

## Research Article

# Device-Oriented Automatic Semantic Annotation in IoT

Fagui Liu, Ping Li, and Dacheng Deng

School of Computer Science & Engineering, South China University of Technology, Guangzhou, China

Correspondence should be addressed to Ping Li; 18813759473@163.com

Received 11 January 2017; Revised 11 April 2017; Accepted 27 April 2017; Published 21 June 2017

Academic Editor: Houbing Song

Copyright © 2017 Fagui Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Semantic technologies are the keys to address the problem of information interaction between assorted, heterogeneous, and distributed devices in the Internet of Things (IoT). Semantic annotation of IoT devices is the foundation of IoT semantics. However, the large amount of devices has led to the inadequacy of the manual semantic annotation and stressed the urgency into the research of automatic semantic annotation. To overcome these limitations, a device-oriented automatic semantic annotation method is proposed to annotate IoT devices' information. The processes and corresponding algorithms of the automatic semantic annotation method are presented in detail, including the information extraction, text classification, property information division, semantic label selection, and information integration. Experiments show that our method is effective for the automatic semantic annotation to IoT devices' information. In addition, compared to a typical rule-based method, the comparison experiment demonstrates that our approach outperforms this baseline method with respect to the precision and  $F$ -measure.

## 1. Introduction

The Internet of Things (IoT) is a new dynamic network generated by information communication between people and things [1], which is capable of realizing the information exchange and seamless connection among IoT entities [2]. It enables IoT entities possessing sensorial and computing capabilities to work together efficiently [3] and provides a new way for the fine management, operation, and maintenance of smart city [4]. To enhance the intelligent interoperability in heterogeneous environments [5], semantic technologies are always applied to facilitate the semantic data access and integration, semantic reasoning, and knowledge extraction [6], so that the information in IoT can be understood by machines. For example, as an extension of Internet, semantic Web applies XML, RDF, and ontology technologies to semantically annotate the resources and information on the traditional Internet. Ontology is a conceptualized and formalized specification of domain knowledge. Moreover, ontology individuals are instances of ontology. As a key index in semantic Web, semantic similarity is applied in many fields including semantic Web service discovery [7], semantic Web service clustering [8], and P2P grids [9]. In the service-oriented architecture, to improve the ability of collaboration between heterogeneous entities, the function

of entities and data from the physical world are described by the forms of semantic services accessed by unified interface. Consequently, the semantization and servitization of IoT are able to promote the automation and dynamism of entity discovery, selection, negotiation, and so on. As one of the most important semantic technologies, semantic annotation is the key ingredient to make the information in IoT machinery understandable and to acquire semantic IoT services.

Semantic annotation in the area of text annotation is the process of associating machine-understandable labels (i.e., semantic information, ontology concepts' URI) to a word or a sentence from text [10]. Similarly, semantic annotation for IoT entities, especially for IoT devices, can be treated as the process to annotate IoT entities with semantic labels and further transform them into semantic IoT services. In this way, they can be depicted by the unified and rich semantic forms and support semantic service discovery. Along with the development of wireless network technology, the number of IoT devices, a typical kind of IoT entities, is in a rapid growth. It is estimated that there will be around 50 billion IoT devices by 2020 [11]. Due to the large-scale and heterogeneity feature of data flows generated by IoT [12] and continuous changes in the state of IoT devices as well as data and volatility of IoT environments, semantic data handling in IoT becomes

more challenging and fraught with technical difficulties. Recently, the researches on semantic annotation mainly focus on manual or semiautomated annotation [2, 13–18]. Since the manual or semiautomated annotation methods for such massive amount of IoT devices are often inefficient, the automated semantic annotation of IoT devices is becoming a challenging issue to be addressed.

The purpose of this paper is to describe a device-oriented automatic semantic annotation method in IoT, including a series of processes and corresponding algorithms. The remainder of this paper is organized as follows. Section 2 mainly introduces the related work of semantic annotation and Section 3 provides a device description framework in IoT. The process and corresponding algorithms of automatic semantic annotation of IoT devices are presented in Section 4. The experiments of our methods, analysis of experiment parameters, and method comparison are described in Section 5. We close the paper by describing some conclusions and presenting our future work.

