Indonesian Journal of Electrical Engineering and Computer Science
Vol. 27, No. 1, July 2022, pp. 504~512
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v27.i1.pp504-512

Augmentation of contextual knowledge based on domain dominant words for IoT applications interoperability

Prakash Shanmurthy[1], Poongodi Thangamuthu[1], Balamurugan Balusamy[1], Seifedine Kadry[2]
[1]School of Computing Science and Engineering, Galgotias University, Greater Noida, India
[2]Department of Applied Data Science, Noroff University College, Oslo, Norway

Article Info

Article history:

Received Apr 6, 2022
Revised May 5, 2022
Accepted May 20, 2022

Keywords:

Internet of things
Natural language processing
Semantic web
Web ontology language

ABSTRACT

Semantic web technology is adapted to the internet of things (IoT) for web-based applications to globally connect the services. Web ontology language (OWL) domain ontology is a powerful machine-readable language for domain knowledge representation. The developer stored the IoT application relevant ontology in a repository or catalogue. Hence, IoT application-related ontology files are available for reuse, but many of the IoT application-relevant ontology files are publicly not available or inaccessible. The proposed idea is to extract the contextual knowledge of IoT applications that contain inaccessible ontology files. The context-wise specific domain IoT applications are not obtainable, hence respective ontology-based research papers are identified and their frequent terms are computed. The selected contextual dominant frequent terms from the transport domain are passed into the skip-gram flavour of word2vector modelled natural language processing (NLP) corpus which produces most similar terms. The domain experts select the appropriate terms to annotate in OWL ontology for contextual knowledge augmentation. Finally, 1422 contextual terms were generated based on dominant terms of selected IoT applications.

Corresponding Author:

Prakash Shanmurthy
School of Computing Science and Engineering, Galgotias University
Greater Noida, Uttar Pradesh, India
Email: research4prakash@gmail.com

1. INTRODUCTION

Semantic web ontology's role is to annotate the atomic web concepts and relations in machine-readable form which generates the inference and provides interoperability for multiple domains. Researchers utilize the power of web ontology language (OWL) ontology for Internet of Things (IoT) related applications to virtually represent the sensor names, relationships between sensors, sensor-generated values, and relevant protocols in an unstructured way. Natural language processing (NLP) has two major parts, namely natural language understanding (NLU) and natural language generation (NLG). The NLU involves mapping the given input into required representation and analysing various aspects of language. The NLU is much harder than NLG and in specific NLP corpus efficiency is completely dependent on the input text file size. The word2vec is a popular technique to generate word embedding which contains two architectures namely continuous bag of words (CBOW) and skip-gram. The CBOW model generates focus words based on context words which require a small corpus with fast training for frequent words. The skip-gram model generates context words based on focus words to explore relevant contextual words. The CBOW model dataset training is made by the negative sampling method, and skip-gram model training is made by the Hierarchical Softmax method [1].

This article concentrates on designing a knowledge augmentation methodology to extract the knowledge of unavailable ontology-related IoT applications. This methodology will support the reusability of existing IoT application knowledge in further use. Section 2 deals with the related work of different ontology construction methodologies. Section 3 proposes the knowledge augmentation methodology. Section 4 explains the experimental analysis of transport domain skip-gram model corpus along with dominant words. Section 5 describes the result and discussion part. Finally, section 6 presents the conclusion and future work.

2. LITERATURE REVIEW

All text contents are converted into concepts and relations using hybrid supervision and neural network. In this work, a standard pre-trained neural network is not utilized [2]. The various levels of an assertion like substance-level, structure-level, intention-level, and situation-level are taken into consideration for ontology construction [3]. Building topology ontology (BOT)was created and linked with world wide web consortium (W3C) that describes catalogues, IoT devices, and sensors also demonstrated for web-based applications [4]. Fuzzy ontology was designed based on fuzzy rules and fuzzy metrics for the medical domain [5]. Development of Industry ontology from unstructured text fully depends on domain expert's support. Automation and Intelligence techniques are not utilized in ontology design methodology [6]. Ontology alignment is applied for entities of different ontologies. Once the domain-related standard words are framed, constructing a new ontology concept with some similar concept names is quite easier. Dismantle the ontology into terms for finding similar words of the existing terms [7].

