Towards aligning IoT data with domain-specific ontologies through Semantic Web technologies and NLP

Mandeep Singh[1,2], Edlira Vakaj[2], Stamatia Rizou[1] and Wenyan Wu[2]

[1]SingularLogic, Achaias 3, Kifisia 145 64, Greece
[2]Birmingham City University, 15 Bartholomew Row, Birmingham B55JU, UK

Abstract

Internet of Things (IoT) data has the potential to be utilized in many domain-specific applications to enable smart sensing in areas that were not initially covered during the conceptualization phase of these applications. Typically, data collected in IoT scenarios serve a specific purpose and follow heterogeneous data models and domain-specific ontologies. Therefore, IoT data could not easily be integrated into domain-specific applications, as it requires ontology alignment of diverse data models with the end application. This poses a big challenge to semantic interoperability during the integration of IoT data into a pre-established system. In this line, the alignment process is cumbersome and challenging for an ontology engineer, since it requires a manual review of the relevant ontologies that could be aligned with the IoT data. Additionally, before aligning each term used in the IoT data with the concepts defined in the domain-specific ontologies, all similar/related terms in the given ontologies must be considered. In this paper, we propose a solution that supports the alignment process by utilizing semantic web technologies and Natural Language Processing (NLP). Our novel solution proposes an NLP-based term alignment with a similarity score that supports identifying the relevant terms used in IoT data and ontologies and stores the similarity scores among terms based on different similarity algorithms. We showcase our solution by aligning IoT sensor data with the water and IoT domain ontologies.

Keywords

Internet of Things (IoT), Smart Water Network (SWN), Linked Data (LD), ontology, Knowledge Graph (KG), NLP, word2vec, semantic similarity, term alignment

1. Introduction

Around the world, software providers are building Internet of Things (IoT) applications by integrating various solutions and systems that enable remote and continuous monitoring and diagnosis of problems, manage maintenance issues and optimize domain-specific problems by utilizing data-driven and knowledge-driven approaches. The gradual deployment of data-enabled IoT devices, such as smart sensors and actuators, by organizations has offered an

SEMANTICS 2023 EU: 19th International Conference on Semantic Systems, September 20-22, 2023, Leipzig, Germany
msingh@singularlogic.eu;mandeep.singh11@mail.bcu.ac.uk;mandeep.singh@bcu.ac.uk (M. Singh); edlira.vakaj@bcu.ac.uk (E. Vakaj); srizou@singularlogic.eu (S. Rizou); wenyan.wu@bcu.ac.uk (W. Wu)
https://singularlogic.eu (M. Singh); https://www.bcu.ac.uk/computing/about-us/our-staff/edlira-vakaj (E. Vakaj); https://singularlogic.eu (S. Rizou); https://www.bcu.ac.uk/engineering/about-us/our-staff/wenyan-wu (W. Wu)
0000-0001-6381-1140 (M. Singh); 0000-0002-0712-0959 (E. Vakaj); 0000-0002-3683-060X (S. Rizou); 0000-0002-4823-3685 (W. Wu)

opportunity to build a cohesive 'overlay network' in the IoT landscape. Once applications and IoT devices are networked, they can start communicating and exchanging information. However, their interoperability (exchange and making use of information) can not be successful without the syntactic (structure) and semantic (meaning) interoperability of the data/information they share. For instance, at the point of decision-making, a Decision Support System (DSS) relies on the understanding of every bit of data/information that is available from every single IoT device and database, otherwise, it would not be able to advise correctly.

Interoperability of IoT-enabled applications is still a subject of research as cited in [1], one of the main obstacles towards the promotion of IoT adoption and innovation is data interoperability. A key challenge to achieving semantic data interoperability is the alignment of heterogeneous data models among the diverse implementations. Typically, this process needs the calculation of the so-called semantic similarity scores among potential synonymous terms. Despite the existence of similarity calculation algorithms in the literature, this process requires considerable manual work from data workers and analysts to align the application-specific data models to domain-specific ontologies and standards. As a response to this challenge, in this paper, we present our approach that relies on the so-called Semantic Similarity Scoring Ontology (S3O), which automatically identifies the pairs of potential synonymous terms between a given IoT data and existing ontologies and standards and stores their similarity scores to make them available for future reference and reuse. Our approach is expected to significantly improve the efficiency of the cumbersome IoT data alignment process, providing a framework that could be extended to incorporate several ontologies, standards, and similarity score algorithms. To validate our proposed approach, we present a reference implementation of the S3O and will validate its application in an IoT-enabled smart water application.

The rest of the paper is organized as follows. Section 2 discusses the related works in ontology-based semantic interoperability and Natural Language Processing (NLP)-based semantic similarity. Section 3 defines the research questions and proposes our novel approach to address these questions. In section 3.3, the proposed Semantic Similarity Scoring Ontology (S3O) is described. In section 4, we present a showcase of our approach by applying in the water domain for IoT-enabled Smart Water Networks (SWNs).

2. Related Work

Existing approaches to support semantic interoperability in the relevant IoT projects are Smart End-to-end Massive IoT Interoperability, Connectivity and Security (SEMIoTICS) [2] and Bridging the Interoperability Gap of the IoT (BIGIoT) [3]. They propose interoperability solutions that are based on the transitive conversion model for data protocols, e.g., if Message Queuing Telemetry Transport (MQTT) can be converted to/from Constrained Application Protocol (CoAP) and CoAP can be converted to/from Representational State Transfer (REST) than MQTT can be converted to/from REST. Closer to our application scenario in the water domain, a similar interoperability approach is adopted in the water-related projects e.g., Water analytics and Intelligent Sensing for Demand Optimised Management (WISDOM) [4] and Water Enhanced Resource Planning (WatERP) [5], where at first a base ontology (e.g., WISDOM ontology) is aligned with all possible standards and ontologies then it is used to convert from one standard/ontology to

another. Overall, these approaches assume that an ontology of IoT data already exists or has already been adopted. However, this assumption may not hold in real-world scenarios, where IoT data may be re-used in different domain applications, each one using different terms to label their data. Therefore, these works do not address the problem considered in this paper, which also includes the automatic identification of potential synonymous terms among existing ontologies and standards.

