheikhrouhou[6] and Muhammad Shafiq[7,*]

[1]Auckland University of Technology, Auckland, 1010, New Zealand
[2]Department of Information Technology, University of the Punjab Gujranwala Campus, Gujranwala, 52250, Pakistan
[3]Department of Computer Science, COMSATS University Islamabad, Islamabad, 45550, Pakistan
[4]Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia
[5]Faculty of Engineering, Moncton University, E1A3E9, Canada
[6]CES Laboratory, National School of Engineers of Sfax, University of Sfax, Sfax, 3038, Tunisia
[7]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, Korea
*Corresponding Author: Muhammad Shafiq. Email: shafiq@ynu.ac.kr

Received: 26 August 2021; Accepted: 09 October 2021

Abstract: The public sector provides open data to create new opportunities, stimulate innovation, and implement new solutions that benefit academia and society. However, open data is usually available in large quantities and often lacks quality, accuracy, and completeness. It may be difficult to find the right data to analyze a target. There are many rich open data repositories, but they are difficult to understand and use because these data can only be used with a complex set of keyword search options, and even then, irrelevant or insufficient data may eventually be retrieved. To alleviate this situation, ontology-based semantic search has been proven to be an effective way to improve the quality of related content queries in such repositories. In this paper, we propose a new method of semantic linking and storing open government datasets of New Zealand's agriculture, land and rainfall sectors based on the use of ontology. The generated ontology can construct integrated data, in which a unified query can be applied to extract richer and more useful information. To validate our model, we showed how to link ontology manually and automatically. Manual linking requires domain experts, and automatic linking reduces the overhead of relying on domain experts to manually link concepts. The results of this method are promising in terms of improving data quality and search efficiency. In future, the proposed model can be integrated with other domain ontologies.

Keywords: Open data; ontology; linked open data; semantic search

1 Introduction

The main goal of the World Wide Web (WWW) has always been to allow people to easily access information, regardless of whether machines also use the web network to transmit information. The Semantic Web is a web of data that uses Semantic Web standards for annotation [1], such as Resource Description Framework (RDF) [2] and Ontology Web Language (OWL) [3], which are usually related to each other based on context. We live in a world where data is ubiquitous and integral to our lives and is indispensable as members of organizations and communities [4]. The amount of data is growing at an unprecedented rate, and it is believed that the potential growth pattern will continue to rise [5].

Open data can be obtained through a variety of channels, such as open data repositories, portals, websites, and open-source tools. However, the open availability of data does not guarantee the integrity and consistency of the published data. The heterogeneous nature of data makes data extraction lengthy and time-consuming. Alternatively, portals and data on the website are only suitable for keyword-based searches, where the keywords entered by the user match the available data descriptions [6]. However, the problem is that users are not properly told what to search for and how they can modify keywords to get the best results. In addition, for a better search, the same alternative words may appear, but the user can't know these terms because the user may not be familiar with the structure used by the data publisher to describe it. There may be synonyms that match the user's intent, but the user does not know the actual terminology the publisher uses to refer to their repository. Semantic search solves this problem and aims to improve the accuracy of the search by considering the searcher's intention and the contextual importance of the search term [7]. The motivation of our research is to explore whether semantic technology can improve the usability and efficiency of search in more and more open data.

An effective semantic search engine attempts to analyze the user's intention to search for content and the expected meaning of particular search content. If we link data by analogy, it will help citizens find and use information more easily. The openness of data links allows developers to build useful applications and helps contribute to the development of the country. Entrepreneurs can use connected open data to build innovative business ideas and products that help and stimulate the country's economic growth. Linked open data enhances public participation and indirectly helps governments improve efficiency and decision-making. The government will inform citizens of their behaviour. This will help build trust between the government and citizens. It will also improve government processes and services. However, making OGD [8] useful is challenging. Although the number of attempts to disclose OGD is rapidly increasing, it is still a huge challenge to reach the maximum capacity of OGD sources and support the consumption and release of these data by all partners. One of the main challenges in solving this problem is the heterogeneity of the data format and structure used by government agencies. Due to this heterogeneity, both data providers and data users face technical challenges.

In this article, we introduced a new method for real-world analysis of open data and demonstrated an example of using ontology. Ontologies are semantically related, so SPARQL queries are imposed to extract useful information [9]. A prototype has been implemented that supports the semantic linking of concepts related to the agricultural, land and rainfall sector datasets published by various departments of the New Zealand government. Preliminary findings indicate that ontology semantic search can be applied to open data to improve the quality and effectiveness of the search. The SPARQL query applied to the generated ontology will bring information according to the user's request. This means that users no longer need to search for different data sources because all the information is concentrated in one place.

We summarize the main contributions of this paper as follows.

- We conducted a study to analyze New Zealand's open data set and convert it into linked open data to extract useful information for novice users and society.

- We developed a prototype to generate the semantical link ontology for the open government datasets automatically. We present a case study-based evaluation where open datasets of Agriculture, land and Rainfall sectors are used to generate semantically linked ontology.

- We designed a SPARQL interface to impose queries on the generated ontologies so that knowledgeable data can be extracted more easily.

The structure of the paper is as follows. Section 2 summarizes the related work. Section 3 provides detailed information on the proposed method of automatically creating and linking semantic ontology. Section 4 describes the results and discussion. In the last section, we concluded.

2 Related Work

This section introduces the background of work carried out in the field of open government data implementing ontology frameworks, models, or prototypes. This review aims to compare the proposed solutions and technologies. Kalampokis et al. [10] proposed architecture for determining indirect links between entities to create linked data. The main goal is to create owl: SameAs links between Uniform Resource Identifiers (URIs) to connect different Open Government Data (OGD). In addition, a prototype scene was developed that involved three participants: the school, the Athens Regional Secondary Education Bureau, and the Ministry of Education. To access data consumption related to a particular school, 5 execution steps were performed. The SPARQL Protocol and RDF Query Language (SPARQL) endpoint are used to access the specified data set. However, the whole process is slow and tedious, requiring users to put in a lot of effort. An open-source framework called Silk was discussed to merge different data sources, but there were no implementation details on how it was implemented. More research is needed to understand the relationship between political priorities and data models. In addition, it is necessary to describe a method or approach to semantically encoding different data sources. The result of the SPARQL query is knowledge data (or linked data), which can provide multiple benefits such as transparency, reusability, economic growth, and public participation [11].