Various ontology design methodologies are analysed and waste management ontology is constructed with the parameter of reuse, interoperability, and knowledge acquisition, but how the interoperability is achieved is not proved [8]. The learning ontology is constructed using pre-processed electronic textbooks and NLP techniques [9]. Superscale ontology and real-time ontology were framed to reduce the ontology search time. It provides multi-domain correlation and multi-domain intercommunication. Instead of machine learning techniques, the adaptive filter algorithm was introduced to integrate the domain ontology [10]. The multiple ontology semantic reasoning is not possible instead of single ontology semantic reasoning. Using the deep learning technique, new inference rules were found based on many semantic networks which are formed by ontology triples and relations [11]. During the software testing phase, ontology solves the knowledge silo problem which contains the software testing process information and failure details [12]. The cross-domain knowledge is integrated into multi-aspect ontology which supports the decision system [13]. The semantic virtualization technique suggests the removal of the vertical barrier between various IoT application standards which leads to data acquisition plans [14]. The author proposed an ontology learning algorithm that develops the ontology from the property graph, aligns the developed ontology corresponding to domain ontology, and automated mapping performed on relational to RDF model and non-relational to resource description framework (RDF) model. The drawback of this model is the way of addressing terminological heterogeneity issues because the string similarity is computed using two models only [15]. The knowledge extraction methodology is proposed to retrieve the domain knowledge from the freebase RDF dumps, this methodology addressed the challenges while retrieving the triples from freebase triples. The main drawback of this methodology is that the intelligent technique is not utilized for the extraction of objects from triples [16]. The proposed methodology tries to remove semantic, syntactic, and structural heterogeneity of homogeneous domain databases using a data turn and query turn scheme [17].

In the OntoKhoj model, performed functions are ontology crawl, ontology classification, and ontology rank based on the semantic web [18]. Ontology is developed using the method of ontology mapping and merging in which instead of keywords entire ontology is used as an input query [19]. Ontodia is an open-source JavaScript tool that supports to visualization of complex ontology for learning purposes [20]. OntoSearch is a search engine based on Java server pages (JSP) and Jena technologies which include the utility search ontology using keywords and visualizing the ontology elements [21]. Nowadays, a large amount of text is available as unstructured data, semi-structured data, and web pages. The relational extraction model explores the feature multiple relation extraction three label methods (i.e., entity category, relation category, and relation condition) [22]. The required context text content is extracted from the original cover text using the adaptive binary coding method [23]. The author addresses the problem of parallel computation-based classification method and it is difficult to adopt another platform because of the various requirement of users. Platform independent parallel classification method is performed by OWL ontologies with parallel reasoning [24].

The ontological concepts are extracted as terms, then these frequent ontological terms are passed into the NLP corpus that generates similar terms and these terms are clustered using the K-means algorithm. But in several IoT applications, ontology is publicly not available in the ontology catalogue [25].

3. PROPOSED METHOD

Figure 1. describes the knowledge augmentation methodology which produces the similarity words based on dominant words. The dominant terms in the input are obtained from research articles that are related to IoT based ontology files that are inaccessible. The index, token, Dictionary_count, word_count, lower case, upper case, character, and underscore is respectively referred to as ‘I’, ‘TK’, DC, WC, LC, UC, ch and US.

