DOMAIN INTEROPERABILITY FOR IOT APPLICATION THROUGH KNOWLEDGE EXTRACTION

S.Prakash

Research Scholar, School of Computing Science and Engineering, Galgotias University,
Greater Noida, Uttar Pradesh 201310, India
research4prakash@gmail.com

Dr. T.Poongodi

Associate Professor, School of Computing Science and Engineering, Galgotias University,
Greater Noida, Uttar Pradesh 201310, India
tpoongodi2730@gmail.com

Abstract

The recent IoT applications belong to various domains like smart city, weather forecasting, healthcare, transportation, etc. Generally, the domain knowledge is represented by a domain ontology which is designed by RDF, RDF-S, and OWL. A machine can understand the domain ontology, but the machine cannot understand the cross-domain knowledge in IoT applications because of interoperability issues. There are no standards available to describe all IoT-related transport domain terms. Due to a lack of interoperability, machines are not capable of processing various domain knowledge. In the proposed system, the ontological terms are converted into vectors using Google word2vector algorithms and the machine can process these vectors efficiently. The vectorized transport terms are clustered using their semantic similarity distance. Moreover, the contribution is made in terms of clustering in the domain of transportation. The transport domain terms are extracted based on 1000 research papers and 90 transportation domain ontologies.

***Keywords*: Knowledge Extraction; Internet of Things (IoT); Semantic Web; Web of Things (WoT); Natural Language Processing (NLP); Machine Learning (ML).**

1. Introduction

Web of Things (WoT) is an emerging technology in present-day remote correspondence. The things are associated with neighbor items or neighbor sensor hubs. The figure 1 shows the various visions of IoTwhich goes about as a significant part taking all things together kind of professional food industry, training, medical care, transportation, and climate anticipating. IoT is a package of "Thing", "Web", and Semantic. "Thing"- focused dreams comprises Radio Frequency Identification (RFID), UID, spimes, shrewd things, NFC, regular items, remote sensors, actuators, and WISP. "Web"- arranged dreams that consist of IP for savvy protests, the Web, and the trap of things. "Semantic"- situated goals it comprises Resource Description Framework (RDF), RDF-Schema, Web Onology Language (OWL), semantic advances, philosophy loader, metaphysics reasoner, and semantic execution conditions [1].

Arduino supports pluggable augmentation board (Wi-Fi, GPRS, BLE, ZigBee), Raspberry Pi gives the interfacing office Ethernet, expansion through USB, BLE (Pi3), Intel gives the associating office Wi-Fi and BLE, BeagleBoard underpins the Ethernet, augmentation through USB and shields, Texas Instruments bolsters the availability Wi-Fi, BLE, and ZigBee, Broadcom upholds the network Wi-Fi, BLE, and ZigBee [2]. Machine Learning (ML) algorithm calculations consequently figure out how to act as needs be. The unsupervised K-means algorithm calculation groups the items utilizing their distance and incentive inside the bunch; the terms are more comparative in contrast with another cluster [3]. Natural Language Processing provides the functionality where the machines can extract the required knowledge from documents or websites. The word2vec calculation comprises a different sort of learning model. The first is the Continuous Bag of Words (CBOW) model and the subsequent one is Skip-gram (SG) model. The texts are changed over into multidimensional vectors utilizing the word2vec calculation [4]. The paper is organized in a way that section 2 presents the literature survey, section 3 describes the proposed methodology, section 4 discusses dataset preparation, Transport corpus creation procedures are presented in section 5, section 6 describes the experimental analysis, section 7 explains the result analysis. In section 8, we conclude the article with a discussion of the proposed work and future directions.

DOI : 10.21817/indjcse/2022/v13i1/221301045

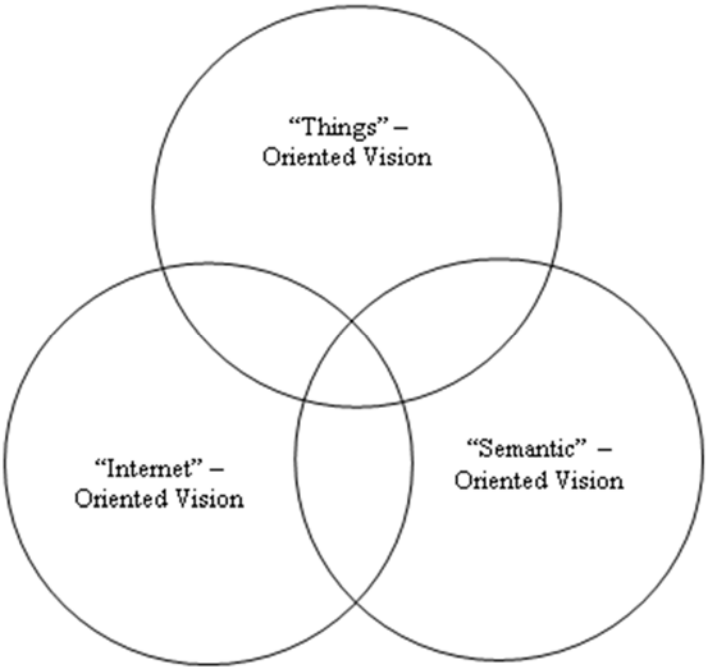

Fig. 1. Visions of IoT

2. Literature Review

Describes the contextual interoperability in this model. It portrays context information minimum error, mean error, timestamp, and quality. The ontology reasoner is not enough to find the relationships in the application. The intelligent techniques are not applied [5]. IoT Domain model is created with the help of domain-specific programming language. The metadata and properties are mentioned in the device object model. This model does not reuse the existing IoT application semantics. So, the scalability of this technique is questionable [6]. This paper provides the direction of ontology interaction especially highlights the ontology parameter coherence, consistency, and their relationships. And also suggests the principles of ontology parameter coherence [7]. This article highlights W3C does not have domain-specific knowledge. Thing Directory (TD) vocabulary is JSON-LD format for domain-specific vocabulary implemented with user assistance. The gap analysis is, this technique is not appropriate for all domains which require some automation in the translation of vocabulary into TD [8]. Address the device incompatibility problems for Enterprise Framework [9]. Open cross-layer framework is implemented for solving interoperability and encourages the semantic web provides interoperability for Global standard and also motivates the reuse of W3C SSN ontology [10]. User Interface is implemented between doctor and patient. Data Analytics techniques are used to predict patterns in medical data; those patterns are annotated in ontologies. The guidelines of ontology concept annotation are missing and not following the standard w3c ontologies [11].