In [12] and [13], the author connects the open data project by using an interface based on the E-GIF ontology and the Jena framework. Jena is an open-source semantic web framework. It provides an application programming interface (API) for extracting data from RDF. The SPARQL endpoint is also used to query the data set. However, the focus is on the operation and features of the Jena framework. The proposed method seems to be based on the effectiveness of the tool. In addition, the RDF model is created using a pre-designed E-GIF ontology. There is no clear evidence of the status of the implementation details of the E-GIF ontology. Jiang et al. [14] proposed a search engine prototype, which is used to link the concept of transportation domain ontology manually and automatically.

The results proved a higher quality and more effective search for open data. Nevertheless, the proposed system can be enhanced by using additional data sets and provide more detailed metadata descriptions, which helps to generate semantic link data. At this stage, only one domain is considered, and there is no evidence on how to incorporate other domains into the prototype. In [15], the author proposed a state-of-the-art theory that provides a new method for creating and publishing ontology-based systems by using linked open data from Valencia's water resources management. The proposed ontology can be used to identify the correlation between water sources, leaks, and population. Water resource management decision-making needs to integrate multiple heterogeneous data sources and various data domains. The main goal is to help decision-makers achieve better results by using rich and comprehensive information. However, the interconnection can be enhanced by merging additional data sets from different fields.

With the increase in knowledge representation, deep learning, Natural Language Processing (NLP), machine learning, and daily data volume, ontologies have become more and more important. Ontology engineering is the method and process of research and development of ontology, including the representation, formal naming and description of categories, properties and relationships between concepts, data and entities [16]. In addition, the implementation of e-government by semantic network technology has brought various types of challenges, including economic, cultural, human, technical, social, data quality, and legislation. By strengthening knowledge sharing, citizens can gain greater advantages from using semantic web applications in e-government [17].

The above-mentioned studies only used open government data from different countries and departments to investigate prototypes, E-GIF ontologies, search engines, and ontology-based frameworks. Although extensive research has been conducted, there is still more room to implement mechanisms and technologies to take advantage of open data procedures to extract valuable information for the benefit of the public. There is an urgent need for a new method, especially to convert different open data sources into a common form, so that a huge knowledge base can be created, in which multiple data sources can be used to generate semantically rich data. The purpose of this research is to use the open government data set of the New Zealand government to develop a simple and accurate method to generate semantically rich automatic ontology.

3 Methodology

This section describes how to convert comma-separated value (CSV) data to Web Ontology Language (OWL) [18]. Since it is fully automated and therefore less time-consuming, the proposed method is different from traditional data conversion. It converts the CSV data set to OWL format and allows the generation of semantic links between different ontologies to develop an automated knowledge base that can use SPARQL to query and extract data.

The basic concepts of this method are as follows:

• The syntax and semantics of CSV follow the constraints and definitions specified in the RFC4180 document describing the dialect. RFC4180 is used as the dialect description because it can automatically recognize the CSV file format. The default method of the CSV format library is used to parse CSV files.

• The CSV data is annotated using Dublin Core Metadata [19] to achieve interoperability with the OWL metadata vocabulary. It allows accurate and consistent organization and enrichment of data across multiple modes.

• The union of two ontology files and the HyperSQL(HSQL) [20] database is used to semantically link one or more ontology. The union here consists of all ontology classes, individuals, and data properties of the two ontology files. The main purpose of creating a joint file is to save all the information in the ontology file without losing information.

• HSQL database helps to convert ontology classes into tabular form. This conversion makes the semantic link search operation simple because the tabular form is easy to traverse when performing linear search operations.

• The cosine similarity measure is used to identify the similarity between the literal values of the ontology data properties. It provides a function for comparing two strings and returns the similarity score used to identify the most suitable match for an individual.

• For the ontology merging process, we select common properties from the generated ontology, and generate semantic links based on these properties. This helps to align two or more ontologies into a single modular ontology.

• For ontology visualization, protection tools are used. In addition, SPARQL [21] query is imposed on the ontology, and Apache Jena is used to converting OWL triples into RDF/Turtle form so that SPARQL queries can be executed to obtain the desired results.

3.1 Architecture and Process Flow of the Application

A prototype has been developed using the proposed method. We have implemented the prototype of the CSV to OWL conversion and the related conversion mechanism. The conversion method accepts a CSV file and Dublin core meta-words as input and outputs the converted OWL. The created OWL file is located in the local memory of the system. In addition, two or more generated ontologies are semantically linked through the use of union, HSQL, and sequential search operations. To obtain accurate results, the cosine similarity measure is applied to the semantically generated ontology to eliminate duplication. Then convert the generated OWL file to Turtle format to execute SPARQL queries. In our proposed model, the entire ontology generation process is divided into three stages: 1) use the Protege tool to convert and visualize CSV to OWL; 2) generate semantic links between two or more ontologies; and 3) query the SPARQL of the created ontology Interface. In Fig. 1, we have shown the overall design and process, as well as the key elements of the architecture.