Other related works focus on the alignment of similar terms among different dictionaries. In this context, the alignment process of a dictionary or an ontology aims to align the terms used in data models of different applications to achieve semantic interoperability, i.e., find semantic similarity/relatedness of two terms that originate from different ontologies or data models. Since the initiative of Semantic Web (SW), interest in developing and using ontologies for semantic interoperability has grown. This has also led to research approaches for ontology alignment in recent decades. In [6], a survey and comparison of most of the ontology-based similarity/relatedness measures is presented. It also proposes a feature-based similarity measure based on taxonomical features of an ontology to calculate semantic similarity. In [7], a semantic similarity measure based on information distance for ontology alignment is presented. A recently published paper[8] summarizes and compares ontology matching solutions that use the same type of information, and analyses the challenges in different types of information. The list of similarity calculation algorithms is growing with time, as there is no single algorithm to find the perfect semantic match of terms we will need to consider and reason on the similarity score of all possible algorithms during the alignment process. In this context, less attention has been given to a solution that could assist in managing and reasoning the similarity of all computed similarity-based algorithms.

Our work differs in this respect since it introduces an ontology that encompasses different similarity algorithms and provides an abstraction to store pairs of similar terms and their scores, respectively. It enables an ontology engineer to create a linked Knowledge Graph (KG) from multiple domains, such as environment, healthcare, finance, and government while aligning IoT data from different sources with domain-specific ontologies through NLP.

3. Research Questions & Proposed Approach

3.1. Research Questions

Despite the existing work discussed in Section 2, it is evident that there is a lack of automated tools to support the semantic interoperability of IoT data. In this paper, we argue that a holistic approach for the alignment of IoT data to existing ontologies and data models is necessary to support and promote semantic interoperability across the heterogeneous IoT landscape. This holistic approach should provide automated processes for the calculation of the so-called *similarity score* between different terms (i.e., data labels) and should subsequently create a persistent model, expressed through an ontology that will store the relation between the terms under comparison in the associated metadata (including similarity scores, scoring algorithm etc.). The resulting similarity scoring ontology could then be queried to retrieve similarity scores and support fully automated or semi-automated processes for aligning different semantic models. To this end, the research questions addressed in this paper are summarized below:

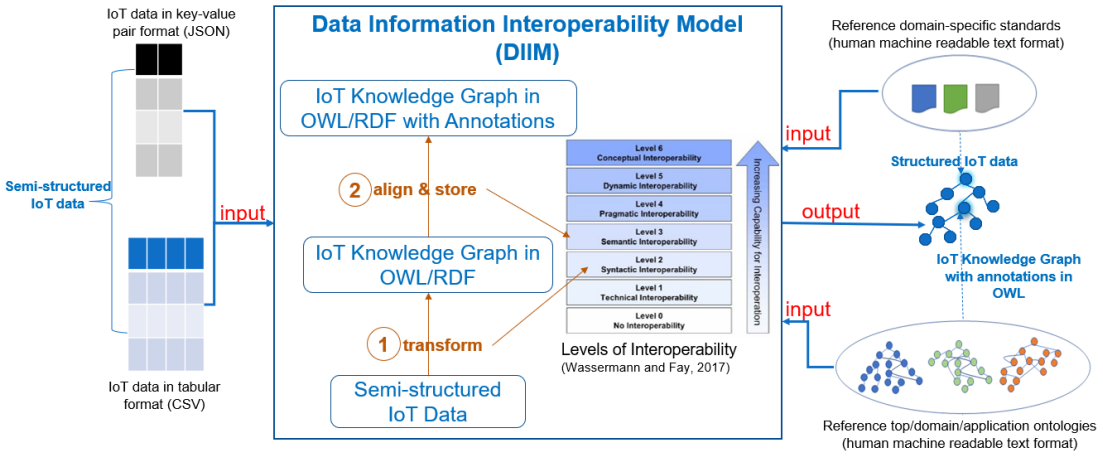


Figure 1: Data and Information Interoperability Model (DIIM)

1. Which terms belonging to existing ontologies or data models can characterize a given entity, e.g., object or attribute name, in an IoT dataset?
2. Given a list of relevant terms identified, which is the semantic similarity score (potentially calculated with different algorithms) between two terms that may potentially refer to the same entity?
3. Given a term associated with an entity, which are the terms that show a semantic similarity score above a given threshold?
4. How the semantic similarity score between two terms could be retrieved efficiently to support recurring queries and avoid score re-calculation?

3.2. Proposed Approach

To answer the above research questions, our approach builds on the conceptual Data and Information Interoperability Model (DIIM) introduced in [9]. As shown in figure 1, Data and Information Interoperability Model (DIIM) takes any IoT dataset and domain-specific ontologies as input and generates an annotated IoT KG as output. DIIM transforms the semi-structured IoT dataset into an IoT KG. It aligns the terms/words (labels to name data/values) of IoT the dataset with the terms/words used in ontologies to describe the concepts, relations, instances, and axioms. Found alignments are stored in the IoT KG as annotations of the respective terms to link the terms to related ontologies. All ontologies must be reviewed regardless of the alignment process nature (manual or computer-assisted). In a computer-assisted alignment, the number of ontologies to be reviewed could become high when all possible alignments are searched in a big repository of ontologies. Here, NLP-assisted alignment of the IoT data with domain-specific applications could be beneficial and aid the semantic communication process. In a computer-assisted alignment process, one finds various alignment algorithms to find term-similarity and

Table 1
Parameters and functions used in algorithms

Name	Description
$dataDir$	The URI of the directory or repository where IoT_{data} or $Ontos_{data}$ are held in Unicode Transformation Format – 8-bit (UTF-8) format.
IoT_{data}	Semi-structured IoT data presented in human-machine readable UTF-8 text format and serialized in JavaScript Object Notation (JSON) or Comma-separated Values (CSV) file.
$Ontos_{data}$	Ontologies described in human-machine readable UTF-8 text format and serialized in a text, Extensible Markup Language (XML), HyperText Markup Language (HTML), Resource Description Framework (RDF), or Web Ontology Language (OWL) file.
$corpora$	It is a collection of human-machine readable UTF-8 files that are processed after reading the resource from a given URI of IoT_{data} or $Ontos_{data}$.
IoT_{corpus}	Corpus of IoT data presented in human-machine readable UTF-8 text file.
$Ontos_{corpus}$	Corpus of ontology presented in human-machine readable UTF-8 text file.
f_{corpus}	This function takes IoT_{data} or $Onto_{data}$ as input and transforms the input to a IoT_{corpus} or $Onto_{corpus}$ that is suitable for NLP operations.