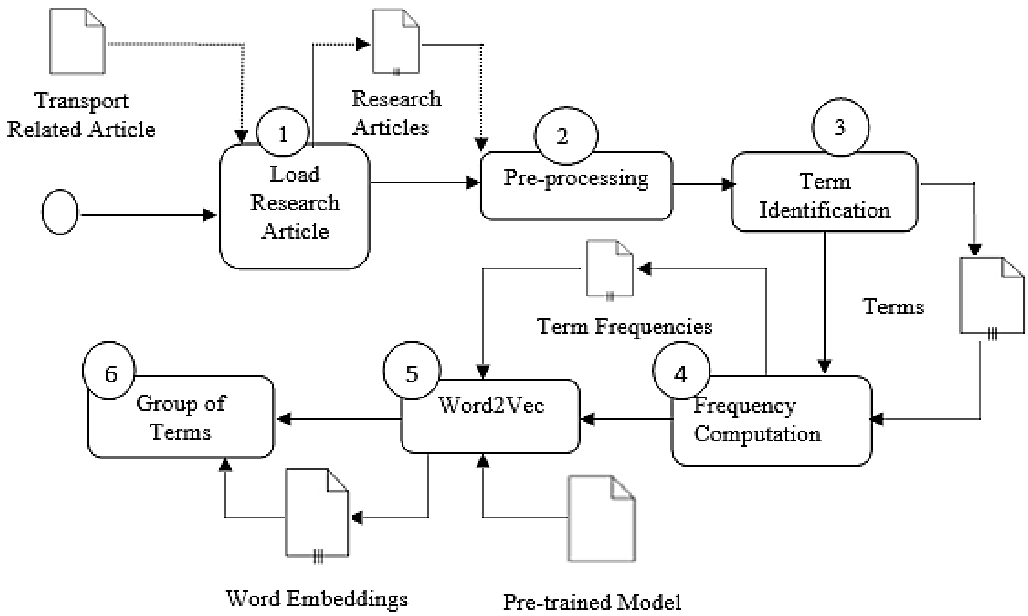

Figure 1. Knowledge Augmentation Methodology

- Step 1 (Loading of text file): The domain-relevant research articles are loaded for pre-processing.
- Step 2 (Pre-processing): The research articles are converted into strings; where the author citations are eliminated. Then the digits punctuations and whitespace are removed from the input file. Figure 2. Portrays the removal of author citations.
- Step 3 (Term identification): Strings are changed into lower case except for the multiword camel case and pascal case. The Figure 3. Illustrates identification of single words and multiword with required template.
- Step 4 (Dominant word computation): The frequencies of unique words are computed among the set of research articles whose ontology is publicly unavailable or inaccessible under specific contexts in the transport domain. The computed high-frequency words are considered dominant words which are described in Figure 4.
- Step 5 (Word-to-vector): The research articles are $\mathrm{D} = \{\mathrm{D}_1, \dots, \mathrm{D}_n\}$, Where 'n' is the total number of the research document. Similar terms generated based on ‘D’ article dominant input terms are $T = \{T_{11}, \dots T_{1m}, T_{i'j'}, \dots T_{i'm}, T_{n1}, \dots, T_{nm}\}$ Where ‘m’ represents the total number of similarity words for specific domain research articles. Let generalize the terms mentioned above into $T_W = \{T_{W_1}, T_{W_2}, \dots, T_{W_m}\}$ Where m represents the total number of terms. The cosine similarity between two words is calculated using (1).

$$\cos(\theta) = \frac{a.b}{|a|.|b|} \text{ a, b} \in T_W \quad (1)$$

- Step 6 (Group of terms transformed as ontology concepts): The previous step 5 generated similar terms that are grouped and represented as ontological concepts say $C_W = \{C_{W_1}, C_{W_2}, \dots, C_{W_m}\}$.

```
Pseudocode: Removal of citation
Input: Text File
Output: Text File without author citation
start:
Convert input string into python tokens.
Initialize I=0
if TK[I]='('then
    if TK[I+1] = digit then
        Skip below statements
    endif
for TK[I]='(' until TK[I+1]=')' then
            if TK[I+1] = string then
            if TK[I+2] = digit then
            if TK[I+3] = '('then
            Remove TKs from I to I+3 in list
            endif
            endif
            endif
endfor
endif
```

Figure 2. Pseudeo code for removal of citation

```
Pseudocode: Case Folding
Input: List with python Token's
Output: List2 tokens are LC
  except for camel case and pascal case
start
Initialize I=0, I2=1
for TK[I] until end_of _list
  if TK [I] = String then
     for TK_Char in TK[1: ]
        if TK_Char[I2] = UC then
              Append TK_Char[I2] in list2
              break the iteration
        endif
        else
           Convert TK_Char[I2] into LC
              Append TK_Char[I2] in list2
              increment I2
     endfor
   elseif TK [I] = '-' then
Remove the hyphen
   elseif TK [I] = '_' then
I2=Individual token US position
              if TK_Char [I2] = '_' then
               Remove US
                After US convert ch into LC
              else
               Remove US
               endif
   elseif TK_Char [I2] = ':' then
     Replace the colon with whitespace
   elseif TK_Char [I2] = LC then
    Append TK[I] in list2
              increment I
endfor
end
```

Figure 3. Pseudeo code for case folding

```
Pseudo code: Dominant Word Count
Input: List with python Token's
Output: Dictionary with word frequency
start
Initialize DC=NULL
for TK[I] until end_of _list
  for TK[I]  in  List
    if TK [I] = LC then
       if TK [I] = LC then
          if TK [I] = LC then
          WC=0
          endif
       endif
  endif
  else
  DC[TK[I]]= WC[TK[I]]+1
            for TK_Char in TK[1: ]
                 if TK_Char[I2] = UC then
                 Append TK_Char[I2] in list2
                 endif
            endfor
  endfor
endfor
end
```