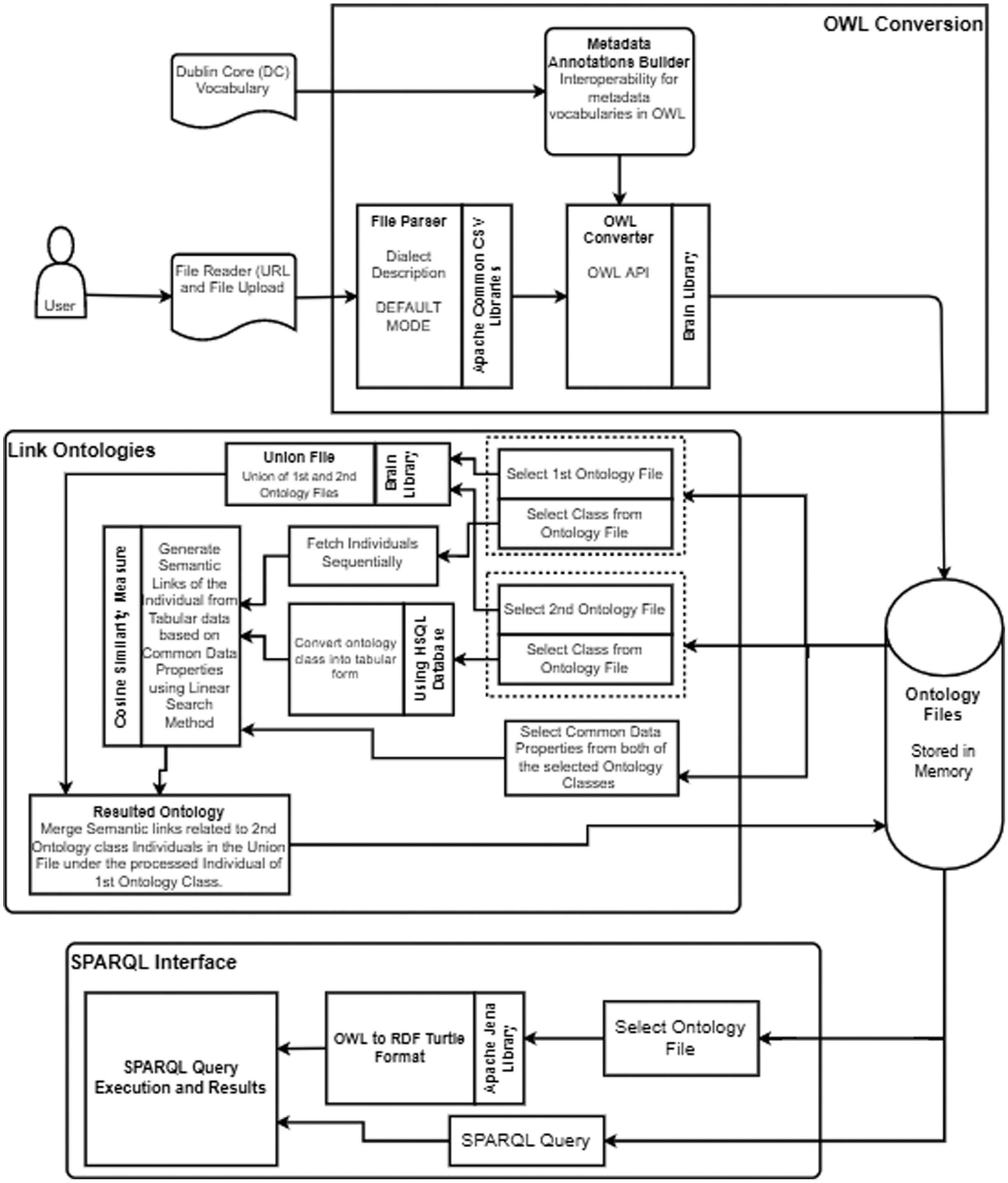

OWL Conversion
Dublin Core (DC) Vocabulary
Metadata Annotations Builder
Interoperability for metadata vocabularies in OWL
User
File Reader (URL and File Upload
File Parser
Dialect Description
DEFAULT MODE
Apache Common CSV Libraries
OWL Converter
OWL API
Brain Library
Link Ontologies
Union File
Union of 1st and 2nd Ontology Files
Brain Library
Select 1st Ontology File
Select Class from Ontology File
Fetch Individuals Sequentially
Cosine Similarity Measure
Generate Semantic Links of the Individual from Tabular data based on Common Data Properties using Linear Search Method
Convert ontology class into tabular form
Using HSQL Database
Select 2nd Ontology File
Select Class from Ontology File
Select Common Data Properties from both of the selected Ontology Classes
Resulted Ontology
Merge Semantic links related to 2nd Ontology class Individuals in the Union File under the processed Individual of 1st Ontology Class.
Ontology Files
Stored in Memory
SPARQL Interface
SPARQL Query Execution and Results
OWL to RDF Turtle Format
Apache Jena Library
Select Ontology File
SPARQL Query

Figure 1: Architecture and process of the proposed model

3.2 CSV to OWL Conversion and Visualization Using Protege Tool

The first step involves automatically creating an ontology from a CSV data file. Users can choose to upload the CSV file directly or enter the CSV file's accessible Uniform Resource Locator (URL). The current process can only accept CSV data streams as input. However, other modes can be easily added, including portable document format (PDF), hypertext markup language (HTML), keyhole markup language (KML), and JavaScript object notation (JSON).

The Apache Commons CSV library is used to parse CSV files, which follow the constraints and definitions described in the dialect of the RFC4180 document. This specifies the format of the CSV file, such as headers, line endings, and escape characters, and helps with the processing of CSV's text-based fields. The default mode specifies that non-Unicode characters in the CSV file are replaced with Unicode based on the dialect definition. The CSV parser uses various functions to read and parse the rows, cell values, and reference values of CSV data.