surely new algorithms will be developed as algorithms are case-specific, and we may need to consider different algorithm results in different cases. Therefore, we have a construct that can hold information about the applied algorithms, their calculated similarity score and terms with their reference relation. We propose an ontology instead of a database schema because we can link the terms directly to the original ontologies and reason on similarity calculated by the different algorithms. In this line, our approach includes the following steps:

Step 1 Build corpus from IoT data and ontologies: The first step is to create corpora of IoT data and domain-specific ontologies, so that NLP could be applied to align the terms. Algorithm 1 explains the process of building a corpus from a given Uniform Resource Identifier (URI). A NLP corpus is the textual representation of IoT datasets and ontologies without any digits and special characters. It is built while eliminating digits and special characters and keeping words (text) in the IoT datasets and ontologies as they occur. Table 1 lists the parameters and functions used in defined algorithms. Inputs and outputs are represented through URI

Step 2 Finding potential ontologies for alignment: In this step, we shortlist the ontologies that resemble the given IoT data to save computation time. Because there could be many ontologies for processing, some of them could be aligned and some of them not. If the number of given ontologies is significantly low, then this step could be skipped. In the following algorithm 2, we use Latent Semantic Index (LSI) to find the term-similarity-based relationship between the given IoT dataset and ontologies. However, any other kind of similarity algorithm could be applied to shortlist the relevant ontologies. Gensim library [10] creates a LSI model of each ontology and indexes these models. Finally, the index is compared with the LSI model of IoT data to calculate their similarity/relatedness. The threshold to filter ontologies is an optional parameter. If it is not provided in the input, 1.0 as 100% similarity becomes the default value.

Step 3 Build dictionaries and Word2Vector (Word2Vec) models of IoT data and ontologies: As shown in the algorithm 3, we first build dictionaries (list of used terms) and Word2Vec models (representation of used terms as vectors) from given IoT corpus and ontology corpora

Algorithm 1 Build NLP corpus from given URI

Input $dataDir$ /* URI */
Output $corpora$ /* URI */

procedure f_{corpus}
  $corpora \leftarrow List()$
  for each $file \in dataDir$ **do**
    fileString ← *readFile(file)*
    lowerString ← *lowerCase(fileString)*
    cleanedString ← *removeNumeric(lowerString)*
    cleanedString ← *removeSpecialChars(cleanedString)*
    cleanedString ← *removeStopwords(cleanedString)*
    $corpus \leftarrow tokenize(cleanedString)$
    $corpora.add(corpus)$
  end for
  return $corpora$
end procedure

Algorithm 2 Build LSI from IoT data or ontologies and calculate their similarity score

Input path of IoT_{corpus}, $Onto_{corpus}$ and $threshold$ /* optional parameter*/
Output *pathOfLsiSimilarOntos*

procedure $f_{lsiSimilarity}$
  iotCorpus ← *readCorpus(pathOfIoTCorpus)*
  ontoCorpus ← *readCorpus(pathOfOntoCorpus)*
  ontoLSIModel ← *buildLSIModel(ontoCorpus)*
  ontoLSIIndex ← *buildLSIIndex(ontoLSIModel)*
  iotLSIModel ← *buildLSIModel(iotCorpus)*
  iotOntoLSISimilarity ← *calculateSimilarity(ontoLSIIndex,iotLSIModel)*
  lsiSimilarOntos ← *filterOntologies(iotOntoLSISimilarity,threshold)*
  pathOfLsiSimilarOntos ← *writeFile(lsiSimilarOntos)*
  return *pathOfLsiSimilarOntos.*
end procedure

(list of corpus). Then, we train Word2Vec models of ontologies with given IoT corpus.

Step 4 Calculate algorithm-based similarity score of terms in IoT data and ontologies: In this step, we want to calculate the algorithm-based similarity of each term in IoT data with terms used in given ontologies. In algorithm 4 we use Word2Vec similarity and String-search Matching (SSM) algorithms to demonstrate the similarity calculation procedure. Hence, any other similarity calculation algorithms of choice can be added to the procedure to have preferred results. As output, $iotOntoW2vSimilarityList$ and $iotOntoSsmSimiarity$ lists have all terms of IoT data and ontologies, and their calculated Word2Vec and String-search Matching (SSM) similarity score.

Step 5 Build an ontology and store algorithm-based similarity score: In the final step,

Algorithm 3 Build Word2Vec models and dictionaries of IoT data and ontologies

Input $iotCorpora$ and $ontosCorpora$

Output $iotW2vModels$, $iotDicts$, $ontosW2vModels$, $ontosDicts$, $iotOntosW2vModels$, and $allIotOntosW2vModels$

procedure f_{w2v}
  $ontosW2vModels \leftarrow$ *buildW2VModel(ontosCorpora)*
  iotW2vModels $\leftarrow buildW2VModel(iotCorpora)$
  $ontosDicts \leftarrow buildDict(ontosW2VModels)$
  $iotDicts \leftarrow buildDict(iotW2VModels)$
  $allIotOntosW2vModels \leftarrow List()$
  for each $iotCorpus \in iotCorpora$ **do**
    $iotOntosW2vModels \leftarrow List()$
    for each $ontoW2vModel \in ontoW2vModels$ **do**
      $iotOntoW2vModel \leftarrow train(ontoW2vModel, iotCorpus)$
      $iotOntosW2vModels.add(ontoWIotW2vModel)$
    end for
    $allIotOntosW2vModels.add(iotOntosW2vModels)$
  end for
  return $iotW2vModels$, $iotDicts$, $ontosW2vModels$, $ontosDicts$, $iotOntosW2vModels$, $allIotOntosW2vModels$
end procedure

we build an ontology that stores the information on the similarity of IoT data with an ontology. It also stores the applied algorithm-based similarity score as a relation between two terms that are used in IoT data and ontology. The following algorithm 5 depicts the procedure to populate the ontology with facts and create similarity relationships among entities. S3O contains all terms of IoT data and ontologies, and their calculated Word2Vec and SSM similarity score. Additionally, it also contains the LSI similarity score of given IoT data in relation to ontologies.