Figure 4. Pseudeo code for dominant word count

4. EXPERIMENTAL ANALYSIS

The NLP corpus is created using the transport domain-related 1000 research articles which are downloaded from reputed journals under the context of aircraft, airport, ambulance, bicycle, bike, bus, car, cargo ship, crane, electric vehicle, electric vehicle charging, flight, gps, parking, road traffic, train, vehicle. The article's citation, an article template, reference, and bibliography are removed from the research papers to attain the efficient NLP corpus. The 15 lakh words are taken for training the skip-gram model neural network. The multi-words are converted into appropriate words by using a case-folding algorithm. According to Figure 3. Case Folding pseudo code, the following rules are applied; i) rule 1 the multiword camelCase multiword 'deviceNode' is not changed, ii) rule 2 the pascal case multiword 'DeviceNode' also not changed, iii) rule 3 the kebab-case multiword 'device-node' is converted into camel case multiword 'deviceNode', iv) Rule 4 the RDF label 'skos:altLabel' converted into single words namely 'skos' and altLabel, and v) rule 5 the upper snake case multiword 'HAS_NEXT' is transformed into pascal case multiword 'HasNext'.

Table 1. describes the transport domain-related context and corresponding dominant words. The airport, ambulance, cargo ship, and road traffic have the same dominant word "system". The high-frequency three words are illustrated in the table format along with high-frequency words. The "Aircraft" has the following high-frequency words: information, datum, aircraft, assembly, network, management, component, power, ot, inventory, application, process, control, technology, battery, and operation. The "Airport" has the following high-frequency words system, sensor, device, node, address, airport, ot, network, information, server, baggage, lora, send, agent, strategy, rfid, service, tag, passenger, base, time, parking, design, location and architecture, reader, and communication. The "Ambulance" has the following high-frequency words: system, patient, datum, ambulance, device, sensor, ot, health, network, information, smart, paramedical, disease, control, and base. These dominant terms were passed into the skip-gram algorithm that generate 1264 words. The context-wise generated word counts are ambulance (40), bicycle (40), bus (48), electric vehicle (48), aircraft (56), bike (56), crane (56), car (64), road traffic (64), airport (72), GPS (72), flight (88), parking (88), rapid transit (88), train (88), transport (88), cargo ship (96), and vehicle (112). The percentatge of reuseable concepts is calculated using (2).

$$\text{Reuseability} = \frac{q}{m} \quad (2)$$

Let 'q' is the number of concpets matched with standarded IoT related ontology and 'm' represents the total number of terms. The matching between ontology concpets made by SPARQL query language.

Table 1. List of dominant word name

Transportation-related context	Dominant Word Names
Aircraft	information, datum, aircraft, assembly
Airport	system, sensor, device, node
Ambulance	system, patient, datum, ambulance.
Bicycle	document, smart, bicycle, city.
Bike	bike, datum, share bicycle.
Bus	bus, time, network, datum.
Car	car, system, use, datum.
Cargo Ship	system, service, use, cloud.
Crane	datum, digital, twin, crane.
Electric Vehicle	ev, system, use, vehicle.
Electric Vehicle Charging	ev, charge, battery, use.
Flight	av, vs, datum, ot.
GPS	technology, system, ot, gps.
Parking	parking, car, system, user.
Road Traffic	system, traffic, transportation, datum.
Train	tensor, datum, ot, flow.
Vehicle	vehicle, system, accident, ot.

5. RESULT AND DISCUSSION

Figure 5. shows the ontological terms inputs and dominant terms input percentage. In the context of aircraft, cargo ship, and flight, the bar chart contains the high percentage of dominant words which shows a greater number of IoT applications ontology there are not accessible. The transport, road traffic, bus, and ambulance context show less percentage of dominant words which indicates the mentioned context IoT application ontologies that are accessible.

