2020 5th International Conference on Computer and Communication Systems

Automatic Knowledge Graph Construction Based on Relational Data of Power Terminal Equipment

Zheng Su, Mukai Hao
Industry Development and Promotion Center
Ministry of Industry and Information Technology
Beijing 100864, China
e-mail: {suzheng, haomukai}@idpc.org.cn

Qiang Zhang, Bo Chai, Ting Zhao
Artificial Intelligence on Electric Power System State Grid Corporation Joint Laboratory (GEIRI)
Global Energy Interconnection Research Institute Co. Ltd, Beijing 102206, China
e-mail: {zhangqiang1, chaibo, zhaoting}@geiri.sgcc.com.cn

***Abstract*—In the last years, Knowledge Graph (KG) has been widely applied in domain knowledge modeling, and therefore has received more attention in the Ubiquitous Power Internet of Things (UPIoT) that consists of a large number of power terminal equipment and sensing devices. In this paper, we propose an automatic framework to construct power terminal equipment knowledge graph by extracting ontological information and instantiating objects from the relational data of power terminal equipment. First, the ontological concepts and the hierarchy between concepts are built by analyzing the data schemas of different types of devices and their properties. Then the knowledge graph for different types of device objects is constructed by instantiating every object and their binary relations. In addition, we give the relevant algorithms of automatically constructing the complete knowledge graph of power terminal equipment. We argue that the proposed approach will greatly facilitate the information sharing of multi-source heterogeneous data of power terminal equipment on the environment of UPIoT.**

Keywords-knowledge graph; power terminal equipment; ontology; relational data; ubiquitous power internet of things

I. INTRODUCTION

With the rapid development of information technology, power information systems are entering the stage of rapid development. State Grid Corporation of China (SGCC) put forward the "Strong Smart Grid Development Strategy" in 2009. After ten years of rapid development, on October 14 2019, SGCC released the Ubiquitous power Internet of things white paper 2019, which meant that SGCC has begun to accelerate the construction of the Ubiquitous Power Internet of Things (UPIoT) [1]. UPIoT is the backbone of the future power system, and consists of a large number of power terminal equipment (PTE) and sensing devices. Therefore, the construction of UPIoT has obviously become the core task of the future power systems. UPIoT aims to construct the intelligent, flexible and strong power systems and realize the interconnection of everything in the power system, using some advanced technology including Cloud Computing, Bigdata, Internet of Thing (IoT), Artificial Intelligence and Mobile Computing. It also provides the function of more intelligent human-machine interaction. As such, the stakeholders in the field of power systems including power generation, transmission, distribution, marketing and consummation, can make full use of UPIoT for extending their businesses. UPIoT can be characterized by the comprehensive perception of state, efficient processing of information and flexible convenient application. PTE plays an important role in UPIoT. PTE contains a lot of structured data and unstructured textual descriptions. All these data needs to be fused for accurately power system analysis such as fault diagnosis.

Knowledge graph (KG) [17, 18] is a uniform knowledge representation tool that describes structured and unstructured data based on graph model. As a core part of KG, ontology approach can be used to rapidly describe the concepts and their relationships in the physical world, which deals with building the ontological concepts, the hierarchy between them, the instantiation of objects and the extraction of object relations. Generally speaking, an ontology is a formalized, clear and detailed description of shared concept system [2]. Ontology knowledge base is the key to realize intelligent retrieval. The domain knowledge system provided by ontology can improve the accuracy of understanding of users' requirements and the description of resource, and it also can change the traditional retrieval mechanism from keyword matching to concept and content matching, so as to improve the efficiency of information retrieval system.

In this paper, we concentrate on the automatic construction of KG based on the structured (relational) data of PTE. We propose an automatic framework to construct PTE KG by extracting ontological information and instantiating objects from the relational data of PTE. The ontological concepts and the hierarchy between concepts are built by analyzing the data schemas of different types of devices and their properties. The KG for different types of device objects is constructed by instantiating every object and their binary relations. We also give the relevant algorithms of automatically constructing the complete KG of PTE.

We argue that constructing KG of PTE structured data is the first step towards fusing structured and unstructured multi-source PTE data on the environment of UPIoT.

978-1-7281-6136-5/20/$31.00 ©2020 IEEE

II. Related Work

The DEF5 method [3, 4] is an ontology description acquisition method, which provides a structural method for the development of high-precision domain ontology. The defect of this method is that it does not have the intention of circular development. IDEF5 ontology development process includes the following five activities [5]: organize and carry out activities, data collection, data analysis, initial ontology development, and improvement and verification of ontology.