The conversion of CSV data to OWL mainly requires the addition of metadata notes describing the data interpretation method. Because it has been recognized as a tool, non-experts can use it to quickly generate transparent and informative records of information resources, while also providing effective search for resources in an integrated world. The Dublin Core Resource Description Framework Architecture (RDFS) vocabulary is used to define common metadata. The reason for using the RDFS vocabulary is that it has been widely recognized as a mechanism by which non-experts can easily build accurate and comprehensive records for data sources, and at the same time, they can perform search on such resources well in an interconnected environment. The name of the CSV file entered by the user is designated as the ontology name, the column header of the CSV file is designated as the data property, and the value stored in the data property is regarded as the entity of the ontology. Each row under the column heading is treated as an individual, and each record is assigned a unique identifier name. While performing all the transformations, this process will also help add corresponding axioms between data properties and individuals to explain the meaning of classes and their relationships. After adding all the axioms, the ontology is stored in local memory, and the generated ontology is visualized using the protege 5.5 tools.

3.3 Generating Semantic Links Between Two or More Ontologies

Based on the first step, all ontology files are created, and these files are stored in the system's local storage. To semantically link the ontology, the user must select two or more ontology files. Since the ontology file can have multiple ontology classes, it is necessary to select a specific class from the selected ontology file. The user also needs to select common data properties from the two selected ontology classes to perform semantic link operations. Once the user has generated all the ontologies mentioned in Section 3.2, the user can select two different ontologies to create a semantic link between them. After selecting the two ontology files, the next step is to merge these files to form a union. The union consists of all ontology classes, individuals, and data properties. The main purpose of creating a joint file is to preserve all the information of the ontology file, so as not to lose any information.

As part of this process, we use the HSQL database to convert the classes of the second ontology file into tabular form. The reason behind this conversion is to make the semantic link search operation simple because the table form is easy to traverse when performing a linear search operation. The next step is to get the individuals one by one sequentially from the first ontology class. Further, the data properties of the first ontology type individual are compared with the tabular data, and a linear search is performed to find the semantically related second ontology type individual. It is challenging to semantically link these data sets based on common properties because individuals may have duplicate values. Therefore, the ideal solution for linking these ontology classes is based on standards that determine the similarity between the literal values of the data properties of the ontology classes. To overcome this challenge, we used the cosine similarity measure, which provides a function for comparing two strings and returning the similarity score.

Since we have repetitions in the area names, cosine similarity is useful for situations where repetition is important. The linear search method will locate individuals in tabular data to identify semantic links on selected public properties. It will continue to browse through each individual of the data in order until it finds a match or searches the entire table. The generated semantic link is further merged into the second ontology individual in the union file. It will lead to a semantically linked ontology of two different data departments or domains. The overall architecture flow of this process is shown in Fig. 1. In addition, the linking process mentioned here is a way, which means that we can link ontology class 1 to ontology class 2, and vice versa. The two-way link process will be considered in future enhancements.

3.4 SPARQL Interface to Query the Generated Ontology

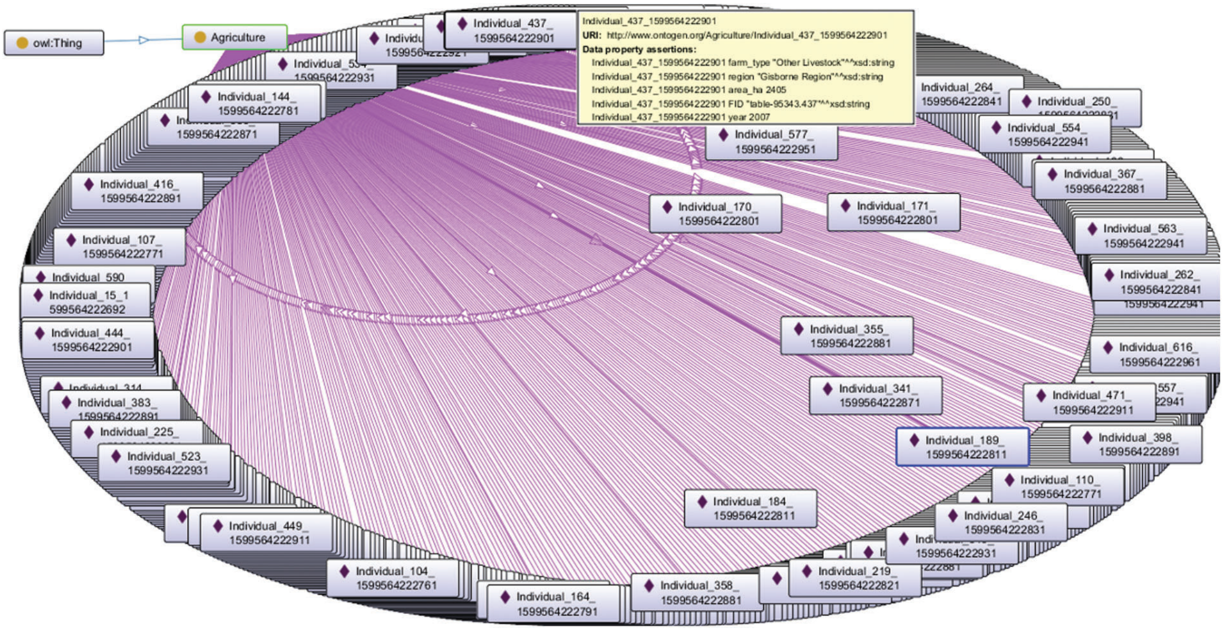
Once we have obtained our semantic link ontology. The third and final stage is to query the generated ontology with the help of the SPARQL interface. The design of the SPARQL interface allows users to select the required ontology file for the query. To query in SPARQL, we need to convert the generated OWL ontology into an RDF format, such as Turtle. Writing SPARQL queries involving complex OWL expressions ranges from challenging to unpleasant because SPARQL query syntax is based on Turtle [22], and this does not apply to OWL. SPARQL queries for OWL data must encode the RDF serialization of OWL expressions: these queries are often lengthy, difficult to write and understand. For the conversion process, we used Apache Jena, which is a Java library that can be used to convert OWL files to RDF Turtle format and provides APIs to query SPARQL from Java applications. Fig. 1 highlights the process of the SPARQL interface.