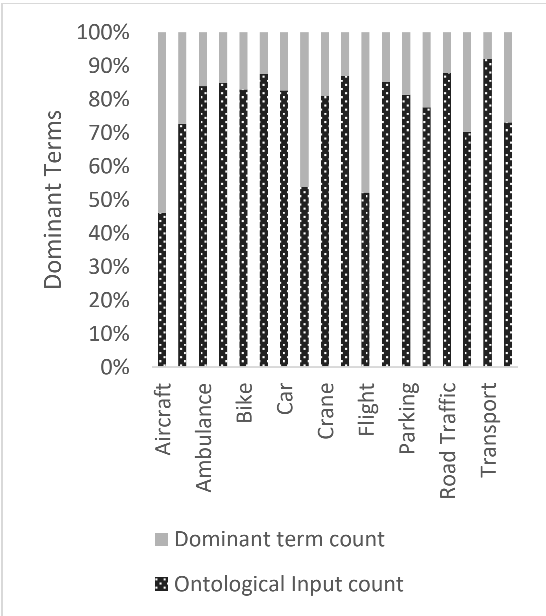

Figure 5. Comparison of ontological terms inputs and dominant words inputs

Figure 6. Portrays the number of available ontologies concepts in the transportation domain. The 16300 contextual terms were generated based on ontological inputs. The obtained contextual ontological concepts based on dominant words is 1422. The context transport, car, and road traffic have more number concepts compared to other contexts. The contexts aircraft, flight, and cargo ships have a smaller number of concepts. Figure 6. Clearly shows no correlation between the number of ontology concepts based on ontological inputs and the number of ontology concepts concerning dominant words inputs. It seemly shows all the transport context contains inaccessible ontology that is in which the semantic web best practice guidelines are not followed.

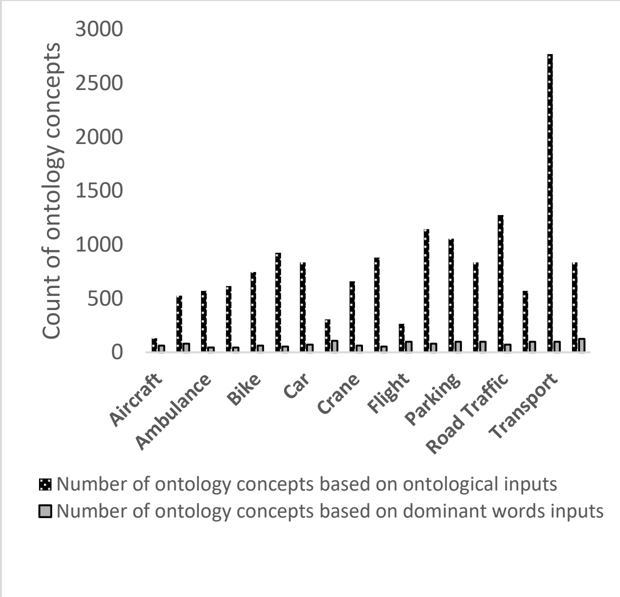

Figure 6. Count of ontology concepts based on ontological inputs and dominant words inputs

Table 2. shows a set of contextual words corresponding to interoperability percentage. The interoperability parameter is defined as specific domain ontology concept reuse in some other domain applications. The context Transport generated ontology concept reuse 40%. The context 'road traffic' generated ontology concept reuse 35%.

Table 2. Contextual knowledge reuse

Context	Percentage of ontology concept reuse
Aircraft	8
Airport	22
Ambulance	19
Bicycle	26
Bike	31
Bus	33
Car	28
Cargo Ship	11
Crane	18
Electric Vehicle	22
Flight	8
GPS	27
Parking	27
Rapid Transit	29
Road Traffic	35
Train	19
Transport	40
Vehicle	26

Figure 7. llustrates knowledge extraction for context methodology [25] generated ontology concepts percentage based on Transportation relation IoT applications ontological concepts. The context Airport, Ambulance, Bicycle, Crane, and Train reserves the 4% of ontological concepts. The context word Bus, Car, Electric Vehicle, Rapid Transit, and Vehicle reserve 6% of ontological elements. The context bike reserves 5% of ontological concepts. The context parking reserves 7% of ontological concepts and the context GPS reserves 8% of ontological concepts. The context cargo ship and flight reserves 2% of ontol ogical concepts. The context Transport reserves the highest count of ontological concepts. The context of road traffic reserves the second-highest count of ontological concepts.