The skeleton method [6, 7] is summarized from Enterprise Ontology (EO). This method clearly describes the basic process and guidelines of ontology development, which has an important guiding significance for the current practice of ontology development.

The TOVE method [8], is also called as evaluation method. It does not directly construct the logic knowledge model described with the form of ontology, but firstly establishes the informal description of ontology, and then formalizes the description. The steps of TOVE include the following [9]: obtaining the incentive plot, express the problem of informal ability clearly, standardization of terms, determining the problem of formal ability, determining the axioms and definitions in formal languages, and evaluating the integrity of ontology.

Tang et al. [10] proposed a method of building domain ontology prototype based on thesaurus, which combines the Enterprise method, the METH ONTOLOGY [11] method and the waterfall software development model. Jia et al. [12] proposed to grasp four basic principles when transforming thesaurus into ontology.

In the system of Chinese sharing thesaurus, Zeng et al. [13] formulated the rules which convert the Chinese Thesaurus to OWL ontology according to the characteristics, logic description and OWL grammar of the Chinese Thesaurus. In addition, the conversion program can automatically convert Chinese thesaurus to the OWL format.

The large-scale ontology environment prototype system (lode) [14] was proposed which is based on the Agricultural Thesaurus as the central knowledge base. A good ontology model means that the required domain ontology can finally be applied in practice when facing specific applications [15]. Ontology construction is generally made based on manual methods. Moreover, ontology construction method has not been a complete system yet [16].

In the field of power systems, although ontology methods have been applied, there is no ontology for PTE. Therefore, in this paper, we propose an automatic KG construction based on relational data of PTE.

III. The Proposed Frmework of Ontology Construction

A. Problem Statement

UPIoT is facing an obvious problem of integrating and sharing multi-source heterogeneous data due to the fact that UPIoT consists of a large number of PTE and sensing devices. These devices possibly are produced by different manufactures and have different types of protocols, hardware and software. The typical power terminal devices in UPIoT are DTU (distribution terminal unit), FTU (feeder terminal unit), LCT (Load Control Terminal) and TCU (Telematics Control Unit), and so on. These power terminal devices contain a large amount of data information, such as manufacturer name, production date, model type, voltage level, firmware, operating system and versions of the operating system, etc. These types of information can be stored in relational databases. On the other hand, the operational data and introductory information of PTE are often unstructured descriptive texts. For example, some unstructured information for device loopholes can be often found in some websites of device manufactures and vendor, or in some web pages about information security and industrial control security. These multi-source heterogeneous data mentioned above makes the knowledge sharing, data retrieval and analysis more difficult. In the situation, it is necessary for better analyzing the operation state of UPIoT to fuse these multi-source heterogeneous data by constructing the KG for these different types of power terminal devices.

A KG for PTE is possibly from the relational data and unstructured data. Structured data for PTE is stored in the relation database in server or device storage. It often includes different attributes of different types of power terminal devices, such as the names, manufactures, model types, firmware types, system version, etc. In contrast, the unstructured data is often from the device vendors and manufactures who need to record the information about certain devices. These unstructured data can be often obtained in the websites. As the first step toward the KG fusion of structured relational data and unstructured textual data of PTE, we in this paper will concentrate on the automatic construction of KG based on relational data.

Due to the fact that DTU, FTU, LCT and TCU are the typical power terminal devices in UPIoT, we will construct KG from these PTE. So the structure of the DTU, FTU, LCT and TCU data is important for the construction of KG. Table I and Table II are two simple examples of DTU and FTU data.

Tables I and II respectively are the simple segments of the DTU and FTU data stored in relational database. As we see in the two tables, both the DTU and FTU have many attributes including device name, voltage level, model name and so on, some other attributes omitted are not shown in the table above. With contrast to the two tables, we can find that DTU and FTU have attributes including some common properties and some exclusive properties. For example, the instance “Baihui Number 1 Ring Main Unit” in Table I is the device name of a device object of DTU, and its’ voltage level rates AC220V, and it’s model number is PDZ821. Besides, batch number, manufacturer name and manufacture date are the common attributes in DTU, but operating system and operating system version are the unique properties in DTU. As to Table II, FTU also has attributes like device name, voltage level, model number, batch number, manufacturer name and manufacture date, but main program version of firmware and terms of an agreement and version are the unique attributes in FTU.