## 2. Related Work

In the past several decades, the main concentration of the researches on semantic annotation is semantic annotation tools and platforms, semantic annotation of Web documents, and semantic annotation in IoT. In particular, semantic annotation of Web documents occupies the majority of all researches. Semantic annotation tools and platforms mainly consist of two categories: pattern-based tools and machine learning-based tools. While pattern-based tools include GATE (<https://gate.ac.uk/>), AeroDAML [21], AeroSWAR [10], and SMT [22], machine learning-based tools contain MnM [23], Armadillo [10], and so on.

Semantic annotation of Web documents transforms Web content into semantic Web documents. De Maio et al. [10] proposed a fuzzy-based automatic semantic annotation method (FBASAM) of Web documents based on formal concept analysis and relational concept analysis. The approach is that, starting from Web resources, content with a high level of abstraction is obtained: concepts, connections between concepts, and instance-population are identified and arranged into ontology. The framework is designed to process resources from different sources and to generate an ontology-based annotation. Charton et al. [19] proposed an automated semantic annotation method for named entities (ASAM4NE). The method is based on an algorithm that compares the set of words appeared before and after the named entities with the content of Wikipedia articles and identifies the most relevant one by means of a similarity measure. Then, it establishes a connection between the named entities and some URI in the semantic Web. Diallo et al. [20] proposed an ontology-based semantic annotation approach (OBSAA) to automate the semantic annotation of texts using Natural Language Processing (NLP) technology. Based on concept frequency (TF) and inverse document frequency (IDF), the method selects ontology concepts from an existing biomedical ontology to semantic annotate texts. Rong [6] summarized seven semantic annotation methods of Web documents and proposed a similar rule strategy method

(SRSM) and a method on the basis of tree conditional random fields (MTCRF).

Currently, a few of existing researches on semantic annotation in IoT focus on sensor network data. Barnaghi et al. [13] discussed a semantic model (SM2SS) to describe the sensor streams and to demonstrate how data from sensor streams can be published, indexed, queried, and discovered in a distributed network. Kolozali et al. [14] proposed a knowledge-based approach for real-time IoT data stream (KBA4IoTDS) annotation and processing. The framework aims to support semantic annotation of IoT stream data by taking dimensionality and reliability into account to enable delivery of large volume of data using Message Queuing Protocol (AMQP). Wei and Barnaghi [15] discussed a semantic annotation method of sensor data (SAM4SD) and focused on the idea of semantic sensor Web by extending the discussion of semantic annotation using concepts taken from various domain ontologies. Chenyi [16] proposed a service-oriented entity semantic annotation framework (SOESAF), which manually annotates the function, state, and basic information of entities. It discussed a semantic annotation ontology model of IoT entities, which manually packages the information of IoT entities to Web services and annotates the function of IoT entities using Web services after clustering [8]. Bing [17] proposed a semantic annotation method for IoT documents (SAM4IoTDoc). This method selects an appropriate concept in ontology to add semantic information to files (documents, pictures, etc. in IoT). Junling et al. [2] created a template of IoT resource description to facilitate resource semantic annotation. Ming [18] proposed a semantic annotation method for WSDL files of Web services (SAM4WSDL). This method classifies Web services into particular domain ontology. In addition to text annotation, semantic annotation of Web services also needs to match the Web service interfaces of domain ontologies according to user input/output data and function descriptions.

In previous researches on semantic annotation, the researches have focused on the semantic of Web documents, and a few researches pay attention to semantic annotation in the environment of IoT. As shown in Table 1, we have compared the previous semantic annotation methods in five aspects: “Automatic,” “Training Set,” “Application Domain,” “Data Type,” and “Main Technology.”

Table 1 shows the comparison results of many semantic annotation methods from five aspects and indicates the following:

- (1) Most of automatic semantic annotation methods focus on the Internet field and are applied for Web documents.
- (2) The researches of semantic annotation methods for Web documents mainly pay attention to automatic semantic annotation methods.
- (3) Most of the researches on semantic annotation methods in the environment of IoT are manual annotation semantic methods. Moreover, they primarily focus on data models and annotation frameworks.

TABLE 1: Comparison of semantic annotation methods.