In this article, the devices, protocols, and profiles interoperability issues were discussed. For the huge volume of data, intelligent algorithms are utilized. The NLP and Text Analytics found the fire accidents key events, frequently used words, and reasons are found. The IoT devices automatically found the record of the fire accidents using that analyze the reasons. The fire accident is interconnected with time, weather, and location and this technique tried to provide data interoperability [12, 13]. Underline some of the reused ontology research papers and highlights the ontologies Semantic Sensor Network (SSN) ontology, SensorML ontology, Service Ontology, Device Ontology, FOAF Ontology, IoT Ontology, OntoSensor Ontology, Semantic Actuator Network (SAN) ontology, and Machine-to-Machine Measurement (M3) Ontology [14]. Provided the direction about future intelligent algorithms and standards in IoT [15]. The product ontology is attached to the product for making interoperability [16]. Ontology analyzes tool, checks the size, structure, and specific taxonomic structure of ontology and plugin into the proteges. Presently many query languages DL Query, SPARQL, and SWRL are available to perform this task [17]. The semantic web gives a typical system that permits information to be shared and reused across community boundaries, applications, and endeavors. The Semantic Web is in this manner viewed as an integrator across various substances and data applications and frameworks [18]. Giancarlo et al. described the features of device abstraction, interface abstraction, multiple data sources, device management, context-awareness, security, and privacy [19].

IoT ontology describes the reusability of existing SSN, DUL, QUDIT, and FOAF ontology and extending their features with new IoT environment sensors, motion sensors, human body sensors, motion sensors, RFID sensors, visual sensors, and temperature vendors [20]. Lightweight ontology portrays the reusability of W3C/OGC SSN ontology to overcome heterogeneity issues. The created ontology is available online, but this ontology has parser exception error [21]. SOSA ontology is the replacement of core SSN ontology created by domain experts [22]. The framework is designed for interoperability among IoT platforms. The fundamental problem of semantic web issue was highlighted that is many ontologies files are inaccessible. But how to offer the interoperability service among various domains is missing [23]. Catalog Linked Open Vocabularies tagged 400 ontologies and described 200 ontology-related projects relevant to the Internet of Things. Suggest M3 ontology, M3 dataset, and M3 rules for IoT application [24]. IoT-O ontology designed for to annotate devices name and corresponding relations. The existing ontology (IoT-ontology, SAREF, OWL-IoT-S, SA, IoT-Lite, SPITFIRE, SSN, oneM2M, IoT-O) is compared based on the concept's actuator, action, service, sensor, observation, energy, lifecycle, device, and software agent [25].

The building climate requires maintaining a similar temperature. In this situation, temperature sensors are produced by European merchants, USA sellers, and FIWARE; these sensors measure the temperature Celsius, Fahrenheit, and Celsius, respectively. The flat (across the gadget) and vertical correspondence (stage) set up paying little mind to the model and maker. The temperature sensor sends the information in (℃) and utilizes JavaScript Object Notation (JSON) arrangement to communicate the data. The comparing actuator requires information in (°F) and XML format. This paper settles the semantic clashes in the temperature. The interoperability isn't accomplished totally for other kinds of sensors and actuators. The context information not used and the common IoT vocabularies are is not distinguished for future IoT applications correspondence [26]. The various device property introduced for device compatibility. The interoperability temperature parameter only considered and other parameter conflicts are not addressed [27]. OWL Ontology [28], characterizes the service profile, service grounding, and service model in the philosophy. The service profile contains the item property, information property, and subclass. Liang L et al. briefly discussed the whole ML algorithm, then summarized the dynamic learning in networks and made some decisions [29].

M3 [30] explains different sensors like ozone sensor, NO2 sensor, sun-powered radiation sensor, and electric charge sensor. Additionally, it depicts numerous gadgets like home Machine-to-Machine (M2M), feeling M2M, energy M2M, climate M2M and automated M2M. The M3 system is executed and fulfills the attributes like reusing space information, cross-domain knowledge, thinking, compelled gadgets, and security. The information extraction measure is not a programmed interaction and the ML calculations are not utilized. The machine doesn't deal with the LOV indexes. Noy NF et al. [31] give the guidelines to make ontologies and demands the reuse of domain ontology. The semantic web index sindice, swoogle search engine is not accessible at this point. Linked Open Vocabularies (LOV) contain more than 400 ontologies. Climate space (wind, temperature, overcast cover, pressure) transportation area (driver feeling, street, climate, commotion), and horticultural space (contamination, environment, feeling) continually reclassified the particular area idea reused in some other area [32]. Noura M et al. implemented the programmed information extraction for explicit domains utilizing ML algorithms. They made a corpus for smart homes, smart cities, and smart weather. It contributes to the semantic interoperability for the WoT [33]. The proposed work is to make the NLP corpus for transportation and give the direction to reuse retrieved information from different domains into the future ontology construct environment.

3. Proposed System

The domain experts select the domain ontology for the specific domain. The domain ontology classes, subclasses, and properties are extracted from ontology then converted into terms. The terms frequency is calculated and the terms are passed into the word2vec algorithm. Corpus is created based on the weights of the terms for input transport-related research articles. The corpus contains the terms and corresponding vector values. After that, the terms are clustered using the K-means algorithms. The domain experts assign the name for the cluster based on the semantic meaning of the clustered terms and the frequency of the terms. This clustered knowledge is utilized as contextual knowledge in future ontology design methodologies. Under the context class, these cluster names are added as subclass or property names. These steps are available in Knowledge Extraction for Interoperability Methodology. The figure 2 shows the step-by-step knowledge extraction process.