3.3. Semantic Similarity Scoring Ontology

In this section, we describe the proposed Semantic Similarity Scoring Ontology (S3O) [11] that stores the terms used in IoT data and ontologies, and stores the similarity score based on the applied various algorithms. Additionally, it stores the directly calculated similarity between any IoT data and an ontology. S3O covers all research questions for aligning the IoT terms with domain-specific ontologies. When S3O is loaded and populated with re facts it will hold information to answer the question from section 3.1.

Figure 2 displays the S3O ontology that was developed in Protégé [12]. S3O ontology starts with an abstract class *Thing*. It has two data properties, *serialization_format* to represent the data representation format and *URI* for identification, that are inherited by its subclasses. Its direct subclasses are *Document*, *Term*, and *Similarity*. *Term* class represents a word or phrase used to describe a thing or express a concept used in any IoT data or an ontology. *Document* class represents an object of the text sequence type. In our approach, we identify *IoT_data* and

Algorithm 4 Calculate algorithm-based similarity score of terms in IoT data and ontologies

Input $iotDicts$, $ontosDicts$ $iotOntosW2vModels$

Output $iotOntoW2vSimilarityList$ and $iotOntoSsmSimiarityLlist$

procedure f_{sim}
  $iotOntoW2vSimilarityList \leftarrow List()$
  $iotOntoSsmSimiarityList \leftarrow List()$
/* SSM-based similarity calculation*/
  for each $iotDict \in iotDicts$ **do**
    for each $iotTerm \in iotDict$ **do**
      for each $ontoDict \in ontoDicts$ **do**
        $iotOntoSsmTermSimiarity \leftarrow List()$
        for each $ontoTerm \in ontoDict$ **do**
          $iotOntoSsmSimilarity \leftarrow calculateSsm(iotTerm, ontoTerm)$
          $iotOntoSsmTermSimiarity.add(iotOntoSsmSimilarity)$
        end for
        $iotOntoSsmSimiarityList.add(iotOntoSsmTermSimiarity)$
      end for
    end for
  end for
/* Word2Vec-based similarity calculation*/
  for each $iotDict \in iotDicts$ **do**
    for each $iotTerm \in iotDict$ **do**
      $iotOntoW2vTermSimiarity \leftarrow List()$
      for each $iotOntosW2vModel \in iotOntosW2vModels$ **do**
        $iotOntoW2vSimilarity \leftarrow$
$calculateW2vSimilarity(iotTerm, iotOntosW2vModel)$
        $iotOntoW2vTermSimiarity.add(iotOntoW2vSimilarity)$
      end for
      $iotOntoW2vSimilarityList.add(iotOntoW2vTermSimiarity)$
    end for
  end for
  return $iotOntoW2vSimilarityList$, $iotOntoSsmSimiarityList$
end procedure

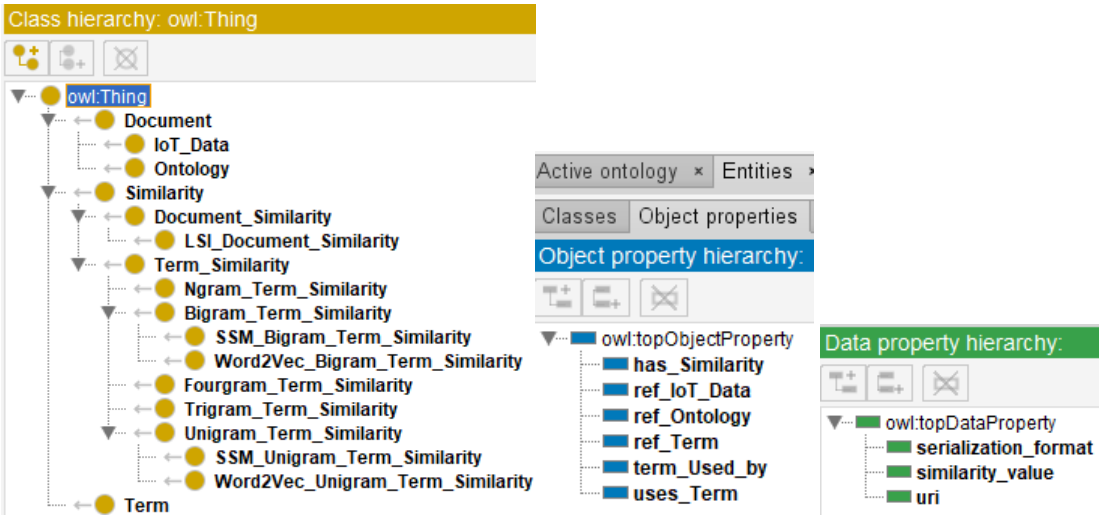
Ontology as document objects for NLP. *IoT_data* contains *metadata* and *measurement data* that can be accessed from any IoT device. The *Ontology* class contains the terms, relations, and properties used to describe the data, information, and knowledge of a specific domain application. The *Similarity* abstract class abstracts over all similarity algorithms, e.g., *SSM_Similarity Word2Vec_Similarity* that are applied to calculate the similarity of documents and terms. It holds *similarity_value* data property to store the similarity score of the documents or terms. We introduce *Ngram* classes to store the similarity of combined terms, e.g., *temperature sensor* (bigrams) that appear as sequences of words in IoT data and ontologies. We have introduced

Algorithm 5 Store the algorithm-based similarity score in an ontology

Input $iotDicts$, $ontosDicts$, $iotOntoW2vSimilarityList$, $iotOntoLSISimilarity$, $iotOntoSsmSimiarity$, and S3O

Output S3O

1: **procedure** f_{s3o}
2: *s3o* ← *loadS3OSchema()*
3: *s3o* ← *createTermRelations(s3o,iotDicts,ontosDicts)*
4: *s3o* ← *createSimilarityRelations(s3o,iotOntoW2vSimilarityList)*
5: *s3o* ← *createSimilarityRelations(iotOntoLSISimilarity)*
6: *s3o* ← *createSimilarityRelations(s3o,iotOntoSsmSimiarity)*
7: *pathOfS3o* ← *writeFile(s3o)*
8: **return** *pathOfS3o.*
9: **end procedure**



(a) Class entities **(b)** Object properties **(c)** Data properties

Figure 2: Schema of S3O

object property *has_Similarity* to store the information on similarity-based relationships among *IoT_Data*, *Ontology*, and *Term* classes. *ref_IoT_Data*, *ref_Ontology*, and *ref_Term* are object properties for references. *term_Used_by* and *uses_Term* are inverse object properties to store the information when the term is used by IoT-data or ontologies.