Methods	Automatic (yes/no)	Training Set (yes/no)	Application Domain	Data Type	Main Technology
FBASAM [10]	Yes	No	Internet	Web documents	Rule, formal, and relational concept analysis
ASAM4NE [19]	Yes	No	Internet	Web documents	Semantic similarity, linked data
OBSAA [20]	Yes	No	Biomedicine	Biomedical texts	NLP, TF-IDF
SRSM and MTCRF [6]	Yes	No	Internet	Web documents	Rule, CRFs
SM2SS [13]	No	Yes	IoT Sensor Network	Sensor networks	Sensor streams model
KBA4IoTDS [14]	No	No	IoT	IoT data streams	IoT data model
SAM4SD [15]	No	Yes	IoT Sensor Network	Sensor networks	Sensor streams model
SOESAF [16]	No	No	IoT	IoT entity information	Entity semantic annotation framework
SAM4IoT [17]	No	No	IoT	Documents	Rule
SAM4WSDL [18]	No	No	IoT	WSDL files of Web services	Rule, machine learning

In summary, the existing semantic annotation tools and platforms are mainly utilized for the annotation of Web documents, and the results are single or multiple independent semantic ontology resources. Those resources cannot be organized structurally. Therefore, the tools and platforms are not suitable for IoT devices whose resources should be organized structurally. Besides, existing semantic annotation methods mainly focus on Web documents whose annotation objects are Web documents. They do not meet users' requirements when annotating the information of IoT devices due to physical properties of IoT devices (space, time, environment, etc.). The researches on semantic annotation in IoT mainly concentrate on sensor data and manual annotation methods. However, manual or semiautomatic semantic annotation methods are often inefficient for numerous IoT devices and unable to meet the demands of semantic annotation in IoT. Thus, the existing semantic annotation methods of Web documents and IoT are not suitable for the massive amount of IoT devices. Automatic semantic annotation methods in IoT remain a central challenge to be addressed.

### 3. Our Device Description Framework in IoT

As the basis of automatic semantic annotation of IoT devices, device description framework is a description pattern of devices' information. The device description framework in IoT relies on the characteristics of IoT devices. Although the definition of IoT devices is different from different perspective of IoT, they commonly have the following several characters:

- (1) An IoT device should be provided with a unique identification.
- (2) An IoT device can be accessed through information networks via the communication interface.

- (3) Spatial-temporal characteristics.
- (4) IoT devices have computing power and storage ability.
- (5) IoT devices can not only obtain information from the surrounding environment but also process this information.

The nature of IoT is the bridge of the physical and information world. In this paper, IoT devices are classified into three categories: sensor devices, processor devices, and actuator devices. Sensor devices correspond to device between the physical world and information world. Processor devices refer to the information world and information world. Actuator devices associate with the information world and physical world. According to the characteristics of IoT devices, we propose a device description framework in IoT to describe IoT devices, as shown in Figure 1.

Figure 1 illustrates multiple components of the device description framework. The arrows in Figure 1 refer to the relationship in device ontology. For example, the arrow "hasIdentification" means that device concept in device ontology has an attribute "Identification." The details of each component are shown as follows:

- (1) *Identification*. It provides recognition of description information for IoT devices and is applied to describe the identity characteristics of IoT devices. A device can obtain a unique identification when it is associated with IoT.
- (2) *Performance*. It refers to the technical specifications, operating parameters, voltage, and so on. It is applied to describe some characteristics of IoT devices, such as computing power, storage ability, and energy efficiency.
- (3) *Function*: it identifies the function description of devices and is an important basis of user queries

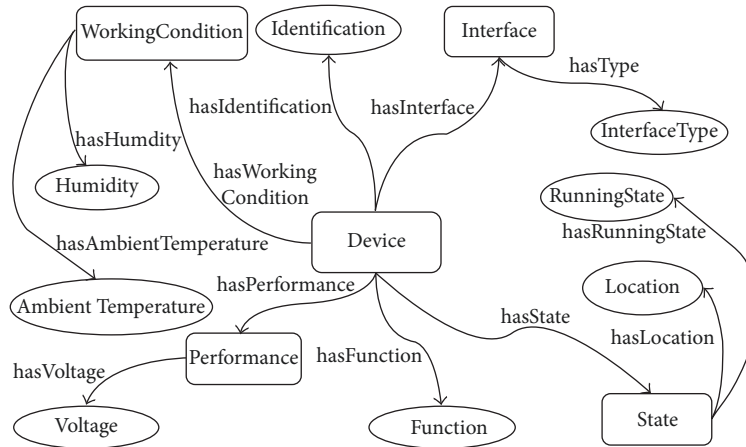


FIGURE 1: Device description framework in IoT.