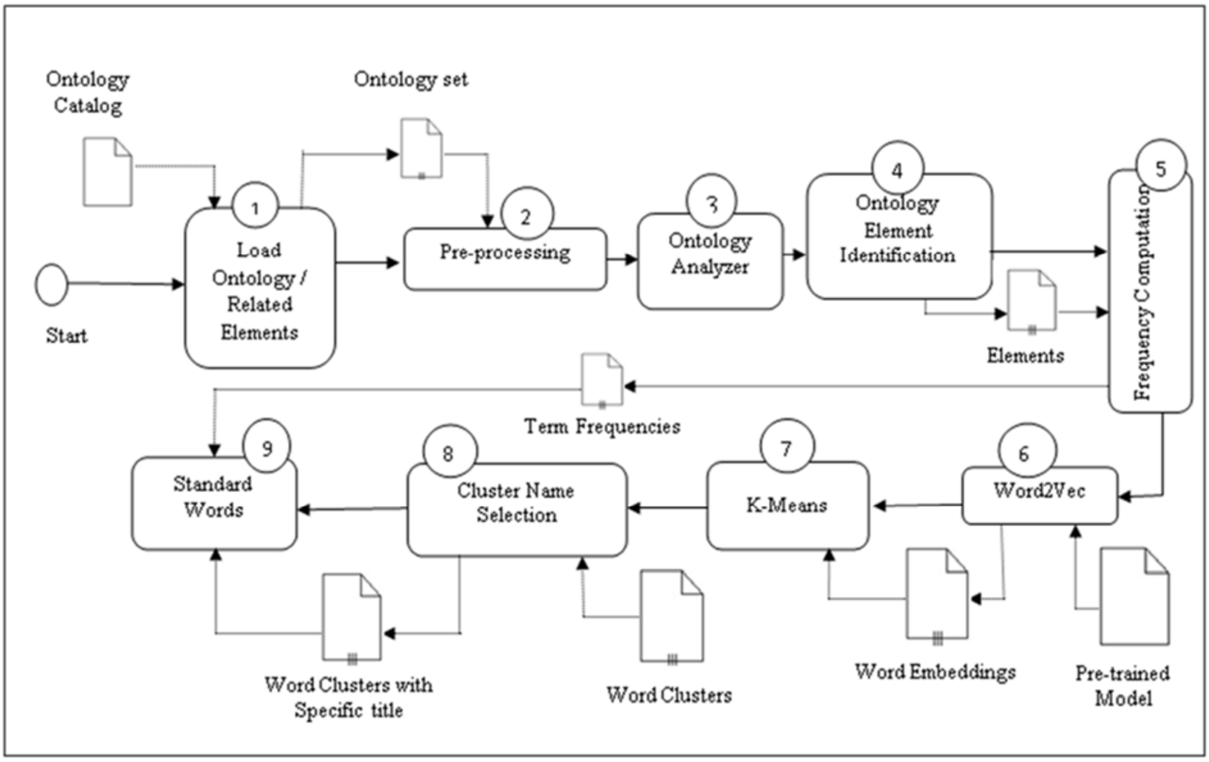

Fig. 2. Knowledge Extraction for Context Methodology.

3.1. *3.1.Knowledge Extraction for Interoperability Methodology*

Step 1 (Loading of ontological Elements):
The relevant RDF, RDFS, and OWL ontology loaded for preprocessing.
Step 2 (Ontology Pre-processing):
i) Ontological characters are encoded into the UNICODE format. ii) Loaded ontology is stored as a "Turtle" (TTL) or RDF/XML or OWL file format.
Step 3 (Ontology Analyzer):
It scans ontology files and removes irrelevant characters then converts all text into lower cases, except the camel case and pascal case.
Algorithm: Text Analysis Algorithm
Input: Pre-processed Ontology file
Output: Ontology Elements
Tool Used: Protégé
Step 3.1 (Subject and Object Retrieval): The ontology elements are classified into concepts, data property, and object property using protégé and SPARQL.
Step 3.2 (String Conversion): Ontology elements are converted into strings.
Step 3.3 (Digit Elimination): Remove the digits present in the strings.
Step 3.4 (Lower Conversion): Except camel case, pascal case template of multiple words, the remaining strings are transformed into lower case.
Step 3.5 (Remove the punctuations): Unwanted punctuations characters are listed in the python code
Step 3.6 (Whitespace Deletion): Remove the tabs or extra spaces that persist in the input text.
Step 3.7 (Model Creation): Run the model spacy.load('en_core_web_lg') for ontology element tokenization.
Step 4 (Identification of Ontology Elements):
Ontology classes and properties are classified. The pascal case and camel cases are identified.
Step 5 (Ontology Frequency computation):
Number of occurrences of concept and property is calculated based on ontology elements and stored in the dictionary then copied into Excel sheet .CSV file
Step 6 (Word2Vec Algorithm):
Dominant ontology elements are passed into the word2vec Skip-Gram model. The similar ontological terms are explored using the pre-trained model. The pre-trained model creation is explained in section 5 Transport corpus creation.
The ontology elements are

(i) $\mathcal{O} = \{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$ Where 'n' is the total number of ontology elements.
(ii) Similar words based on ontology elements are $T = \{T_{11}, \ldots T_{1m}, T_{i\prime j\prime}, \ldots T_{i\prime m}, T_{n1}, \ldots, T_{nm}$ Where 'm' represents the total number of similarity words for specific ontology elements. Let generalize the terms mentioned above into $T_W = \{T_{W_1}, T_{W_2}, \ldots, T_{W_m}\}$ Where m represents the total number of terms.
(iii) Vector points are Vector points are $V_{W_m} = \{V_{W_1}, V_{W_2}, \ldots, V_{W_m}\}$