4 Results and Discussion

The above methods have been evaluated using the open datasets of agriculture, land and rainfall on the New Zealand government website. First, we create the ontology of agriculture, land, and rainfall datasets. To do this, we enter the URL of the dataset (https://data.mfe.govt.nz) into the system. The entered URL is parsed using a file parser, and the CSV data set is converted into an OWL file using an OWL converter. These OWL files are stored in the system's local memory. The schematic diagram of the agricultural ontology is shown in Fig. 2. Agriculture is a subcategory of owl: it has 5 data properties (i.e., area hectares, farm type, FID, region, year) and 620 individuals (from individual 1_159956422651 to individual 602_159956422961). When the mouse pointer hovers over an individual, it will highlight the data property assertion for that particular individual. Similarly, we create and store ontologies for land and rainfall datasets. In addition, these created ontologies are semantically linked so that SPARQL queries can be applied to extract knowledgeable data. To do this, we select second ontology files from the local drive of the system and manually identify their common data properties.

In the second ontology, year and region are common data properties. Therefore, these data properties are selected for semantic link generation. We create a joint file by combining two ontology files to avoid any loss of information. The joint file has been put aside and merged with the resulting ontology later. For semantic link generation, agriculture and land ontology are regarded as the first and second ontology files, respectively. If the ontology has multiple classes, we need to ensure that the appropriate class is selected from the selected ontology file. From the agriculture OWL file and the land OWL file, we have selected the agriculture and land categories respectively. The agricultural individuals are obtained one by one, and the land is transformed into a table form, which makes the operation of semantic link search simple, and the table form is easy to traverse when performing linear search operations.

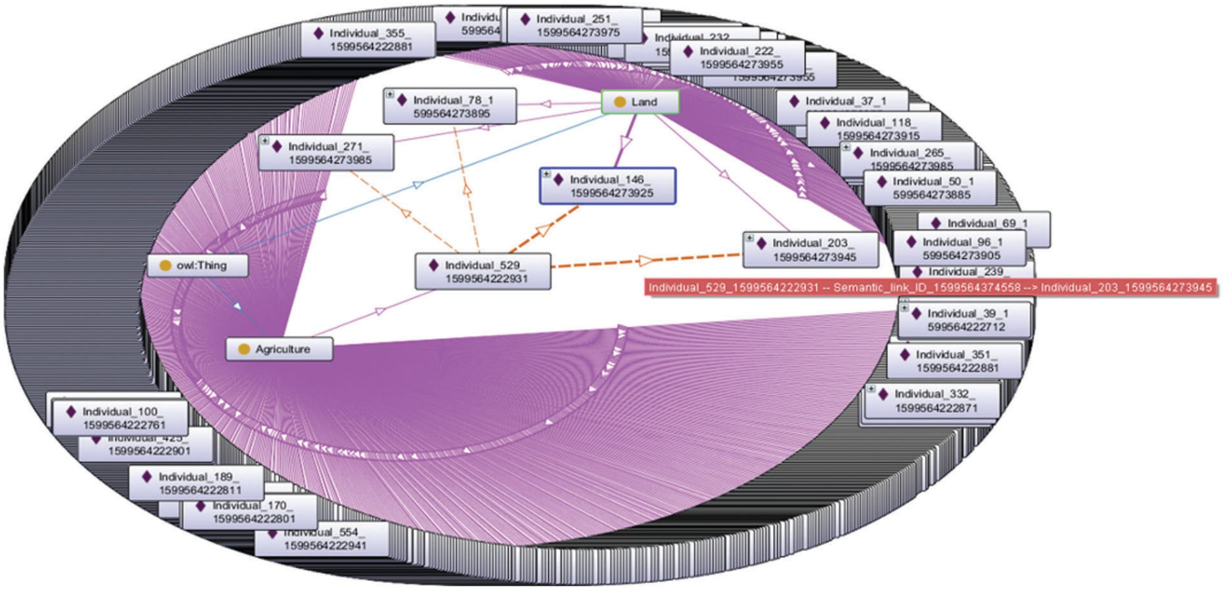
Finally, a linear search of agricultural individual and land table data is used to generate semantic links. Here, cosine similarity is used to avoid repetition. Before getting the final linked ontology, we combine the previously created joint file with the generated ontology file, so that we can put semantic and non-semantic data together. Fig. 3 shows the semantically related ontology of the agricultural and land datasets. The red dotted arrow indicates the semantic connection between land and agricultural individuals. The total number of individuals is 892, and the total number of semantic links is 168. Due to screen size limitations, we cannot display all semantically linked individuals. In order to create the ontology, the existing ontology is not used. All ontologies are newly generated using open datasets from the three departments of the New Zealand government's agriculture, land, and rainfall. The agricultural profile and all generated ontology are OWL-RL, because it is mainly for applications that require scalable reasoning without giving up too much representational potential. In addition, it also provides features, specifications, axioms, reasoning and expressions for the design of RDF triples [23].



Individual_437_1599564222901
URI: http://www.ontogen.org/Agriculture/Individual_437_1599564222901
Data property assertions:
Individual_437_1599564222901 farm_type "Other Livestock"^^xsd:string
Individual_437_1599564222901 region "Gisborne Region"^^xsd:string
Individual_437_1599564222901 area_ha 2405
Individual_437_1599564222901 FID "table-95343.437"^^xsd:string
Individual_437_1599564222901 year 2007

owl:Thing
Agriculture
Individual_437_ 1599564222901
Individual_144_ 1599564222781
1599564222871
Individual_416_ 1599564222891
Individual_107_ 1599564222771
Individual_590
Individual_15_1 599564222692
Individual_444_ 1599564222901
Individual_383_ 1599564222891
Individual_225_
Individual_523_ 1599564222931
Individual_449_ 1599564222911
Individual_104_ 1599564222761
Individual_164_ 1599564222791
Individual_358_ 1599564222881
Individual_219_ 1599564222821
Individual_184_ 1599564222811
Individual_246_ 1599564222831
Individual_110_ 1599564222771
Individual_189_ 1599564222811
Individual_398_ 1599564222891
Individual_471_ 1599564222911
Individual_341_ 1599564222871
Individual_355_ 1599564222881
Individual_616_ 1599564222961
Individual_262_ 1599564222841
Individual_563_ 1599564222941
Individual_367_ 1599564222881
Individual_554_ 1599564222941
Individual_250_
Individual_264_ 1599564222841
Individual_577_ 1599564222951
Individual_170_ 1599564222801
Individual_171_ 1599564222801