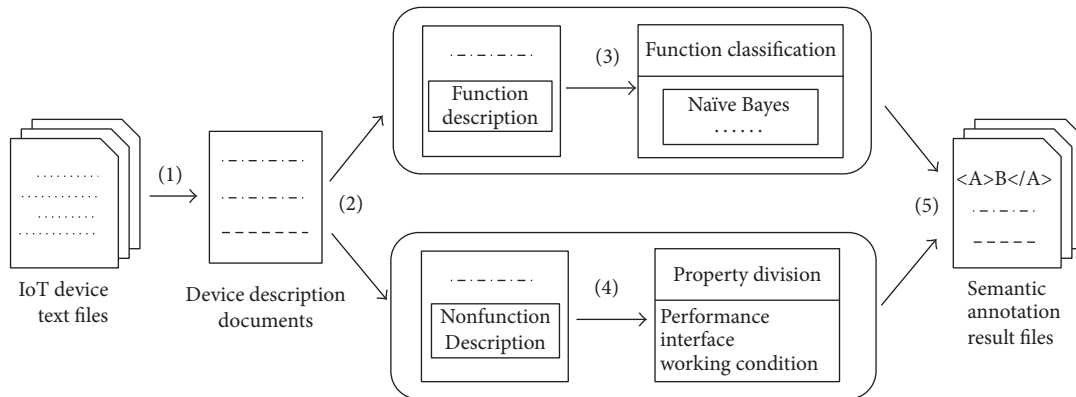


FIGURE 2: The process of automatic semantic annotation in IoT.

and device discovery, including input, output, and profile.

- (4) *State*. It is applied to describe the devices' state in IoT. The state of a device is generated from hardware devices which monitor this device in real-time. It relates to spatial-temporal characteristics of IoT devices.
- (5) *Interface*. It describes the interface and the communication between devices and networks, including access method. When a device is accessed to IoT, the device can obtain the interface information, such as Bluetooth and IP. It relates to the communication interface of IoT devices.
- (6) *Working Condition*. It indicates the surrounding environment for devices' normal work, including temperature, humidity, operating voltage, and working current.

The state component above contains some dynamic characteristics, such as mobility, location, and other characteristics that embody the space, time, and environment characteristics of IoT devices.

## 4. Our Automatic Semantic Annotation Approach in IoT

**4.1. The Process of Automatic Semantic Annotation.** The semantic annotation of IoT devices' information can be considered as the process that extracts special information from this piece of information and marks the information of IoT devices with semantic labels. It needs to address five issues as follows: (1) the representation and description of IoT devices' information, (2) the extraction of key information, (3) the selection of semantic labels, (4) the generating of device ontology, and (5) the expansion of device ontology. The process of automatic semantic annotation in IoT is shown in Figure 2.

The process of automatic semantic annotation in IoT consists of the following five steps:

- (1) *Preprocessing*. The text information of IoT devices, such as instructions, contains some information which users are not interested in, such as the specific internal structure, outline, and specific installation process. Thus, the text information should be filtered manually. Only the text information that describes devices' function and some technical parameters