Step 7 (K- Means Algorithm):
The pseudocode for ML K-means algorithm is described below.

```
Input:
    $V_{W_m} = \{V_{W_1}, V_{W_2}, \ldots, V_{W_m}\}$
k Number of clusters
    $Z = \{C_1, C_2, \ldots, C_{k\prime}\}$
where k' is the total number of clusters.
Output:
A set of k clusters
1.Initialize cluster centroids $\delta_1, \delta_2, \ldots, \delta_{k'}$
2.Repeat until convergence: {
For every λ, set
    $C^I := arg \min_J \left\| {V_W}^{(\lambda)} - \delta_J \right\|^2$
For every J, set
    $\delta_J := \frac{\sum_{\lambda=1}^{m} 1\left\{{C'}^{(\lambda)} = J\right\} {V_W}^{(I)}}{\sum_{\lambda=1}^{m} 1\{C'^{(\lambda)} = J\}}$        (1)
}
```

Step 8 (Cluster Name Selection):
Transportation domain expert assigns the specific name to the cluster

$$C_\lambda \in Cluster\_Name$$

Where λ=1…k'.
Step 9 (Lemmatization and Standardization of Domain Words):
All terms are converted into corresponding root words. Each cluster contains the more relevant words, these terms are valid for the reusability of future IoT applications in a specific domain. High-Frequency words occupy the initial place within the particular cluster rows.

4. Dataset

The following queries are applied to find the research article (Transport Domain Terms) AND "WoT" OR "IoT" OR "Internet of Things" OR "Web of Things". The last six-year research articles were downloaded from the internet. The ScienceDirect, IEEE Journal, Elsevier, and ACM-related research papers and most cited articles are downloaded. The research article images, references, journal templates do not have the domain information. Therefore, mentioned objects are removed from the research article and combined all documents into a single word document. The downloaded 1000 research papers are combined into a single document. Finally, combined word document input contains 38 lakhs of words. Some of the ontologies are found on the LOV catalog and some other ontology is found in some other sources. Most of the ontology files are not available. The ontology concepts are selected based on available image corresponding ontologies and their research articles. Finally, 90 ontology files were found relevant to the transportation domain.

5. Transport Corpus Creation

The research articles are loaded for preprocessing and the author citations are removed. The words are converted into the lower case; numbers are removed from the input file. Then punctuations and white space are eliminated. Text file words are transformed into tokens using the python package Spacy. Stop words are deleted based on predefined stop words in the spacy package. The journal and conference common templates (Abstract, Literature Review, Proposed Work, Conclusion, and Reference) are deleted. The frequency of words present in a text file is computed and words are converted into a one-hot encoded vector. These detailed procedures are defined in the Vectorization algorithm.

Step 1 (Vectorization):
Algorithm: Vectorization using word2vec Skip-Gram Model
Input: Text input with frequency (.csv).
Vector Notation:
Input Layer $a_E = \{a_1, \ldots, a_{i,} \ldots, a_E\}$
Hidden Layer $h_I = \{h_1, \ldots, h_{a,} \ldots, h_I\}$
Output Layer $y_{D \times E} = \{\Delta_1, \Delta_2, \Delta_3\}$

Where,

$$\Delta_1 = \{b_{1,1}, \dots, b_{1,s,} \dots, b_{1,V}\}$$
$$\Delta_2 = \{b_{d,1}, \dots, b_{d,s,} \dots, b_{d,E}\}$$
$$\Delta_3 = \{b_{D,1}, \dots, b_{D,s}, \dots, b_{D,E}\}$$

The input layer has E-dimensions, the output layer has D×E-dimensions and the hidden Layer has I-dimensions.
Package: Pandas
Step 1.1 (Initialization): Generate a data frame of vocabularies with frequencies.
Step 1.2 (Retrieval of Preprocessed Tokens): Read the vocabulary frequency file and build the list of the distinctive vocabularies. The number of distinct vocabularies that say ‘E’
Step 1.3 (Context Window): Set the context window size W=ρ, where ρ >1
Step 1.4 (Information Pair Dictionary): Produce the pair of vocabularies for Window Size W.
Step 1.5 (Feature Selection): The count of data points defines the size of the hidden matrix column value ‘H’.
Step 1.6 (One-Hot Encoded Vectors): The Data Pair Dictionary is transformed into data points. These encoded vectors are represented as a Matrix M_{Ex1}.
Step 2 (Creation of Artificial neurons in NLP):
Deep learning techniques create the neurons. The skip-gram model contains three layers, namely input Layer a_E hidden Layer h_I and output layer $y_{D\times E}$
Algorithm: Deep Learning
Input Layer: For the vocabularyM_{Wi}
Vector Matrix $V_{Wi'}^T$
Hidden Layer: Vector Matrix $V_{Wi'}^T$
Output: M'_{Wi}
Package: Keras
The subscript t, r, s denotes an input layer, an output layer, and a hidden layer, respectively.
Step 2.1 (Embedding Limit): Define the size of the embedding. Let consider Window=constant value
Step 2.2 (The Neural Network):
Step 2.2.1 (Define the Hidden Layer):

$$h = M_{(t,.)}^T \coloneqq V_{Wl'}^T \tag{2}$$

Where M relates to input vocabulary Wl
Step 2.2.2 (Multinomial Distribution)
The Q multinomial distribution calculated by

$$p = (W_{d,s} = W_{o,d} \,|\, W_l) = y_{d,s} = \frac{exp(n'_{d,s})}{\sum_{b'=1}^{V} exp(n_{b'})} \tag{3}$$