Figure 2: Agriculture ontology with all individuals



Individual_529_1599564222931 -- Semantic_link_ID_1599564374558 --> Individual_203_1599564273945

Individual_355_ 1599564222881
Individual_251_ 1599564273975
Individual_222_ 1599564273955
Individual_78_1 599564273895
Land
Individual_37_1
Individual_118_ 1599564273915
Individual_265_ 1599564273985
Individual_50_1 599564273885
Individual_271_ 1599564273985
Individual_146_ 1599564273925
Individual_203_ 1599564273945
Individual_69_1
Individual_96_1 599564273905
Individual_239_
owl:Thing
Individual_529_ 1599564222931
Agriculture
Individual_39_1 599564222712
Individual_351_ 1599564222881
Individual_332_ 1599564222871
Individual_100_ 1599564222761
1599564222901
Individual_189_ 1599564222811
Individual_170_ 1599564222801
Individual_554_ 1599564222941

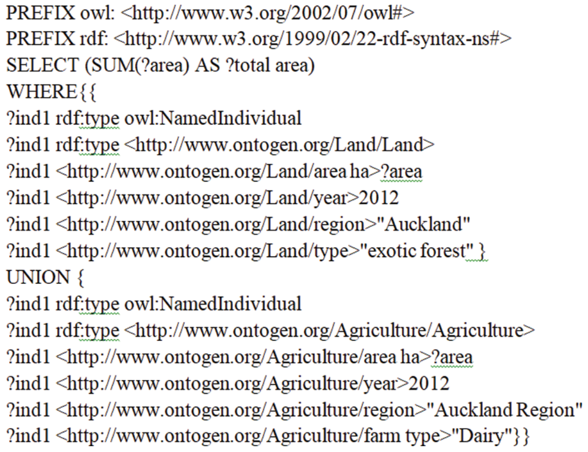
Figure 3: Semantically linked ontology of agriculture and land datasets

It is important to analyze the validity of the generated ontology to evaluate our method. To achieve this, we provide some sample SPARQL queries to the system. We record the responses to these queries and manually evaluate their consistency. SPARQL queries are applied to data sets to retrieve valuable information. Traditional information extraction techniques are very time-consuming because the data sets must be thoroughly read, but the RDF query language ultimately makes the task simple because all data is stored as triples. By using SPARQL queries, you can easily find knowledge-rich data. For testing purposes, we implement the following SPARQL queries on the generated ontology:

- Query 1: Find the area hector used by dairy and exotic forest in the year 2012 in the Auckland region.

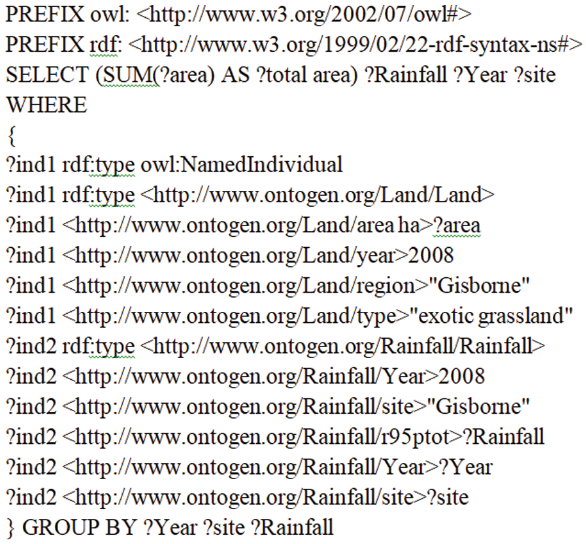
- Query 2: Find the area hector used by exotic grasslands in Gisborne in the year 2008 and what is the rainfall rate for that year and region.

We show the syntax of SPARQL queries 1 and 2 in Figs. 4 and 5, respectively.



```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT (SUM(?area) AS ?total area)
WHERE{{
?ind1 rdf:type owl:NamedIndividual
?ind1 rdf:type <http://www.ontogen.org/Land/Land>
?ind1 <http://www.ontogen.org/Land/area ha>?area
?ind1 <http://www.ontogen.org/Land/year>2012
?ind1 <http://www.ontogen.org/Land/region>"Auckland"
?ind1 <http://www.ontogen.org/Land/type>"exotic forest" }
UNION {
?ind1 rdf:type owl:NamedIndividual
?ind1 rdf:type <http://www.ontogen.org/Agriculture/Agriculture>
?ind1 <http://www.ontogen.org/Agriculture/area ha>?area
?ind1 <http://www.ontogen.org/Agriculture/year>2012
?ind1 <http://www.ontogen.org/Agriculture/region>"Auckland Region"
?ind1 <http://www.ontogen.org/Agriculture/farm type>"Dairy"}}
```