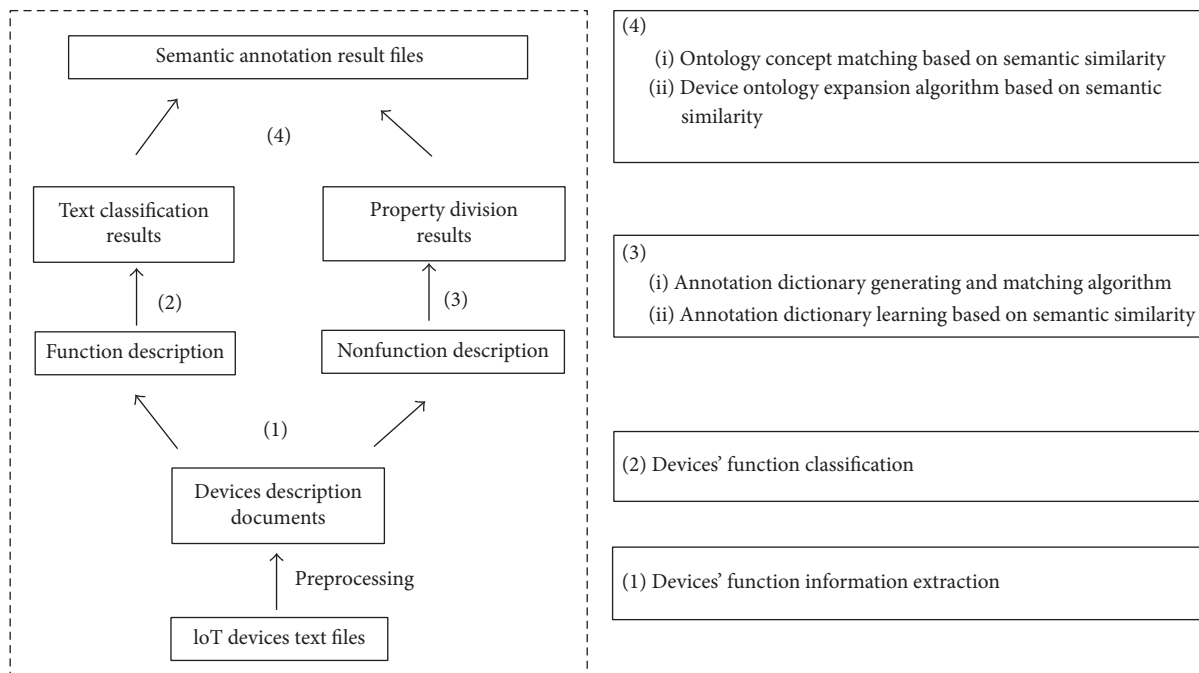


FIGURE 3: The algorithms of each process.

remained. Each message in the filtered text information occupies a row. This step is shown in step (1) in Figure 2.

- (2) *The Information Extraction of Devices' Function.* While the information about function is unformatted and disorganized texts, however, there are three types of IoT devices. Therefore, the goal of this step, shown as step (2) in Figure 2, is to divide devices' information into two components: function description and non-function description. The two components are dealt with in different approaches.
- (3) *The Information Classification of Devices' Function.* According to the description of step (2), devices need to be classified using devices' function description. This is the scope of NLP. The purpose of this step, shown as step (3) in Figure 2, is to classify devices' function description using text processing technologies.
- (4) *Property Information Division.* There are five properties in our device description framework. After the classification of function description in step (3), the information of other properties is dispersed in non-function description, shown as step (4) in Figure 2.
- (5) *Information Integration and Semantic Label Selection.* The aim of this step (shown as step (5) in Figure 2) is to integrate the results of step (3) and step (4), select the semantic labels for annotation, and obtain the result of automatic semantic annotation.

**4.2. Algorithms Description.** For the text information of IoT devices, while function description is commonly described by

unformatted texts, nonfunction description which includes the information about the performance, interface, and working condition of our device description framework in IoT generally has a particular format. Each step in Figure 2 applies different approaches to process data, as shown in Figure 3.

Figure 3 shows the process and the corresponding algorithms of automatic semantic annotation. The details of each algorithm are shown as follows.

(1) *Devices' Function Information Extraction.* For devices text information in IoT such as instructions, devices' function description is usually between pluralities of subtitles. For example, it may be between "Product Overview" subtitle and "Model Description" subtitle or between "Product Overview" subtitle and "Product Features" subtitle. This process consists of two phases: training phase and extraction phase. In the training phase, this process trains the classifier using subtitle training set and then learns a dictionary which contains words and corresponding word frequency appeared in the training set. In the extraction phase, a new sample is matched with trained dictionary and this process recognizes the subtitles appeared in the new sample. Then, this process extracts the content between adjacent recognized subtitles and the extracted content is reorganized into a document. This document is named function description in step (1) in Figure 3.