Where
$W_{d,b}$ is the s^th vocabulary on d^th plane of the output layer; Where $W_{o,q}$ refers the qth vocabulary in the o^th plane of the output context vocabularies; Where $y_{q,s}$ refers the output of the s^th term on q^th plane of the output layer; Where $n'_{q,s}$ refers the net input of the s^th unit on q^th plane of the output layer;
The output plane shares the same weights

$$n'_{q,s} = n'_s = V'_{Ws}{}^T.h, for\ q = 1,2, \dots, Q \tag{4}$$

Where V'_{Ws} is the output vector of the s-th term in the vocabularies.
Step 2.2.3 (Loss Function):

$$L = -\,log\ log\ p(W_{o,1,}W_{o,2,} \dots, W_{o,Q} \,|\, W_l) \tag{5}$$
$$L = -log\ \prod_{q-1}^{Q} \frac{exp(n'_{q,s_q^*})}{\sum_{s'=1}^{V} exp(n_{s'})} \tag{6}$$
$$L = -\sum_{q=1}^{Q} n'_{s_q^*} + Q.log\ log\ \sum_{s'=1}^{V} exp(n'_{s'}) \tag{7}$$

Where
b_d^* refers the index of the actual qth output context term in the vocabulary.
Apply the partial derivation

$$\frac{\partial L}{\partial n'_{q,s}} = y_{q,s} - t'_{q,s} \coloneqq e_{q,s} \tag{8}$$

Step 2.2.4 (Prediction errors for V-dimensional vector):

$$Ll_s = \sum_{q=1}^{Q} e_{q,s} \tag{9}$$

Define an E-dimensional vector$Ll = \{Ll_1, \cdots, Ll_v\}$

$$\frac{\partial L}{\partial W'_{rs}} = \sum_{q=1}^{Q} \frac{\partial L}{\partial n_{d,s}} \cdot \frac{\partial n_{d,s}}{\partial W'_{rs}} = Ll_s \cdot h_a \tag{10}$$

Step 2.2.5 (Update the above equation for hidden to output matrix M'):

$$M'_{rs}{}^{(new)} = M'_{rs}{}^{(old)} - \beta \cdot Li_s \cdot h_r \tag{11}$$

Step 2.2.5 (Derive the equation based on input to the hidden matrix):

$$LI'_r = \sum_{s=1}^{v} Ll_s \cdot w'_{rs} \qquad (12)$$

Where LI' is a Y-dimensional vector
Step 2.3 (Input Layer Activation):
Link activation of the hidden Layer is linear.
The input layer and output layer weight are $E \times I$ matrix M.
Where,

$$M - input \times hidden\ weight\ matrix$$
$$V_W - \text{represent a row of weight matrix (M)}$$

Every row of matrix M belongs to the I-dimension vector representation V_W. It is related to the input layer. The row of matrix M is represented as V_W^T.

$$h = M_x^T = M_{(T,\cdot)}^T := V_{Wl'}^T$$

The above equation copies the tth row of M to h. $V_{Wi'}^T$ refers the vector representation of the input term W_i.

$$V_l^{\ (new)} = V_{W_l}^{(old)} - \beta \cdot LH^T \qquad (13)$$

Step 2.4 (Output Layer Activation):
The input layer and the output layer weights are
$D \times E$ matrix M'
Where,

$$M' - hidden \times output\ \ weight\ matrix$$
$$V'_W - represent\ row\ of\ weight\ matrix\ (M')$$

The output layer is activated using hierarchical SoftMax. Weight matrix
$M_{E\times I}$ is generated by the skip-gram model based on input terms such as semantic similarity measures.

$$V'_{W_s}{}^{(new)} = V'_{Ws}{}^{(old)} - \beta \cdot Ll_s \cdot h \qquad (14)$$
$$for\ s = 1,2,\cdots, E \cdot$$

The previous stage weight matrix
$M_{E\times I}$ contains the original values of the skip-gram model. To find the loss function LI_b using Hierarchical Softmax Algorithms. Then update loss function into an output vector V'_t.

$$V'^{(new)}_s = V'^{(old)}_s - \alpha(\sigma(V'_s{}^T h) - t'_s) \cdot h \qquad (15)$$

Where α is the learning rate.

6. Experiment Analysis

Identified Name	Ontology Reference
Zhao Toyota1 Car	[34]
Fuchs 2008 Transport	[35]
OpenStreetMap Route Planning Codescu 2014 Tags Ontology	[36]
Davis 2011 Transit	[37]
Santipantakis 2017 datAcron Ontology	[38]
Traffic Danger Waliszko 2011	[39]
SEAS Transport Electric Vehicle 2017	[40]
Martin HeppVehicle Transport	[41]

Table 1. Transport Related Available ontology.

The ontology files are stored in a temporary repository. Table 1 illustrates available ontology files in transport-related IoT applications. Unfortunately, only limited ontology files are accessible. Many of the ontology files are not available and only related research papers are available. The open-source ontology editor is a powerful tool to visualize and query the ontology concepts and its properties. But various ontology files are not supported by the protégé tool because of inference problems. The online ontology visualization tool WebVOWL supports most of the ontology files. Few authors designed the editor to visualize the ontology axioms.

The SPARQL Query language is used to retrieve the ontology concepts and properties. In the case of the tool WebVOWL, the concepts and properties are filtered. Ontospy is the python package which is used to visualize the ontology and inspect the ontology axioms and it supports all types of ontology files. The main drawback of the semantic web is some of the ontology files neither available nor supported. Hence, considered their relevant research article popular concepts with the help of a domain expert.