Figure 8. shows knowledge expansion methodology generated ontology concepts percentage based on dominant words. The context aircraft, crane, bike, bus, and electric vehicle reserves 4% of ontology concepts. The context car and road traffic reserve 5% of ontology concepts. The context airport, and GPS

reserves 6% of ontology concepts. The context flight, train, rapid transit, parking, and transport reserves 7% of ontology concepts. The context ambulance and bicycle reserve 3% of ontology concepts. The context cargo ship has the second-highest percentage (8%) of ontology concepts. The context vehicle has the highest percentage (9%) of ontology concepts. The aircraft and flight contextual ontology concept reuse 8% which shows mentioned IoT application are compared to the context of transport and road traffic is very less.

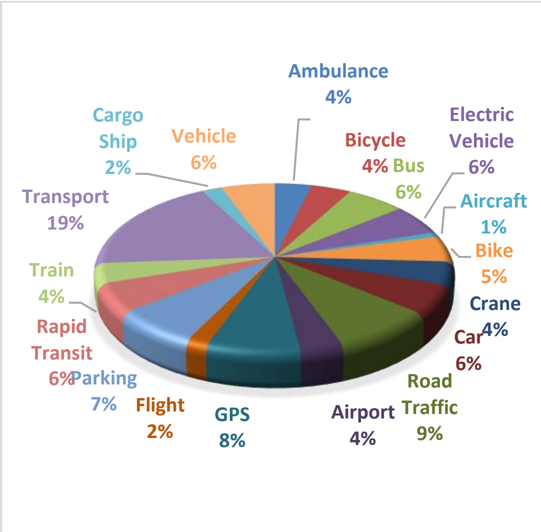

Figure 7. Generation of ontology concepts based on ontological inputs

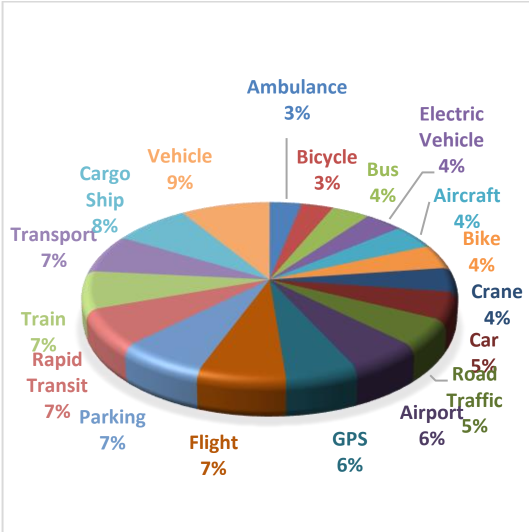

Figure 8. Generation of ontology concepts based on dominant words inputs

Figure 9. describes the percentage of ontology match between generated ontology concepts and popular IoT applications ontologies like SSN ontology, LOV4IoT, M3 ontology. The knowledge extraction methodology [25] produces 16300 concepts that support more interoperability. The knowledge expansion methodology 1422 contextual concepts generated, but this method also supports reuse of ontology concepts. In the case of ontology concept reuse is higher based on a hybrid methodology that is knowledge extraction methodology and knowledge expansion methodology explored concepts concerning ontological elements and dominant terms. It gives maximum interoperability in the various context of transport-related IoT application domains.