(2) *Devices' Function Classification.* Devices' function description is unformatted and disorganized text. There are three types of IoT devices: sensor devices, processor devices, and actuator devices. Different categories of devices have different input and output. For sensor devices, such as a humidity sensor, the input is stimulation and the output



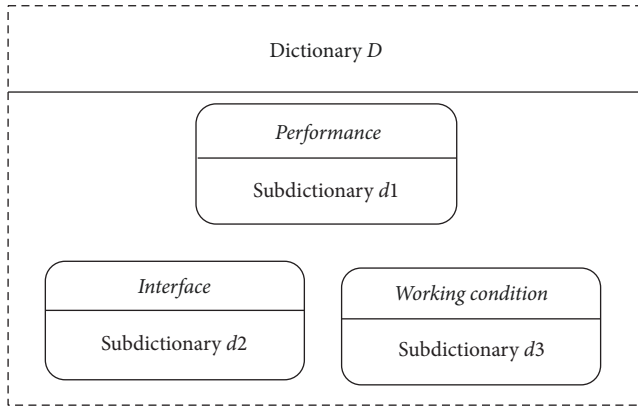


FIGURE 4: The structure of annotation dictionary.

is data. For processor devices, the input and output are both data. For actuator devices, the input is data and the output is action. Different categories of devices have different functions. Many text classification algorithms can be applied in devices' function classification, such as SVM [1], Naïve Bayes [2], Decision Tree [2], Artificial Neural Networks [3], and KNN [4]. However, SVM has a high training time complexity. Decision Tree is actually a rule-based classifier with inadequate scalability and constructed tree is huge when the scale of text sets is large. Artificial Neural Networks require multiple iterations and have heavy computing burden. KNN needs to compare all texts in the training set when determining the category of a new sample text and the result of classification is especially susceptible by unbalanced sample data. Thus, in this paper, we select a relatively simple and effective Naïve Bayes algorithm for experiments. First of all, a text classification training set should be constructed manually and the devices' function description of which is manually annotated their category. Then, the training set is applied to train Naïve Bayes text classifier. Finally, a new sample can apply the trained classifier to determine its category.

(3) *Annotation Dictionary Generating and Matching Algorithm.* In our device description framework in IoT, the identification of devices is obtained when accessed to IoT. Relating to dynamic characteristics, the state of devices is generated from hardware devices which monitor those devices in real-time. Thus, nonfunction description only contains three components: performance, interface, and working condition. Nonfunction description is a text, the format of which has been processed in step (1) in Figure 2. Each row of the text represents a message. Therefore, the problem of property information division can be considered as a classification problem that is to classify the message of each row in nonfunction description. Annotation dictionary generating and matching algorithms are proposed to address this classification problem and include two phases: annotation dictionary training phase and classification phase. The structure of annotation dictionary is shown in Figure 4.

Annotation dictionary contains three subdictionaries corresponding to the performance, interface, and working condition in our device description framework. The word frequency dictionary  $TF$  has the same structure as the annotation dictionary  $D$  and the two dictionaries are corresponding to each other. In the phase of dictionary training, the content of each property in training set is segmented to a sequence of words that are added to  $D$  and  $TF$ . The specific process of annotation dictionary training phase is given in Algorithm 1.

In Algorithm 1, the input is a training set  $Nf$  that has fixed format, and the outputs are the annotation dictionary  $D$  and the word frequency dictionary  $TF$ . Each component of  $Nf$  is segmented into a sequence of words that are added to  $D$ . Meanwhile, the word frequency of each word is gathered statistically and added to  $TF$  in Step 1. All results are combined in Step 2. Given the average word number of  $Ws_i$   $n$  and the scale of  $D$   $m$ , the time and space complexity of Algorithm 1 are  $O(nm)$ .

In the phase of annotation dictionary classification, this algorithm divides the nonfunction description into multiple components. The main idea of this algorithm is to segment the nonfunction description into a sequence of words marked as  $W$ . Then this algorithm matches each word in  $W$  with an annotation dictionary and a word frequency dictionary. The nonfunction description is divided according to the matching results. In particular, if there are multiply results that match success, the result with maximum word frequency will be the most appropriate. The detailed process of annotation dictionary matching algorithm is shown in Algorithm 2.