Topic	Clustered Terms
Information	data, information, specification, datatype, database, domain, content, set
Device	device, lightDevice, meter, lightSwitch, broadband, camara, break, wheel
Sensor	actuator, sensory, observation, environment, captureoperator, transmit, autonomous, temperature, proximity
Units	kelvin, kilowatt, hour, flux, pascal, dollar, walt, temperature, fahrenheit
Energy	energy unit, generator, humidity, HVAC, power
Fuel	coal, petrol, electricity, wind, sun, power, solar, current, voltage, battery, charge, electric
Vehicle	bus, car, bicycle, crane, twoWheeler, fourWheeler, flight, train, metro, aircraft, ship, boat, transport
Traffic	signal, road, pedestrian, crossing, stop, start, alert, fine, safety, engine off, engine start, TrafficObject, TrafficLight
Road	highway, oneway, subway, flyover, bridge, lane, expressway, track, railway, roadConditions, ParticipantUseLane
Semantic web	ontology, axioms, data property, object property, SPAROL, SWRL, hermit, reasoner, pallet, RDF, OWL
User	driver, operator, automate, automatic, membership
System	hardware, switch, computer, router, Modem
Location	search, GPS, source, destination, mobility, latitude, longitude, map, route, roam, geographic, radar
Airport	boarding, landing, takeoff, pilot, airhost, runway, emergency, firealert, datum, passenger, parking
Technology	software, network, IoT, smart, tensor, cloud, WoT, RFID, system, base
State	running, moving, waiting, suspend, terminate, hold, kill, resume
Process	reservation, consultation, block, travel, insurance, hire, check-in, immigration
Time	clock, speed, rpm, hour, second, startTime, endTime, RPM
Fire safety	alarm, communicator, message, waterlevel, heat
Airconditioning	cool, standby, thermostat, humidity, temperature

Table 2. IoT related Transport domain word cluster

The ontology elements are converted into lowercase except for the pascal case and camel case. The unwanted characters are removed like hyphen, underscore, etc. The term frequency is computed for retrieved ontology elements or text file concepts. The terms data, information, device, domain, sensor, speed, service, signal, context, and time are identified as high- frequency terms. The terms crossing, recognize, start, quality, SAREF, style, framework, hour, intelligence, and direction are identified as low-frequency terms. The high-frequency terms are passed into the IoT-related transportation corpus. The corpus is created by word2vector skip-gram model with input of 1000 IoT-related transportation research articles. The word2vec model generates terms that are similar to input ontology elements. The generated terms are clustered concerning their vector distance value using an unsupervised ML algorithm. For K=48, the transport terms are clustered with the k-means algorithm. of lemmatization, the term “cars” and “trucks” are converted into root word “car” and “truck”, respectively.

7. Result Analysis

The ontology elements are passed into the transport domain corpus using the word2vector skip-gram model. Fig. 3. shows the similar terms car, truck, cars, passenger, trucks, driving, bus, driver and parked for the input concept vehicle. These words are more similar to input term “vehicle”. The output term “car” has a vector value 0.8834 and the term “cars” has a vector value 0.8089. The output term truck has a vector value 0.8651 and the term “trucks” has a vector value 0.7966. In the process

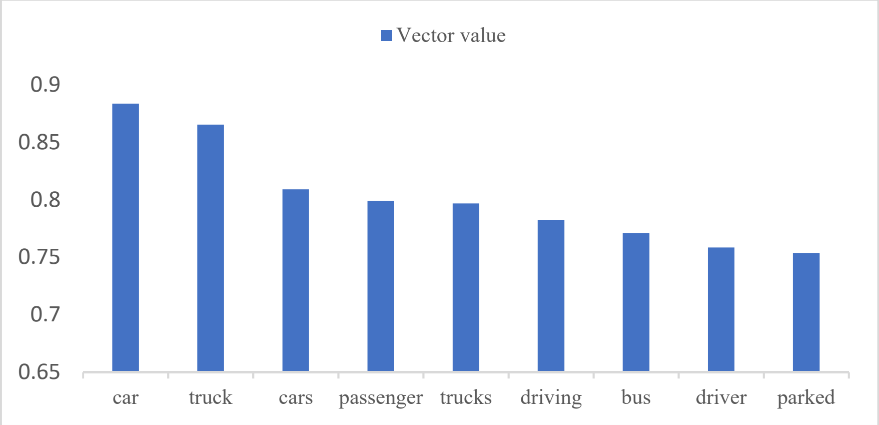

Fig. 3. Similarity terms for the input concept vehicle

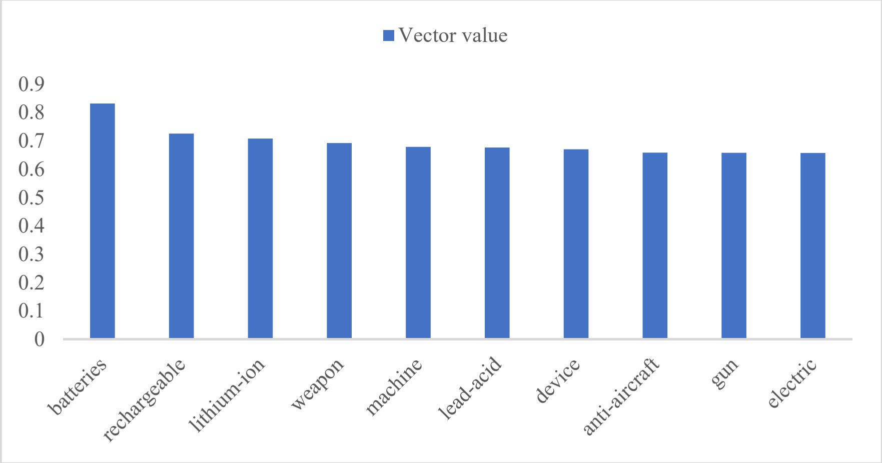

Fig. 4. Similarity terms for input concept battery

Fig. 4. shows the similar terms for batteries, rechargeable, lithium-ion, weapon, machine, lead-acid, device, anti-aircraft, gun, and electric for the input concept vehicle. The output term “batteries” has a vector value 0.8322. The output term “rechargeable” has a vector value 0.7258. The output term “lithium-ion” has a vector value 0.7083. These terms vectors points are more similar to input term “battery”. The output term “anti-aircraft” has a vector value 0.6586. The output term “gun” has a vector value 0.6580. The output term “electric” has a vector value 0.6577. According to the vector value, mentioned terms are less similar comparatively the term batteries, rechargeable and lithium-ion.