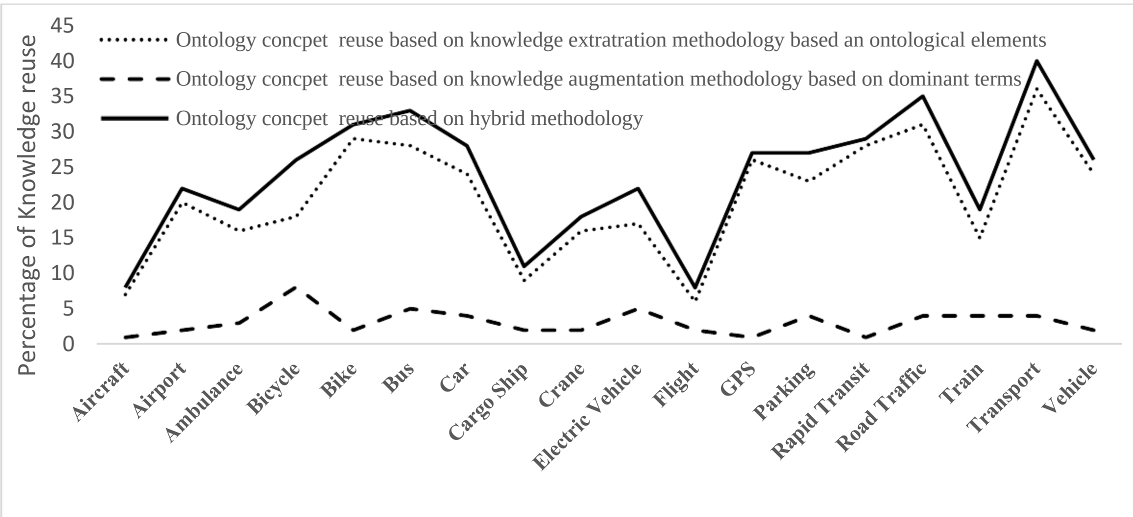

Figure 9. IoT application reusability based on ontological inputs and dominant words inputs

6. CONCLUSION AND FUTURE WORK

The IoT application-related ontologies are identified from the ontology catalogue and repository. Some of the IoT applications ontologies are publicly unavailable or inaccessible. Those IoT application-related research article frequent terms are identified and that are passed into word2vector NLP corpus to generate similar terms which are annotated in ontology as an ontology concept. Then these concepts are added into existing clustered ontological terms which produce a maximum of 40% reusability for the transport domain. In the future, the developed ontology concepts can be converted into a knowledge graph that can be used for communication, decision making, and notification generation.

REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *in 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, Jan. 2013, doi: 10.48550/arXiv.1301.3781.

[2] J. Chen and J. Gu, “ADOL: a novel framework for automatic domain ontology learning,” *Journal of Supercomputing*, vol. 77, no. 1, pp. 152–169, Jan. 2021, doi: 10.1007/s11227-020-03261-7.

[3] L. Olsina, “Applicability of a foundational ontology to semantically enrich the core and domain ontologies,” in *13th International Conference on Knowledge Engineering and Ontology Development (KEOD)*, 2021, pp. 111–119, doi: 10.5220/0010641000003064.

[4] K. Janowicz, M. H. Rasmussen, M. Lefrançois, G. F. Schneider, and P. Pauwels, “BOT: The building topology ontology of the W3C linked building data group,” *Semantic Web*, vol. 12, no. 1, pp. 143–161, Nov. 2020, doi: 10.3233/SW-200385.

[5] H. Akremi and S. Zghal, “DOF: a generic approach of domain ontology fuzzification,” *Frontiers of Computer Science*, vol. 15, no. 3, p. 153322, Jun. 2021, doi: 10.1007/s11704-020-9354-z.

[6] R. Kumaravel, S. Selvaraj, and C. Mala, “A multidomain layered approach in development of industrial ontology to support domain identification for unstructured text,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4033–4044, Sep. 2018, doi: 10.1109/TII.2018.2835567.

[7] X. Xue and J. Lu, “A compact brain storm algorithm for matching ontologies,” *IEEE Access*, vol. 8, pp. 43898–43907, 2020, doi: 10.1109/ACCESS.2020.2977763.

[8] A. Sattar, M. N. Ahmad, E. S. M. Surin, and A. K. Mahmood, “An improved methodology for collaborative construction of reusable, localized, and shareable ontology,” *IEEE Access*, vol. 9, pp. 17463–17484, 2021, doi: 10.1109/ACCESS.2021.3054412.

[9] M. Larrañaga, A. Conde, I. Calvo, J. A. Elorriaga, and A. Arruarte, “Automatic generation of the domain module from electronic textbooks: Method and validation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 69–82, Jan. 2014, doi: 10.1109/TKDE.2013.36.

[10] J. Zhou, X. Song, Y. Li, Y. Gao, and X. Zhang, “Building real-time ontology based on adaptive filter for multi-domain knowledge organization,” *IEEE Access*, vol. 9, pp. 66486–66497, 2021, doi: 10.1109/ACCESS.2021.3076833.

[11] J. Liu, X. Zhang, Y. Li, J. Wang, and H. J. Kim, “Deep learning-based reasoning with multi-ontology for IoT applications,” *IEEE Access*, vol. 7, pp. 124688–124701, 2019, doi: 10.1109/ACCESS.2019.2937353.

[12] Z. Sun, C. Hu, C. Li, and L. Wu, “Domain ontology construction and evaluation for the entire process of software testing,” *IEEE Access*, vol. 8, pp. 205374–205385, 2020, doi: 10.1109/ACCESS.2020.3037188.