In Algorithm 2, the inputs are an annotation dictionary  $D$  generated in Algorithm 1, a word frequency dictionary  $TF$  generated in Algorithm 1 and a sample text  $Nnf$ . The output is a property division result that has the same structure as a text in training set  $Nf$  (as shown in Algorithm 1).  $Nnf$  is segmented and this algorithm obtains a word sequence  $Nw$  in Step 1. Each word in  $Nw$  is matched with  $D$  and  $TF$  and a matching result  $L$  is obtained in Step 2.  $Nw$  is divided according to  $L$  in Step 3. Let  $p$  denote the average word number of  $Nnf$  and  $m$  denote the scale of  $D$ ; the time and space complexity of Algorithm 2 are  $O(pm)$ .

(4) *Ontology Concept Matching Based on Semantic Similarity.* The processes of information integration and semantic label selection include information integration phase and semantic label selection phase. The classification results of function description and the property division results of nonfunction description are combined in information integration phase. In semantic label selection phase, each piece of key information has a label that has no semantic meaning. Taking the information of devices as the example, "operating temperature: 20~30°C," the label of "20~30°C" is "operating temperature" but this label has no semantic meaning. Thus, semantic label selection achieves the mapping between nonsemantic labels and semantic labels. In order to enable machine to understand labels, ontology is introduced to our approach and semantic similarity is applied to measure the similarity degree between two words or two phrases.

**Input:**

Non-function description training set: The format of each element in training set is  $Nf(\text{Pref}, \text{Inter}, \text{WorkCond})$  and  $Nf$  contains three components, i.e., Pref, Inter and WorkCond, respectively meaning the content about the performance, interface and working condition of our device description framework.

**Output:**

A dictionary  $D$  that contains three sub-dictionaries  $d_i (i \in \{1, 2, 3\})$  as shown in Figure 4.

A word frequency dictionary  $TF(tf_i, i \in \{1, 2, 3\})$  that has the same structure as  $D$ .

*Step 1.* For each component  $Nf_i$  that can be Pref, Inter and WorkCond in  $Nf$ :

segment word  $Nf_i$  and obtain a word sequence  $Ws_i$ .

For each  $Ws_{i,j}$  in  $Ws_i$ :

**If**  $Ws_{i,j}$  is not in  $d_i$ , add  $Ws_{i,j}$  to  $d_i$  and add 1 to  $tf_i$

**Else** find the position of  $Ws_{i,j}$  in  $d_i$ , marked as  $p$ . Then, set  $TF[p] = TF[p] + 1$ .

*Step 2.* Obtain a dictionary  $D$  and  $TF$

**Return:**  $D$  and  $TF$

ALGORITHM 1: Annotation dictionary generating algorithm.

**Input:**

An annotation dictionary  $D$ , a word frequency dictionary  $TF$  and a new non-function description  $Nnf$ .

**Output:**

A property division result  $NnfR$ , which contains three components, i.e., Pref, Inter and WorkCond. Those three components are the contents about the performance, interface and working condition of our device description framework.

*Step 1.* Obtain a word sequence  $Nw$  after segment  $Nnf$ .

*Step 2.* For each  $Nw_i$  in  $Nw$ :

**If**  $Nw_i$  in  $d_j$ , the category that  $Nw_i$  belongs to  $l_i = j$ .

(i) find the position of  $Nw_i$  in  $d_j$  and  $tf_j$ , marked as  $p_i$  and  $f_j$ .

(ii) **IF**  $j$  has more than one, choose a  $j$  which can maximize  $f_j$ .

**Else**  $l_i = 0$ .

**Then** obtain a position sequence  $L: (l_1, l_2, \dots, l_q)$ .

*Step 3.* For each  $Nw_i$  in  $Nw$ :

(i) **If**  $l_i = 0$ , **If**  $i \neq 1$ , add  $Nw_i$  to the component of  $NnfR$  that  $l_{i-1}$  belongs to.

(ii) **If**  $l_i = 1$ , add  $Nw_i$  to  $NnfR.pref$ .

(iii) **If**  $l_i = 2$ , add  $Nw_i$  to  $NnfR.inter$ .

(iv) **If**  $l_i = 3$ , add  $Nw_i$  to  $NnfR.workCond$ .