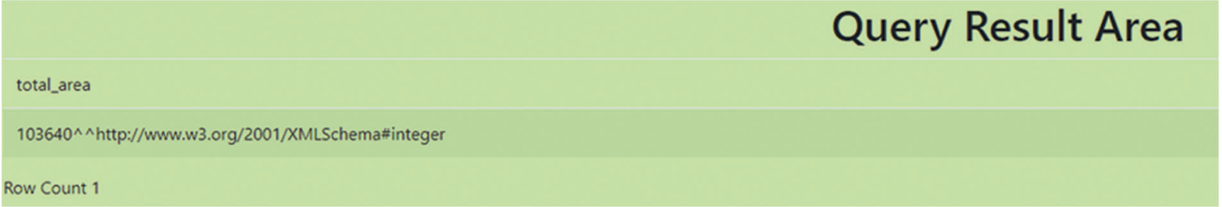
Figure 4: Structure of SPARQL test query 1



```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT (SUM(?area) AS ?total area) ?Rainfall ?Year ?site
WHERE
{
?ind1 rdf:type owl:NamedIndividual
?ind1 rdf:type <http://www.ontogen.org/Land/Land>
?ind1 <http://www.ontogen.org/Land/area ha>?area
?ind1 <http://www.ontogen.org/Land/year>2008
?ind1 <http://www.ontogen.org/Land/region>"Gisborne"
?ind1 <http://www.ontogen.org/Land/type>"exotic grassland"
?ind2 rdf:type <http://www.ontogen.org/Rainfall/Rainfall>
?ind2 <http://www.ontogen.org/Rainfall/Year>2008
?ind2 <http://www.ontogen.org/Rainfall/site>"Gisborne"
?ind2 <http://www.ontogen.org/Rainfall/r95ptot>?Rainfall
?ind2 <http://www.ontogen.org/Rainfall/Year>?Year
?ind2 <http://www.ontogen.org/Rainfall/site>?site
} GROUP BY ?Year ?site ?Rainfall
```

Figure 5: Structure of SPARQL test query 2

Compared with the traditional relational database query, the SPARQL query is simple but efficient as shown in Figs. 4 and 5. SPARQL can be used to query any database, and any middleware can be used to interpret the results as RDF. On the other hand, relational database requests are limited to specific databases. SPARQL is a Hypertext Transfer Protocol (HTTP) protocol that allows connection to any SPARQL endpoint through the structured transport layer. We process the test queries (1 and 2) and capture their results, as shown in Figs. 6 and 7, respectively. For consistency and accuracy, we conduct manual analysis. Therein, it is found that the result captured by the system is accurate.

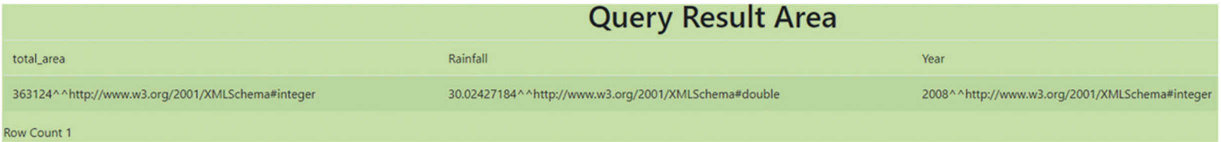


Query Result Area

total_area
103640^^http://www.w3.org/2001/XMLSchema#integer

Row Count 1

Figure 6: Result of the test query 1 for agriculture and land dataset ontology



Query Result Area

total_area	Rainfall	Year
363124^^http://www.w3.org/2001/XMLSchema#integer	30.02427184^^http://www.w3.org/2001/XMLSchema#double	2008^^http://www.w3.org/2001/XMLSchema#integer

Row Count 1

Figure 7: Result of the test query 2 for land and rainfall dataset ontology

At present, the New Zealand open data set does not have a semantically related ontology framework. It is necessary to access information through an open portal. Due to the heterogeneity of the open data set, it is difficult for end users to find useful information without traversing multiple pages or links. Even in most cases, they ultimately have no information. The proposed system helps to improve data quality and search efficiency in less time, because users can easily find linked information in one place, and by applying a query, the required data can be extracted.

It is expected that the introduction of automatic ontology generation prototypes will improve the consumption and access of open data sources by the New Zealand government. This method currently only considers the CSV data set and needs to be further improved, such as trying to use other formats and using two-way semantic encoding. The results are positive, and it is worthwhile to further study the use of this method to connect to other areas of government. This method has practical uses in applications such as generating ontology for any open data set of the New Zealand government. However, the ontology generation of semantic links requires some manual analysis, and users need to identify common properties between different data sets so that semantic links can be generated.

In addition, the protege tool is used to generate the visualization of the ontology. It has plug-ins that can accommodate the visualization of ontology's classes, instances, and data properties. Users can visualize part of the entire ontology. However, when the scale of data grows, the protege cannot handle it, and it becomes challenging to visualize the generated ontology. The large size of the ontology also affects the overall performance and loading time of the knowledge base. However, visualization is not the focus of research, but in the future, visualization can be improved by using third-party plug-ins or creating an additional tool that can improve the overall visualization of the ontology. The implemented SPARQL interface is processing the imposed queries, which are limited to users with technical knowledge. At the current stage of research, natural language queries that can help non-technical and non-SPARQL users are not considered. These functions can be implemented in future enhancements of the proposed system. In addition, the current model is validated using SPARQL endpoints, where example queries are used to generate knowledge-rich data. No other reasoners or editors are used at this stage.

A key advantage of this research is that it will automate the ontology generation process, and semi-automatic methods will help generate semantic links. These findings indicate that automatic ontology generation and semantic encoding of different data sets may be useful tools for leveraging valuable knowledge. Despite the success, an important limitation is that semantic links can be generated in one way. The links work from left to right. For example, if we want to semantically link agricultural and land datasets, we need to keep the agricultural ontology on the left. In this way, we will get the ontology of agriculture->land semantic association. If we want to go the other way, we need to keep the land body on the left. In the future, the limitations of this research must be considered to generate two-way semantic links in one go. In addition, the semantic link generation method can be fully automated in the future. To this end, a process can be implemented that can parse the data set and record public entries, and then the table can be used to generate semantic links.