TABLE I. AN EXAMPLE SEGMENT OF DTU TABLE

Device Name	Voltage Level	Model Number	…	Manufacture Date	Operating System	Operating System Version
Baihui Number 1 Ring Main Unit	AC220V	PDZ821	…	2012.06	Linux	2.6.29.1
…	…	…	…	…	…	…

TABLE II. AN EXAMPLE SEGMENT OF FTU TABLE

Device Name	Voltage Level	Model Number	…	Manufacture Date	Main Program Version of Firmware	Terms of An Agreement And Version
Yiqingyi Line Yichang Main Unit 1	AC220V	PDZ821	…	2012.06	SV56.023	IEC-104 2002
…	…	…	…	…	…	…

A record in database represents a device instance. For the other table in database like FTU, TCU are similar to the DTU and FTU data shown above. Both of them have the same attributes. As for the TCU table, its' common attributes are the same as DTU's, but TCU is divided into billing control unit and control unit, both of which have manufacturer, model, operating system, main program version, protocol and version used.

Formally speaking, a PTE database $DB=\{T_1, T_2, T_3,...,T_n\}$ is the input of our method, which consists of multiple tables T_i, ($1 \le i \le n$), and each $T_i = (P_i, I_i)$, where P_i and I_i are respectively the sets of attributes and instances (i.e., records). Our method will output a KG $G=(V, E)$, where V and E are respectively the sets of nodes and edges.

B. Framework of Knowledge Graph Construction

Both relational databases and ontologies are an organization and description model of knowledge. Data records are used to describe the relationship in relational database. Tables and attributes in relational database may respectively correspond to the concept and property of the ontology. If an inheritance relation exists between tables or columns of database, then it also exists between two concepts of ontology. The records in the database can be transformed into instances of ontology. Automatic translation is feasible from relational database to ontology.

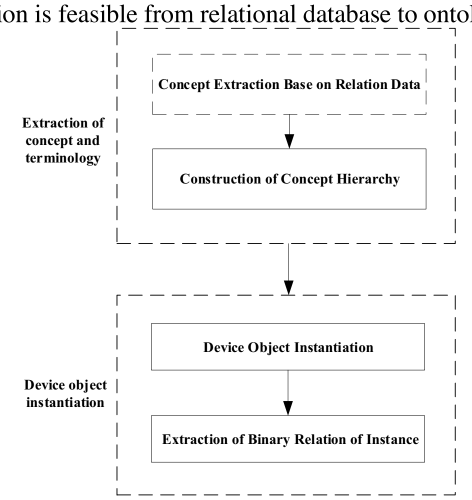

Figure 1. Ontology construction for knowledge graph

The construction of ontology is mainly divided into two steps: the construction of concept layer and instance layer. The specific construction procedure of ontological KG is shown in the Figure 1.

IV. CONSTRUCTION OF CONCEPTS AND TERMINOLOGY

A. Concept Extration Based on Relational Data

A KG is a graph with many vertexes and edges, so the structure of the ontology can be described with two sets, V and E, which respectively represents the set of entities and the set of relations. V and E can be further described by triples $< s, p, o >$, s represents the subject, p represents the property, o represents the object. If we need to define a relation, we can define a subject which is the domain of the relation, and an object which is the range of the relation. For the data in database, attribute or relation is the p in triples, the domain of attribute or relation is the s in triples and the range of attribute or relation is o in triples. A class or entity could be the subject in a triple, while being the object in another triple. An ontology can be regarded as a knowledge graph by using the graph based representation model such as RDF graph that consists of many triples.

To obtain the triples of the form (s, p, o), we need to analyze and extract the core concepts of PTE such as the concept terms of equipment and the related attributes of equipment. Because the data is stored in relational database, the concepts and relations can be extracted with the rules in relational database as follows.

Rules 1: every table name can be extracted as a class or concept. For example, a table in database named "DTU", which is a kind of PTE including many different types of devices, could be define as a DTU class.

Rules 2: every PTE has many properties, such as device name, manufacturer name, manufacture date, model level, operating system, operating system version, etc. The attributes in database can correspond to a property in the ontology. As a result, each the column of attributes in the table can be extracted as a class. For example, the attribute "Operating System" can be used as an object property, and the column corresponding to "Operating System" will be extracted as a class whose name is "Operating System Name" in the ontology.

Rules 3: relations possibly exist between two different entities including classes or objects. We can extract a property class or entity, which can describe the relations with domains and ranges. For example, operating system is a attribute in DTU table, we can define a “has operating system” class, whose domain is DTU, and range is operating system.

According to the above rules, we develop the following Algorithm 1. In this algorithm, the notion *DB* is the input that represents all the different types of device tables in the database, and the output is a RDF Graph *G*, which includes a set of vertices (V) and a set of edges (E). Then, we need to traverse each table in database, and use the name of table as a class (C_{Ti}) that corresponds to table T_i, which is described as an entity in ontology. All the entities are the vertices in the V. The next step is to get the properties of the table as P_j, and the value of P_j is C_{Pj},. Then we define its domain (C_{Ti}) and range (C_{Pj}), which would build relationships belonging to the E.