4. Showcase implementation for Smart Water Networks (SWN) applications

To showcase the validity of our approach, we have created a reference implementation [13] of the proposed approach described in Section 3.2 and tested this in a specific scenario for IoT data

characterizing a Smart Water Network (SWN) application.

Implementation Setup: The solution was developed in Visual Code Studio[14] Integrated Development Environment (IDE). For the implementation of the approach, Python [15] and many Python-based NLP libraries, e.g. Gensim [10] for Word2Vector (Word2Vec) and Latent Semantic Index (LSI), import Matplotlib[16] for visualization, Pandas[17] for data storage and retrieval, RDFLib [18] for processing S3O, were utilized. Protégé [12], an ontology development environment tool, is used to author and examine S3O ontology facts on term similarity written in Turtle RDF serialization format. Pellet[19] reasoner is used to reason the S3O. Snap SPARQL Protocol and RDF Query Language (SPARQL) Query plugin [20] for Protégé are used to query the S3O facts.

Showcase characteristics: The showcase application takes two inputs, IoT data and domain-specific ontologies of the water and IoT domain. In particular, we consider publicly accessible data sets, related to water quality data. The showcase application processed the IoT data serialized in formats CSV or JSON. More information about the IoT data set characteristics can be found in Table 2. Table 3 holds the information on the ontologies used as input. Input ontologies are from the upper, water, biological, or water domain. The showcase application processed these ontologies from serialization formats, e.g., text, XML, HTML, RDF, or OWL. S3O schema was developed in Protégé and exported as an RDF file. RDFlib [18] was used to generate a graph by loading S3O schema and populating it with the facts/information that is computed by the showcase program on the terms of IoT and ontologies with their relations and algorithm-based similarity score. We use descriptive-naming-pattern $< algorithmname >\_< ontology{-}name >\_< ontologyterm{-}name >\_< IoT{-}name >\_< IoT{-}term{-}name >$ for the similarity class instances. For example, in $SSM\_SensorML\_counts\_BristolWaterQuality\_units$ instance, SSM algorithm is used to calculate the similarity of $counts$ from $SensorML$ ontology and $units$ from $BristolWaterQuality$ data.

Table 2
Input: IoT data

File Name	Bristol River water quality.csv [21]	Kaa IoT Data [22]
Format	CSV	JSON
License	Open Government Licence	Public access
Topic	Water Quality	Water temperature
Summary	River quality monitoring data (chemical, physical and bacteriological parameters tested) from 1994.	The data holds the values of a temperature sensor that was simulated locally to send data to the KaaIoT cloud.
Extracted words	access, allison, ammonium, annes, apr, ashton, aug, avenue, avonmouth, badocks, boiling, bottom, bright, briscoe, briscoes, … [155]	auto, bfg, description, fahrenheit, latitude, longitude, mac, measures, model, name, sensor, serial, temperature, timestamp, value, values, water, [17]

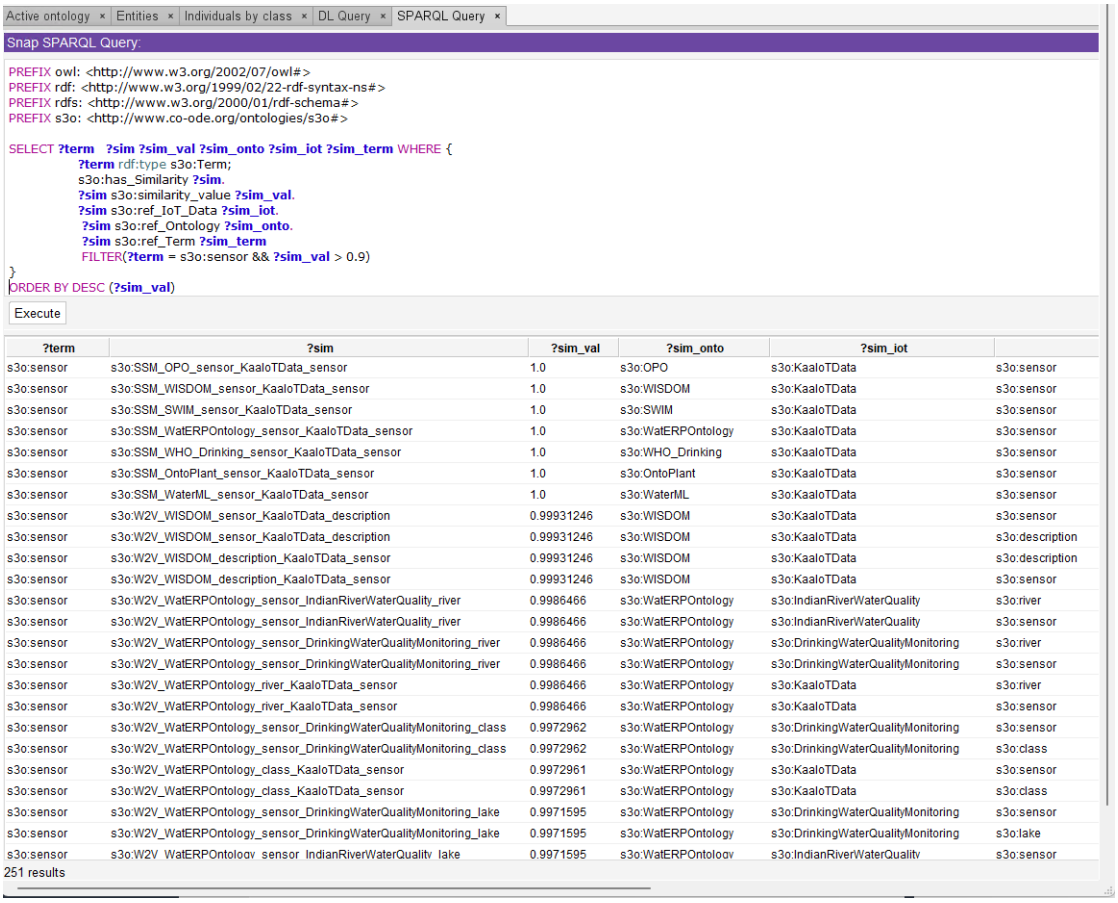
As primary output, the showcase application generates the S3O in RDF turtle format. Other outputs of the showcase application are persisting all calculated information in pickle files and