5 Conclusion

In this paper, we propose a new method of using ontology semantics to link and store open government datasets of New Zealand's agriculture, land, and rainfall sectors. Our comprehensive approach includes the following: 1) The process of reading and parsing the CSV dataset; 2) The conversion process of converting the CSV dataset in OWL using Dublin Core metadata annotations; 3) the realization of the process of generating semantic links between two or more generated ontology; 4) SPARQL interface development is used to query and extract useful information from the generated semantically rich ontology. To validate the proposed solution, the OWL conversion process is used to convert various existing CSV files collected from the New Zealand govt.nz data portal. Our solution can generate an automatic ontology of any data set of the New Zealand government. However, the semantic link generation process requires some manual work in identifying the common properties between ontology files. In the future, we plan to consider the conversion of other formats and implement an algorithm that can simplify the process of semantic link generation. In addition, our case study implementation proves the effectiveness and feasibility of our proposed method. Automatic ontology generation and semantic coding of different data sets can be useful tools for utilizing valuable knowledge. Future work includes improving the visualization and processing time of the generated ontology. In addition, the proposed framework can be used to test open data sets from other countries and fields. In order to allow non-technical users to access the system, natural language query capabilities can be added to future enhancements to the system.

Acknowledgement: We deeply acknowledge Taif University for supporting this study through Taif University Researchers Supporting Project Number (TURSP-2020/150), Taif University, Taif, Saudi Arabia.

Funding Statement: This work was supported by Taif University (in Taif, Saudi Arabia) through the Researchers Supporting Project Number (TURSP-2020/150).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. Berners-Lee, J. Hendler and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.

[2] Z. Ma, M. A. Capretz and L. Yan, "Storing massive resource description framework (RDF) data: A survey," *The Knowledge Engineering Review*, vol. 31, no. 4, pp. 391–413, 2016.

[3] P. Hitzler, “A review of the semantic web field,” *Communications of the ACM*, vol. 64, no. 2, pp. 76–83, 2021.

[4] T. H. Davenport, “Competing on analytics,” *Harvard Business Review*, vol. 84, no. 1, pp. 1–9, 2006.

[5] H. Chen, R. H. Chiang and V. C. Storey, “Business intelligence and analytics: From big data to big impact,” *MIS Quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012.

[6] R. Fleiner, “Linking of open government data,” in *Proc. Int. Symp. on Applied Computational Intelligence and Informatics*, Timisoara, Romania, pp. 1–5, 2018.

[7] N. Shadbolt, T. Berners-Lee and W. Hall, “The semantic web revisited,” *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 96–101, 2006.

[8] E. Ruijer and A. Meijer, “Open government data as an innovation process: Lessons from a living lab experiment,” *Public Performance & Management Review*, vol. 43, no. 3, pp. 613–35, 2020.

[9] X. Yin, D. Gromann and S. Rudolph, “Neural machine translating from natural language to SPARQL,” *Future Generation Computer Systems*, vol. 117, pp. 510–519, 2021.

[10] E. Kalampokis, E. Tambouris and K. Tarabanis, “A classification scheme for open government data: Towards linking decentralised data,” *International Journal of Web Engineering and Technology*, vol. 6, no. 3, pp. 266–285, 2011.

[11] J. Attard, F. Orlandi, S. Scerri and S. Auer, “A systematic review of open government data initiatives,” *Government Information Quarterly*, vol. 32, no. 4, pp. 399–418, 2015.

[12] P. Fragkou, N. Kritikos and E. Galiotou, “Querying Greek governmental site using SPARQL,” in *Proc. Pan-Hellenic Conf. on Informatics*, Patras, Greece, pp. 1–6, 2016.

[13] E. Galiotou, P. Fragkou, “Applying linked data technologies to Greek open government data: A case study,” *Procedia-Social and Behavioral Sciences*, vol. 73, pp. 479–486, 2013.

[14] S. Jiang, T. F. Hagelien, M. Natvig and J. Li, “Ontology-based semantic search for open government data,” in *Proc. Int. Conf. on Semantic Computing*, Newport Beach, CA, USA, pp. 7–15, 2019.

[15] P. Escobar, M. d. M. Roldan-Garcia, J. Peral, G. Candela and J. Garcia-Nieto, “An ontology-based framework for publishing and exploiting linked open data: A use case on water resources management,” *Applied Sciences*, vol. 10, no. 3, pp. 779, 2020.

[16] E. F. Kendall and D. L. McGuinness, “Ontology engineering,” *Synthesis Lectures on the Semantic Web: Theory and Technology*, vol. 9, no. 1, pp. i–102, 2019.

[17] A. Abdullah, N. Alazemi, M. Yousef and A. Alfayly, “Challenges of applying semantic web approaches on e-government web services: Survey,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 2, pp. 1216–1224, 2021.

[18] S. Bechhofer, F. Van-Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness *et al.,* “OWL web ontology language reference,” *W3C Recommendation*, vol. 10, no. 2, pp. 1–53, 2004.

[19] S. Kapidakis, “Consistency and interoperability on Dublin core element values in collections harvested using the open archive initiative protocol for metadata harvesting,” in *Proc. Int. Conf. on Knowledge Discovery and Ontology Development*, Portugal, pp. 181–188, 2020.

[20] S. R. Widianto and I. P. E. Warmayudha, “HSQL database,” *Jurnal Mantik*, vol. 4, no. 3, pp. 1717–1721, 2021.

[21] J. Pérez, M. Arenas and C. Gutierrez, “Semantics and complexity of SPARQL,” *ACM Transactions on Database Systems*, vol. 34, no. 3, pp. 1–45, 2009.

[22] R. Unadkat, “Survey paper on semantic web,” *International Journal of Advanced Pervasive and Ubiquitous Computing*, vol. 7, no. 4, pp. 13–17, 2015.

[23] W3C, “OWL 2 web ontology language profiles,” 2012. [Online]. Available: https://www.w3.org/TR/owl2-profiles/#OWL_2_RL_2.