Algorithm 1: PTE Conceptual Terminology Extraction

```
Input:
DB={T1, T2, T3...Tn}
Output:
G = (V, E)
i ← 1, V ← ∅
for each Ti∈DB do
        CTi ← Ti.name
        V ← CTi
        for each Pj∈Ti.Property do
        V ←CPj
        E ← (Pj, domain, CTi )
        CPj ← Pj.column
        E ← (Pj, range, CPj )
return G
```

B. Construction of Concept Hierarchy

As Tables I and II show, DTU, FTU and other device tables in PTE contain many specific device instances (records in a table) that associated with many properties. And each property can be possibly associated with many instances. What is worth noting that there are many common properties in PTE. If different types of devices include many common properties then it would be better to assign the common properties to a more abstract device concept. Otherwise, it would largely cause data redundancy. Different kinds of devices would have their instances for each of properties which include common and exclusive properties. To reduce the redundancy, we explored a method to reduce redundancy. We can extract the common properties into the properties in a higher layer such that each device only keeps the exclusive properties of their own.

So we need to extract these concepts and terms hierarchically, and then extract the concepts of common attribute of different PTE into a new class. And this new class is as the lower layer of ontology. The remaining different power terminal devices have their own properties. The lower layered classes will be used as the subclasses of the new class. This step can be made in an iterative manner until no common attribute can be extracted from different power terminal devices.

Specifically, we extract the same attribute concept from different table data of PTE to build the top-level concept of PTE, and each specific PTE, such as DTU, FTU, TCU, is a subclass of the top-level concept. For example, we extract the common attributes from PTE, such as equipment model, equipment name, factory date, batch, and manufacturer, and so on, as the properties of the top-level concept of power equipment. For examples, DTU, FTU, TCU and LCT are the subclass of the top concept entity which is named “power device” covering many kinds of different devices. The top concept power device has many properties which are common in DTU, FTU, TCU and LCT, such as device name, voltage level, model name, batch number, manufacturer name and manufacture date.

Due to fact that each value of property corresponds to a specific data type, define data type as a property class, whose domain is property and whose range is corresponding value. For example, DTU has some properties such as model level, operating system, and so on. The data types of model level, operating system, system version, firmware version and operating system version are all the string data type.

V. Construction of instance KG

A. Device Object Instantiation

Semantically, an instance is a concrete object of a class, which is mainly an abstract description of a concrete object in the instance layer. The construction of ontology instance layer can be mainly divided into two steps where device objection instantiation is first step.

In the following, considering that the concept layer can be defined as sets of concepts and the hierarchy between them, so the object instances could be associated with classes and properties in the concept layer.

To instantiate device objects, we need to extract equipment instance. For example, the name of table of power terminal device is as an instance class, and each column of the table serves as an attribute class of the instance. The name of the device instance class is usually the name of the corresponding device concept class plus the serial number. Then we need to instantiate the property class, for example, each attribute in table could be instantiate with its’ attribute name, like operating system is an attribute of DTU, so operating system is one of the instance of property class.

B. Extraction of Binary Relation of Instance

The instance of binary relation can be described by a triple (s, p, o). With instantiation of objects, it will construct four kinds of classes. A relation exists in two different classes, so there are three main binary relations. First one is between device and individuals of device, second one is between device and property, last one is between property and data type.

First, we make instantiation of the individuals of devices and the relations between individuals. For example, the dtu1 (an instance of distribution terminal unit) is the instance of power device class, which can be represented as the triple <dtu1, type, Power Device>.

Next, we begin to instantiate the binary relationships between instances. Each property corresponds to the attribute in database, and the specific data in each cell of each column in the table can be extracted as a corresponding class named data type. The column name of each column can be extracted as the name of an attribute class. For example, the individual dtu1 in DTU has property “device name”, and its corresponding value is “Baihui Number 1 Ring Main Unit”, which can be represented as the triple <dtu1, device name, “Baihui Number 1 Ring Main Unit”>.

Then, we specify the data type for each value in terms of the binary relationship of property. Each value has a corresponding data type, such as “Baihui Number 1 Ring Main Unit” has data type “xsd:string”, which is represented as < “Baihui Number 1 Ring Main Unit”, datatype, string>.

With the above analysis, we give the following Algorithm 2. The $T = (P, I)$ means the set of instances I with its corresponding property set P in table T. And C_T is the class that an instance *ins* belongs to, υ is the value of some property P_i related to *ins*, and A_{Pi} is the datatype of P_i.