Table 3
Input: domain-specific ontologies and standards

Name, **Domain**	**Description**
COSMO[23], upper	It is a foundation ontology that allows it to represent all the basic ('primitive') ontology elements of an application.
DOLCE[24], upper	It is a descriptive ontology for linguistic and cognitive engineering.
GO[25], biological	It provides the foundation for computational analysis of large-scale molecular biology and genetics experiments in biomedical research.
GOIoTP[26], IoT	GOIoTP is developed as part of the INTER-IoT project; it offers modular data structures for the description of entities most commonly appearing in IoT in the context of interoperating various IoT artefacts (platforms, devices, services, etc).
INSPIRE[27], upper	Representation of a set of concepts within a domain and the relationships between those concepts
OntoPlant[28], water	Sottara et al. have extended the SSN ontology to decouple control logic from equipment choices in wastewater treatment plants.
OPO[29], water	It is an Observational Process Ontology for water quality monitoring.
SAREF[30], IoT	It is a shared consensus model that facilitates the matching of existing assets in the smart applications domain.
SensorML[31], IoT	It provides a robust and semantically-tied means of defining processes and processing components associated with the measurement and post-measurement transformation of observations.
SSN[32], IoT	This ontology describes sensors and sensor networks, for use in web applications, independent of any application domain.
SOSA[33], IoT	It can be used directly for lightweight applications, or provide the basis for additional specialization and axiomatization in vertical and horizontal extensions.
SWIM[34], water, IoT	It is developed by Aquamatix for the Device-level IoT semantic model for the water industry
WaterML[35], water	WaterML2 is a new data exchange standard in Hydrology to exchange many kinds of hydro-meteorological observations and measurements. It harmonizes a number of exchange formats for water data with relevant OGC and ISO standards.
WatERPOntology[5], water	It is developed by EURECAT-WatERP. It is a lightweight ontology of generic concepts for water sensing and management.
WHO_Drinking[36], water	WHO standard guidelines to maintain the relevance, quality and integrity of the Guidelines for drinking-water quality (GDWQ), whilst ensuring their continuing development in response to new, or newly-appreciated, information and challenges.
WISDOM[4], water	It is developed by Cardiff University for the cyber-physical and social ontology of the water value chain.

creating bar charts of the top 10 Word2Vec-based on similar terms of each IoT-term.

Querying similarity in S3O: Figure 3 shows an output of querying the S3O with facts [11] for the term "sensor". In particular, a SPARQL query [11] is executed in the Snap SPARQL [20], Query tab of Protégé provides results on similar terms to term "sensor" as shown in the figure 3. *Protégé*[12], an ontology development environment tool, is used to load S3O facts on term



Active ontology × | Entities × | Individuals by class × | DL Query × | SPARQL Query ×

Snap SPARQL Query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX s3o: <http://www.co-ode.org/ontologies/s3o#>