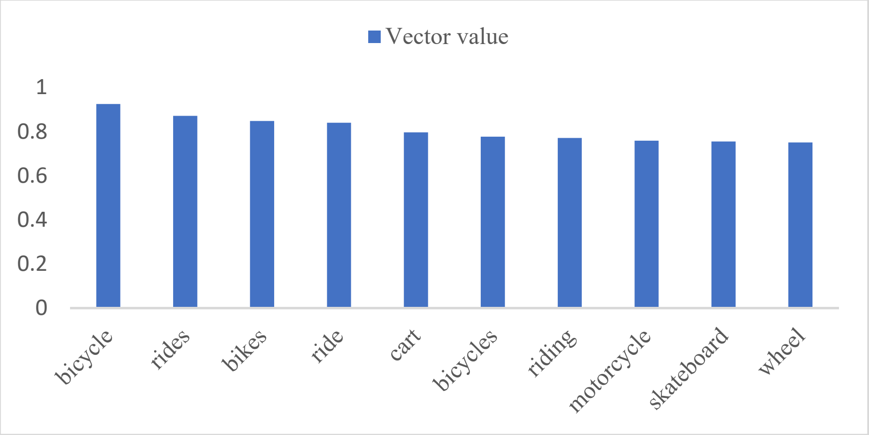

Fig. 5. Similarity terms for an input concept bike

Fig. 5. illustrates the similar terms for bicycle, rides, bikes, ride, cart, bicycles, riding, motorcycle, skateboard, and wheel for input concept bike. The output term “car” has a bicycle value 0.9246 and the term “bicycles” has a vector value 0.7761. The output term “ride” has a vector value 0.8397, output term “rides” has a vector value 0.8710 and output term “riding” has a vector value 0.7702. In the process of lemmatization, the terms bicycles and (rides, riding) are converted into root word namely bicycle and ride. Table 2 portrays the transport domain clustered terms with the topic name. The similarity terms are clustered based on vector distance using the K-Means algorithm, For K=48, we found most similar words within clusters. After that, the domain expert assigns the appropriate name to the clusters. The cluster Traffic has semantically meaningful terms like signal, road, pedestrian, crossing, stop, start, alert, fine, safety, engine off, engine start, TrafficObject, and TrafficLight. The cluster Fire safety has semantically coherent words such as alarm, communicator, message, waterlevel, and heat. The cluster Location has semantically meaningful words search, GPS, source, destination, mobility, latitude, longitude, map, route, roam, geographic, and radar.

8. Conclusion and Future work

The transport-related IoT research papers, downloaded from reputed journals and conferences based on these text file inputs, the corpus is constructed using word2vec skip-gram model. The ontology files are downloaded from the ontology catalog and the internet. Many of the ontology files are not available on the internet; in those cases, domain experts found frequently used concepts from a corresponding research paper. From the ontology file ontology concepts are extracted using ontospy, owlViz, and protégé tool using SPARQL Query. Finally, 6053 ontology concepts are identified and passed as input into the transport corpus. For each input concept related most similar terms are generated by the skip-gram model. For 6053 ontology concepts corresponding to 16300 similar terms, related vector values are generated. These terms are clustered corresponding vector values using an unsupervised ML K-means algorithm. The domain experts assign the unique name to all clusters and order the terms within clusters based on their frequency values.

In future work, the clustered concepts can be annotated in OWL-based ontology. The ontology concepts can be identified from related domain ontology or research articles using ranking algorithms. We will identify more clusters in the healthcare domain, food-related IoT applications, and security domains then annotate these concepts and properties in OWL ontology.

References

[1] Atzori L, Iera A, Morabito G. The Internet of Things: A survey. Computer Network. 2010; 54(15), 2787–2805.

[2] Guinard D, Trifa V. Building the Web of Things: With Examples in Node. JS and Raspberry Pi. Shelter Island, NY, USA: Manning, 2016.

[3] Noura M, Atiquzzaman M, Gaedke M. Interoperability in Internet of Things infrastructure: Classification, challenges, and future work. In Proc. Int. Conf. Internet Things Service 2017 (pp. 11–18).

[4] Ma L, Zhang Y. Using Word2Vec to process big text data. IEEE International Conference on Big Data 2015. (pp. 2895-2897).

[5] Strang, Linnhoff-Popien C, Frank K CoOL: A Context Ontology Language to Enable Contextual Interoperability. DAIS 2003. LNCS 2893, (pp. 236–247).

[6] Negash B, Westerlund T, Rahmani AM, Liljeberg P, Tenhunen H. DoS-IL: A Domain Specific Internet of Things Language for Resource Constrained Devices. Procedia Computer Science 109C 2017. (pp. 416–423). ScienceDirect.

[7] Osman I, Yahia SB, Diallo G. Ontology Integration: Approaches and Challenging Issues. Information Fusion 2021. ScienceDirect.

[8] Novo O, Francesco MD. Semantic Interoperability in the IoT: Extending the web of Things Architecture. Internet Things 2021. ACM Trans.

[9] Naudet Y, Latour T, Guedria W, Chen Y. Towards a systemic formalization of interoperability. Computers in Industry 61 2010. (pp. 176–185). ScienceDirect.

[10] Ganzha M, Paprzycki M, Pawlowski W, Szmeja P, Wasielewska K. Semantic interoperability in the Internet of Things: an overview from the INTER-IoT perspective. Journal of Network and Computer Applications. 2016. 81:111-24.

[11] Ullah F, Habib MA, Farhan M, Khalid S, Durrani MY, Jabbar S. Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare. Sustainable cities and society. 2017 Oct 1;34:90-6.

[12] Nawaratne R, Alahakoon D, De Silva D, Chhetri P, Chilamkurti N. Self-evolving intelligent algorithms for facilitating data interoperability in IoT environments. Future Generation Computer Systems. 2018 Sep 1;86:421-32.

[13] Andročec D, Novak M, Oreški D. Using semantic web for internet of things interoperability: A systematic review. International Journal on Semantic Web and Information Systems (IJSWIS). 2018 Oct 1;14(4):147-71.

[14] Leal GD, Guédria W, Panetto H. An ontology for interoperability assessment: A systemic approach. Journal of Industrial Information Integration. 2019 Dec 1;16:100100.