Algorithm 2. PTE Object Instantiation

Input:
$T = (P, I)$
Output:
$G = (V, E)$
for each $ins \in I$ **do**
    $V \leftarrow ins$
    $E \leftarrow (ins, type, C_T)$
    for each $P_i \in P$ **do**
        $E \leftarrow (ins, P_i, \upsilon)$
        $E \leftarrow (\upsilon, datatype, A_{Pi})$
return G

The related algorithms were implemented based on the following toolsets including Python3.7, owlready2 and pandas library. With the performing of the above code, we could build the KG of the instances from DTU and the instances from FTU.

VI. CONCLUSION

In this paper, we proposed an automatic construction framework that builds the KG for a relational database including multiple tables from power terminal equipment. Two steps were introduced including the extraction of concept terminology and the instantiation of power terminal devices. The related algorithms were developed for automatic construction of KG of power terminal equipment. The constructed KG will be used for fault diagnosis of power terminal equipment in the future.

In the future work, we will also explore to construct the knowledge graph of power terminal equipment by mining and extracting the unstructured textual information, and further fuse them with the KG based on relational data.

ACKNOWLEDGMENT

This work is supported by State Grid Company Research Project “Task-Driven Multi-round Dialogue Generation Based on Semantic Understanding” (No.5455HJ190008).

REFERENCES

[1] Yajian Zhang, Ting Yang, Guangyu Meng, "Review and Prospect of Ubiquitous Power Internet of Things in Smart Distribution System", Electric Power Construction, vol.40, no.6, pp.1-12, 2019.

[2] Xiaohui Fan, Chenguang Shi, “Overview of ontology construction research”, Naval electronic engineering, vol.6, pp.15-18, 53, 2011.

[3] E. Zavitsanos, G. Paliouras, G.A. Vouros, "Gold Standard Evaluation of Ontology Learning Methods through Ontology Transformation and Alignment", IEEE Transactions on Knowledge and Data Engineering, vol.23, no.11, pp.1635-1648, 2011.

[4] R.J. Mayer, J.W. Crump, R. Fernandes, A. Keen, M.K. Painter, "Information integration for concurrent engineering (IICE) compendium of methods report", Wright-Patterson Air Force Base, Ohio45433-7604, http://www.idef.com/pdf/compendium.pdf, 2006.

[5] Darius Jurkevicius, Olegas Vasilecas, "Ontology creation using formal concepts approach", Proc. CompSysTech 2010, pp.64-70, 2010.

[6] C.S. Lee, Y.F. Kao, Y.H. Kuo, M.H. Wang, "Automated ontology construction for unstructured text documents", Data & Knowledge Engineering, vol.60, no.3, pp.547-566, 2007.

[7] Dilek Kucuk and Yusuf Arslan, "Semi-automatic construction of a domain ontology for wind energy using Wikipedia articles", Renewable Energy, 62:484-489, 2014.

[8] Fan Yu, “Domain ontology construction method and case study”, Wuhan University Press, 2015.

[9] Roberto Navigli, Simone Paolo Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network", Artificial Intelligence, vol.193, pp.217-250, 2012.

[10] Z. Zhen, Aimin Tang, Junyi Shen, Shengjun Lu, “The saurus-based approach for building domain ontology with a case study of military aircraft prototype ontology construction”, Journal of Southeast University (English Edition), vol. 9, pp. 353-356, 2006.

[11] J.Z. Jia, “Simple knowledge organization system and Chinese Thesaurus”, Journal of China Library, 2008.

[12] X.H. Zeng, “Owl representation and deep semantic revelation of Chinese Classified Thesaurus”, Journal of information science, 24 (2): 151-160, 2005.

[13] J. Li, “Research on construction method and application of domain ontology”, Beijing: Institute of agricultural information, Chinese Academy of Agricultural Sciences, 2009

[14] Y.S. Liu, “Research on ontology construction methods and development tools”, Modern intelligence, vol.9, pp. 17-24, 2009.

[15] A. Albarghothi, W. Saber, K. Shaalan, "Automatic Construction of E-Government Services Ontology from Arabic Webpages", Procedia Computer Science, vol.142, pp.104-113, 2018.

[16] H.F. Wang, G.L. Qi, H.J. Chen, “Knowledge Map: Method, Practice and Application”, Beijing: Electronic Industry Press, 2019.

[17] R.Q. Lu, C.Q. Fei, C.Q. Wang, S.F. Gao, "HAPE: A programmable big knowledge graph platform", Information Sciences, vol.509, pp.87-103, 2020.