SELECT ?term   ?sim ?sim_val ?sim_onto ?sim_iot ?sim_term WHERE {
            ?term rdf:type s3o:Term;
            s3o:has_Similarity ?sim.
            ?sim s3o:similarity_value ?sim_val.
            ?sim s3o:ref_IoT_Data ?sim_iot.
             ?sim s3o:ref_Ontology ?sim_onto.
             ?sim s3o:ref_Term ?sim_term
             FILTER(?term = s3o:sensor && ?sim_val > 0.9)
}
ORDER BY DESC (?sim_val)
```

Execute

?term	?sim	?sim_val	?sim_onto	?sim_iot	
s3o:sensor	s3o:SSM_OPO_sensor_KaaloTData_sensor	1.0	s3o:OPO	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:SSM_WISDOM_sensor_KaaloTData_sensor	1.0	s3o:WISDOM	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:SSM_SWIM_sensor_KaaloTData_sensor	1.0	s3o:SWIM	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:SSM_WatERPOntology_sensor_KaaloTData_sensor	1.0	s3o:WatERPOntology	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:SSM_WHO_Drinking_sensor_KaaloTData_sensor	1.0	s3o:WHO_Drinking	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:SSM_OntoPlant_sensor_KaaloTData_sensor	1.0	s3o:OntoPlant	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:SSM_WaterML_sensor_KaaloTData_sensor	1.0	s3o:WaterML	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:W2V_WISDOM_sensor_KaaloTData_description	0.99931246	s3o:WISDOM	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:W2V_WISDOM_sensor_KaaloTData_description	0.99931246	s3o:WISDOM	s3o:KaaloTData	s3o:description
s3o:sensor	s3o:W2V_WISDOM_description_KaaloTData_sensor	0.99931246	s3o:WISDOM	s3o:KaaloTData	s3o:description
s3o:sensor	s3o:W2V_WISDOM_description_KaaloTData_sensor	0.99931246	s3o:WISDOM	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:W2V_WatERPOntology_sensor_IndianRiverWaterQuality_river	0.9986466	s3o:WatERPOntology	s3o:IndianRiverWaterQuality	s3o:river
s3o:sensor	s3o:W2V_WatERPOntology_sensor_IndianRiverWaterQuality_river	0.9986466	s3o:WatERPOntology	s3o:IndianRiverWaterQuality	s3o:sensor
s3o:sensor	s3o:W2V_WatERPOntology_sensor_DrinkingWaterQualityMonitoring_river	0.9986466	s3o:WatERPOntology	s3o:DrinkingWaterQualityMonitoring	s3o:river
s3o:sensor	s3o:W2V_WatERPOntology_sensor_DrinkingWaterQualityMonitoring_river	0.9986466	s3o:WatERPOntology	s3o:DrinkingWaterQualityMonitoring	s3o:sensor
s3o:sensor	s3o:W2V_WatERPOntology_river_KaaloTData_sensor	0.9986466	s3o:WatERPOntology	s3o:KaaloTData	s3o:river
s3o:sensor	s3o:W2V_WatERPOntology_river_KaaloTData_sensor	0.9986466	s3o:WatERPOntology	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:W2V_WatERPOntology_sensor_DrinkingWaterQualityMonitoring_class	0.9972962	s3o:WatERPOntology	s3o:DrinkingWaterQualityMonitoring	s3o:sensor
s3o:sensor	s3o:W2V_WatERPOntology_sensor_DrinkingWaterQualityMonitoring_class	0.9972962	s3o:WatERPOntology	s3o:DrinkingWaterQualityMonitoring	s3o:class
s3o:sensor	s3o:W2V_WatERPOntology_class_KaaloTData_sensor	0.9972961	s3o:WatERPOntology	s3o:KaaloTData	s3o:sensor
s3o:sensor	s3o:W2V_WatERPOntology_class_KaaloTData_sensor	0.9972961	s3o:WatERPOntology	s3o:KaaloTData	s3o:class
s3o:sensor	s3o:W2V_WatERPOntology_sensor_DrinkingWaterQualityMonitoring_lake	0.9971595	s3o:WatERPOntology	s3o:DrinkingWaterQualityMonitoring	s3o:sensor
s3o:sensor	s3o:W2V_WatERPOntology_sensor_DrinkingWaterQualityMonitoring_lake	0.9971595	s3o:WatERPOntology	s3o:DrinkingWaterQualityMonitoring	s3o:lake
s3o:sensor	s3o:W2V_WatERPOntology_sensor_IndianRiverWaterQuality_lake	0.9971595	s3o:WatERPOntology	s3o:IndianRiverWaterQuality	s3o:sensor

251 results

Figure 3: Find terms similar to the term "sensor" in ontologies and IoT datasets

similarity written in turtle RDF serialization format. *pellet*[19] reasoner is used to reason the S3O. *Snap SPARQL Protocol and RDF Query Language (SPARQL) Query* plugin for *Protégé* is used to query the S3O facts.

Discussion: In this section, we have described the reference implementation of our approach presented in Subsection 3.2 and have demonstrated its applicability considering public IoT data sets and relevant domain-specific ontologies in the water domain. More specifically, our approach uses NLP as the core method to define and calculate the semantic similarity scores.

We start with term/word alignment because in description logic words are used to describe and to label/name values in datasets and entities in ontologies. Therefore, we can also do structural alignment based on similar/related words when we could deduce structural alignment through N-grams, e.g., *'water has ph'* can be deduced to align with the statement *'water contains ph'* if *'has'* and *'contains'* can be aligned. While following the Data and Information Interoperability Model (DIIM) approach, we want to annotate the terms in IoT KG with similar words found in different domain-specific ontologies and standards. To achieve this, we first try to find similar words with this approach and annotate them when the similarity score is higher than the

given threshold. At the current stage, only a similarity score with a value of 1 is automatically accepted for auto-annotation and all other similar terms with lower scores are suggested for a human review, which poses a challenge to the manual effort in the alignment process. However, the proposed approach to use NLP-based techniques and accommodate different algorithms in combination with S3O significantly ease the alignment process.

The experiments showed that our approach performs well and manages to create the S3O and subsequently calculate and store the similarity scores for the identified terms in the IoT data sets. We propose S3O instead of database schema because we want to use import-feature to link the terms directly to the original ontologies and IoT data converted into KGs and reason on similarity calculated by the different algorithms as federated KG as whole. S3O covers the current requirements and is subject to extension for new requirements.

Further implementation and experiments are planned to measure the performance of our solution (in terms of time) as well as in terms of precision (i.e., compare the output of our solution with respect to the identification of similar terms using a manual process in small-scale scenarios). Overall, we consider that our solution is an initial step towards systematizing and automating the process of semantic interoperability in the heterogeneous IoT landscape.

5. Conclusion