[15] Ahmad A, Cuomo S, Wu W, Jeon G. Intelligent algorithms and standards for interoperability in Internet of Things.

[16] Panetto H, Dassisti M, Tursi A. ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. Advanced Engineering Informatics. 2012 Apr 1;26(2):334-48.

[17] Cross V, Pal A. An ontology analysis tool. International Journal of General Systems. 2008 Feb 1;37(1):17-44.

[18] https://en.wikipedia.org/wiki/Semantic_Web. Accessed 29 July October 2021.

[19] Fortino G, Savaglio C, Palau CE, de Puga JS, Ganzha M, Paprzycki M, Montesinos M, Liotta A, Llop M. Towards multi-layer interoperability of heterogeneous IoT platforms: The INTER-IoT approach. InIntegration, interconnection, and interoperability of IoT systems 2018 (pp. 199-232). Springer, Cham.

[20] Kotis K, Katasonov A. An iot-ontology for the representation of interconnected, clustered and aligned smart entities. Technical report, VTT Technical Research Center, Finland VTT Technical Research Center, Finland. 2012.

[21] Elsaleh T, Enshaeifar S, Rezvani R, Acton ST, Janeiko V, Bermudez-Edo M. IoT-Stream: A lightweight ontology for internet of things data streams and its use with data analytics and event detection services. Sensors. 2020 Jan;20(4):953.

[22] Janowicz K, Haller A, Cox SJ, Le Phuoc D, Lefrançois M. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. Journal of Web Semantics. 2019 May 1;56:1-0.

[23] Ganzha M, Paprzycki M, Pawłowski W, Szmeja P, Wasielewska K. Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective. Journal of Network and Computer Applications. 2017 Mar 1;81:111-24.

[24] Gyrard A, Bonnet C, Boudaoud K, Serrano M. LOV4IoT: A second life for ontology-based domain knowledge to build Semantic Web of Things applications. In2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud) 2016 Aug 22 (pp. 254-261). IEEE.

[25] Seydoux N, Drira K, Hernandez N, Monteil T. IoT-O, a core-domain IoT ontology to represent connected devices networks. InEuropean Knowledge Acquisition Workshop 2016 Nov 19 (pp. 561-576). Springer, Cham.

[26] Lakka E, Petroulakis NE, Hatzivasilis G, Soultatos O, Michalodimitrakis M, Rak U, Waledzik K, Anicic D, Kulkarni V. End-to-end semantic interoperability mechanisms for iot. In2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) 2019 Sep 11 (pp. 1-6). IEEE.

[27] Suman S, Perumal T, Ahmadon MA, Mustapha N, Yaakob R, Yamaguchi S. IoT Device Management using Semantics for Distinguishing Device Compatibility. IEEE 1st Global Conference on Life Sciences and Technologies. 2019. (pp. 185-188).

[28] "OWL-S ontology". Internet: https://www.w3.org/Submission/OWL-S/. Accessed 29 July October 2021.

[29] Liang L, Ye H, Li GY. Toward intelligent vehicular networks: A machine learning framework. IEEE Internet of Things Journal. 2018 Sep 26;6(1):124-35.

[30] "M3". Internet: http://sensormeasurement.appspot.com/m3.owl. Accessed 29 July October 2021.

[31] Noy NF, McGuinness DL. Ontology development 101: A guide to creating your first ontology.

[32] Gyrard A, Bonnet C, Boudaoud K, Serrano M. LOV4IoT: A second life for ontology-based domain knowledge to build Semantic Web of Things applications. In2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud) 2016 Aug 22 (pp. 254-261). IEEE.

[33] Noura M, Gyrard A, Heil S, Gaedke M. Automatic Knowledge Extraction to Build Semantic Web of Things Applications. IEEE Internet Things J.. 2019 Oct 1;6(5):8447-54.
[34] https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/ Ontology/TTICore-0.03/TTICarOnto.owl. Accessed 29 July October 2021.
[35] http://vi.uni-klu.ac.at/ontology/ DrivingContext.owl. Accessed 29 July October 2021.
[36] https://raw.githubusercontent.com/doroam/planning-do-roam/master/Ontology/tags.owl. Accessed 29 July October 2021.
[37] https://raw.githubusercontent.com/iand/vocab-transit/master/vocab.rdf. Accessed 29 July October 2021.
[38] http://ai-group.ds.unipi.gr/datacron_ontology/ TopDatAcronOnto_SSN_FM.owl. Accessed 29 July October 2021.
[39] https://raw.githubusercontent.com/jwaliszko/msc-thesis/master/setup/ontology/TrafficDanger.owl. Accessed 29 July October 2021.
[40] https://w3id.org/seas/ElectricVehicleOntology-1.0.rdf. Accessed 29 July October 2021.
[41] http://www.heppnetz.de/ontologies/vso/ns.owl. Accessed 29 July October 2021.

Authors Profile

Mr. S. Prakash is working as an Assistant Professor in the School of Computing Science and Engineering, Galgotias University, Delhi - NCR, India. He is pursuing Ph. D in Computer Science and Engineering, Galgotias University, Delhi - NCR, India. He has completed M.E in Computer Science and Engineering from Anna University, Tamil Nadu, India. His research interests include the Internet of Things, Natural Language Processing, Semantic Web, Cryptography and Network Security.

Dr. T. Poongodi is working as an Associate Professor in the School of Computing Science and Engineering, Galgotias University, Delhi - NCR, India. She has completed Ph. D in Information Technology (Information and Communication Engineering) from Anna University, Tamil Nadu, India. Her area of research interests includes Big data, Wireless ad-hoc network, Internet of Things, Network Security and Blockchain Technology. She has published more than 50 papers in various international journals, National/International Conferences, and book chapters in Springer, Elsevier, Wiley, De-Gruyter, CRC Press, IGI global, and edited books in CRC, IET, Wiley, Springer, Apple Academic Press.