[13] A. Smirnov, T. Levashova, A. Ponomarev, and N. Shilov, “Methodology for multi-aspect ontology development: ontology for decision support based on human-machine collective intelligence,” *IEEE Access*, vol. 9, pp. 135167–135185, 2021, doi: 10.1109/ACCESS.2021.3116870.

[14] S. Valtolina, L. Ferrari, and M. Mesiti, “Ontology-based consistent specification of sensor data acquisition plans in cross-domain IoT platforms,” *IEEE Access*, vol. 7, pp. 176141–176169, 2019, doi: 10.1109/ACCESS.2019.2957855.

[15] N. Fathy, W. Gad, N. Badr, and M. Hashem, “ProGOMap: Automatic generation of mappings from property graphs to ontologies,” *IEEE Access*, vol. 9, pp. 113100–113116, 2021, doi: 10.1109/ACCESS.2021.3104293.

[16] D. Chen and H. Zhao, “Research on the method of extracting domain knowledge from the freebase RDF dumps,” *IEEE Access*, vol. 6, pp. 50306–50322, 2018, doi: 10.1109/ACCESS.2018.2868516.

[17] M. Asfand-E-Yar and R. Ali, “Semantic integration of heterogeneous databases of same domain using ontology,” *IEEE Access*, vol. 8, pp. 77903–77919, 2020, doi: 10.1109/ACCESS.2020.2988685.

[18] C. Patel, K. Supekar, Y. Lee, and E. K. Park, “OntoKhoj: a semantic web portal for ontology searching, ranking and classification,” in *Proceedings of the fifth ACM international workshop on Web information and data management - WIDM ’03*, 2003, p. 58, doi: 10.1145/956699.956712.

[19] J. Kim, P. Kim, and H. Chung, “Ontology construction using online ontologies based on selection, mapping and merging,” *International Journal of Web and Grid Services*, vol. 7, no. 2, pp. 170–189, 2011, doi: 10.1504/IJWGS.2011.040447.

[20] A. Anikin, D. Litovkin, M. Kultsova, E. Sarkisova, and T. Petrova, “Ontology visualization: Approaches and software tools for visual representation of large ontologies in learning,” in *Communications in Computer and Information Science*, vol. 754, 2017, pp. 133–149.

[21] Y. Zhang, W. Vasconcelos, and D. Sleeman, “OntoSearch: An ontology search engine,” in *Research and Development in Intelligent Systems XXI*, London: Springer London, 2007, pp. 58–69.

[22] J. Liu, H. Ren, M. Wu, J. Wang, and H. jin Kim, “Multiple relations extraction among multiple entities in unstructured text,” *Soft Computing*, vol. 22, no. 13, pp. 4295–4305, Jul. 2018, doi: 10.1007/s00500-017-2852-8.

[23] L. Xiang, Y. Li, W. Hao, P. Yang, and X. Shen, “Reversible natural language watermarking using synonym substitution and arithmetic coding,” *Computers, Materials and Continua*, vol. 55, no. 3, pp. 541–559, 2018, doi: 10.3970/cmc.2018.03510.

[24] Z. Zhou, G. Qi, Z. Wu, and J. Shi, “A platform-independent approach for parallel reasoning with OWL EL ontologies using graph representation,” in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, Nov. 2016, vol. 2016-January, pp. 80–87, doi: 10.1109/ICTAI.2015.25.

[25] S. Prakash and T. Poongodi, “Domain Interoperability for iot application through knowledge extraction,” *Indian Journal of Computer Science and Engineering*, vol. 13, no. 1, pp. 40–50, 2022, doi: 10.21817/indjcse/2022/v13i1/221301045.

BIOGRAPHIES OF AUTHORS

Prakash Shanmurthy 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 can be contacted at email: research4prakash@gmail.com.

Poongodi Thangamuthu is working as 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. She can be contacted at email: tpoongodi2730@gmail.com.

Balamurugan Balusamy is currently working as Professor in the School of Computing Sciences and Engineering at Galgotias University, Greater Noida, India. His contributions focues on Engineering Education. Blockchain, and Data Sciences. He can be contacted at email: kadavulai@gmail.com.

Seifedine Kadry has a Bachelor degree in 1999 from Lebanese University, MS degree in 2002 from Reims University (France) and EPFL (Lausanne), PhD in 2007 from Blaise Pascal University (France), HDR degree in 2017 from Rouen University. At present his research focuses on Data Science, education using technology, system prognostics. He can be contacted at email: skadry@gmail.com.