In this paper, we propose a *novel methodology based on Semantic Web technologies (OWL, KG, RDF, and Linked Data (LD)) and NLP (LSI, Word2Vec and Ngram similarity) to discover related ontologies and align terms of IoT data with these ontologies.* Further, our work contributes to developing a new ontology, the Semantic Similarity Scoring Ontology (S3O). The proposed S3O holds a similarity score of terms based on the similarity evaluation of the applied algorithms. This ontology can be easily extended to include the evaluation results of other algorithms. This way, we do not support a specific algorithm to align terms, rather believe that all alignment algorithms could become relevant at a certain point. Therefore, we store the similarity scores of all alignment algorithms in S3O and an ontology engineer can query the similarity scores explored the linked terms from different ontologies and decide to do the final alignment/mapping. We have showcased the validity of our approach in an IoT-enabled smart water application, however, the proposed solution is extensible in terms of adding new ontologies for alignment and considering newly developed term-alignment algorithms. In our future work, we plan to extend the implementation of the showcase by adding more NLP-based similarity algorithms to support the alignment of the IoT data with the ontologies of the cross-domain applications.

References

[1] N. Kalatzis, G. Routis, Y. Marinellis, M. Avgeris, I. Roussaki, S. Papavassiliou, M. Anagnostou, Semantic interoperability for IoT platforms in support of decision making: An experiment on early wildfire detection, Sensors (Switzerland) 19 (2019). doi:10.3390/s19030528.

[2] G. M. Milis, C. G. Panayiotou, M. M. Polycarpou, Semiotics: Semantically enhanced iot-enabled intelligent control systems, IEEE Internet of Things Journal 6 (2019) 1257–1266.
[3] A. Bröring, S. Schmid, C. Schindhelm, A. Khelil, S. Käbisch, D. Kramer, D. Le Phuoc, J. Mitic, D. Anicic, E. Teniente, Enabling iot ecosystems through platform interoperability, IEEE Software 34 (2017) 54–61.
[4] S. Howell, Y. Rezgui, T. Beach, Automation in Construction Integrating building and urban semantics to empower smart water solutions, Automation in Construction 81 (2017) 434–448. doi:10.1016/j.autcon.2017.02.004.
[5] G. Anzaldi Varas, W. Wu, A. Abecker, E. Rubión Soler, A. Corchero, A. Hussain, M. QUENZER, Integration of water supply distribution systems by using interoperable standards to make effective decisions, 2014.
[6] D. Sánchez, M. Batet, D. Isern, A. Valls, Ontology-based semantic similarity: A new feature-based approach, Expert Systems with Applications 39 (2012) 7718–7728. URL: https://www.sciencedirect.com/science/article/pii/S0957417412000954. doi:https://doi.org/10.1016/j.eswa.2012.01.082.
[7] Y. Jiang, X. Wang, H.-T. Zheng, A semantic similarity measure based on information distance for ontology alignment, Information Sciences 278 (2014) 76–87. URL: https://www.sciencedirect.com/science/article/pii/S0020025514003053. doi:https://doi.org/10.1016/j.ins.2014.03.021.
[8] X. Liu, Q. Tong, X. Liu, Z. Qin, Ontology matching: State of the art, future challenges, and thinking based on utilized information, IEEE Access 9 (2021) 91235–91243. doi:10.1109/ACCESS.2021.3057081.
[9] M. Singh, W. Wu, S. Rizou, E. Vakaj, Data information interoperability model for iot-enabled smart water networks, in: 2022 IEEE 16th International Conference on Semantic Computing (ICSC), 2022, pp. 179–186. doi:10.1109/ICSC52841.2022.00038.
[10] R. Řehůřek, Gensim, 2009. URL: https://radimrehurek.com/gensim/index.html, accessed 2 Mar 2022.
[11] M. Singh, Semantic similarity scoring ontology, 2023. URL: https://github.com/mxrandhawa/s3o, accessed: 04/05/2023.
[12] N. F. Noy, M. Crubézy, R. W. Fergerson, H. Knublauch, S. W. Tu, J. Vendetti, M. A. Musen, Protégé-2000: an open-source ontology-development and knowledge-acquisition environment., in: AMIA... Annual Symposium proceedings. AMIA Symposium, volume 2003, American Medical Informatics Association, 2003, pp. 953–953.
[13] M. Singh, Iot showcase implementation with s3o ontology, 2023. URL: https://github.com/mxrandhawa/iotshowcase, accessed: 08/05/2023.
[14] Microsoft©2022, Visual code studio, 2015. URL: https://code.visualstudio.com/, accessed 7 Feb 2022.
[15] P. S. Foundation, Python™, 2001. URL: https://www.python.org/, accessed 7 Feb 2022.
[16] Matplotlib Development team, Matplotlib: Visualization with python, 2003. URL: https://matplotlib.org/, accessed 8 Feb 2022.
[17] NumFOCUS, Inc., Pandas: Visualization with python, 2020. URL: https://pandas.pydata.org//, accessed 17 Dec 2022.
[18] RDFLib Team, Rdflib, 2009. URL: https://rdflib.readthedocs.io/en/stable/, accessed 8 Feb 2022.

[19] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical owl-dl reasoner, Journal of Web Semantics 5 (2007) 51–53.
[20] M. Horridge, M. Musen, Snap-sparql: a java framework for working with sparql and owl, in: Ontology Engineering: 12th International Experiences and Directions Workshop on OWL, OWLED 2015, co-located with ISWC 2015, Bethlehem, PA, USA, October 9-10, 2015, Revised Selected Papers 12, Springer, 2016, pp. 154–165.
[21] B. C. Council, Bristol river water quality, 2010. URL: https://www.data.gov.uk/dataset/dd6658fc-7d1a-4ab2-9ea4-6aa936d21608/bristol-river-water-quality, accessed: 26/09/2022.
[22] M. Singh, Kaa cloud iot sensor data, 2021. URL: https://www.kaaiot.com/, accessed: 27/12/2021.
[23] P. Cassidy, Cosmo, 2020. URL: http://micra.com/COSMO/, accessed 23 Jan 2021.
[24] N. Guarino, A. Gangemi, Dolce-ultralite, 2017. URL: www.loa.istc.cnr.it/dolce/overview.html, accessed 23 Jan 2021.
[25] G. Consortium, Gene ontology, 2020. URL: http://geneontology.org/.
[26] P. A. o. S. Paweł Szmeja, Systems Research Institute, Generic ontology for iot platforms, 2018. URL: https://inter-iot.github.io/ontology/.
[27] I. Maintenance, Implementation, Inspire ontology, 2015. URL: https://inspire.ec.europa.eu/glossary/Ontology, accessed 10 Dec. 2022.
[28] D. Sottara, J. C. Coreale, T. Spetebroot, D. Pulcini, D. Giunchi, F. Paolucci, L. Luccarini, An ontology-based approach for the instrumentation, control and automation infrastructure of a wwtp, 2014. URL: https://www.academia.edu/download/53892589/An_ontology-based_approach_for_the_instr20170718-3232-1pm8ts1.pdf.
[29] X. Wang, H. Wei, N. Chen, X. He, Z. Tian, An observational process ontology-based modeling approach for water quality monitoring, Water 12 (2020) 715. doi:10.3390/w12030715.
[30] E. STF-578, Smart applications reference ontology, and extensions, 2020. URL: https://saref.etsi.org/, accessed 10 Dec. 2022.
[31] OGC, Sensor model language (sensorml), 2019. URL: https://www.ogc.org/standards/sensorml, accessed 1 Dec. 2022.
[32] W. OGC, Semantic sensor network ontology, 2017. URL: https://www.w3.org/TR/vocab-ssn/, accessed 9 March 2022.
[33] W3C, Sensor-observation-sampling-actuator ontology, 2016. URL: https://www.w3.org/2015/spatial/wiki/SOSA_Ontology, accessed 19 August 2022.
[34] L. Reynolds, The swim concept: an open interoperable data standard, in: IET Conference Proceedings, The Institution of Engineering & Technology, 2013.
[35] H. D. W. g. OGC, Waterml2, 2014. URL: http://waterml2.org/, accessed 23 Jan 2021.
[36] W. H. Organization, Guidelines for drinking-water quality (gdwq), 2021. URL: https://www.who.int/publications-detail-redirect/9789240045064, accessed 10 Jan. 2022.