**Return:**  $NnfR$

ALGORITHM 2: Annotation dictionary matching algorithm.

The main process of semantic label selection for a nonsemantic label is to compute the semantic similarity between nonsemantic labels with all concepts in the device ontology and to find an ontology concept that can maximize the semantic similarity. If the semantic similarity is greater than a certain threshold, the selected concept's URI that is the semantic label will be returned; otherwise, null value will be returned. The specific process of ontology concept matching based on semantic similarity is shown in Algorithm 3.

The inputs of the proposed algorithm are device ontology  $D$ , a threshold  $\delta$ , a word, or a phrase  $W$  and the component  $C$  which  $W$  belongs to in our device description framework.  $C$  can be "Identification," "Performance," "Interface," and so on. The output of Algorithm 3 is the URI of a concept in  $D$ . The concept  $C_c$  which is related to  $C$  and all concepts  $S$  linked with  $C_c$  are found in Step 1. In Step 2, two parameters are set.  $MaxSimilarity$  means the maximum value in  $S$  and  $MS$  represents the index of  $MaxSimilarity$ . In Step 3, each element

$S_i$  in  $S$  is computed semantic similarity with  $W$ , and the URI of a concept in  $D$  that can maximize the semantic similarity is returned in Step 4. Assuming that the average number of  $S$  is  $q$  and the scale of ontology  $D$  is  $r$ , the time and space complexity of Algorithm 3 are  $O(qr)$ .

The text classification results of function description, the property division results of nonfunction description, and the selected semantic labels are reorganized to the final results of automatic semantic annotation.

**4.3. Algorithms Improvement.** Those algorithms above can substantially complete the process of automatic semantic annotation of IoT devices. Moreover, a device ontology expansion algorithm and an annotation dictionary expansion method are proposed to take consideration of the scalability of our approach.

**4.3.1. Device Ontology Expansion Algorithm Based on Semantic Similarity.** The prerequisite of Algorithm 3 is a given

**Input:**  
 A word or a phrase  $W$  and the component  $C$  which  $W$  belongs to in our device description framework.  
 A device ontology  $D$   
 A contain threshold  $\delta$

**Output:**  
 The URI of an ontology concept in ontology  $D$

*Step 1.* Find the concept  $C_c$  which is related to  $C$  in ontology  $D$  and obtain all ontology concepts which are linked with  $C_c$  in  $D$ , marked as  $S: (S_1, S_2, \dots, S_n)$ .

*Step 2.* Assuming that  $MaxSimilarity = 0, MS = 0$ .

*Step 3.* For each  $S_i$  in  $S$ :

- (i) For  $S_i$ , obtain  $sn_i$  after extract concept's name.
- (ii) compute the semantic similarity between  $W$  and  $sn_i$ , obtain  $Similarity$ .
- (iii) **If**  $MaxSimilarity < Similarity$ , set  $MaxSimilarity = Similarity, MS = i$ .

*Step 4.* **If**  $MaxSimilarity < \delta$ , set  $S_{MS} = null$

**Return:**  $S_{MS}$

ALGORITHM 3: Ontology concept matching based on semantic similarity.

**Input:**  
 A device ontology  $Device$ .  
 A contain threshold  $\delta$ .  
 A sub-tree expected to be expanded  $ST: (P, S, V)$ .

**Output:**  
 An extended ontology  $Device$ .

*Step 1.* For each ontology concept  $C_i$  in  $Device$ :

- (i) compute the semantic similarity between  $C_i$  and  $ST$  which is the top concept of  $ST: (P, S, V)$ , obtain  $S_i$ .
- (ii) find the maximum in  $S: (S_1, S_2, \dots, S_n)$ , obtain  $S_m$  and the corresponding ontology concept  $C_m$ .

*Step 2.* **If**  $S_m > \delta$ , add  $ST$ 's child concepts  $P, S$  and  $V$  as the child of  $C_m$ , as shown in Figure 7(a).

**Else If:**

- (i) assuming that  $Tmp = ST$ , set  $ST = P$  or  $ST = S$  or  $ST = V$ , and return to Step 1.
- (ii) **If**  $S_m > \delta$ , let  $Tmp$  becomes a child concept of  $Device$  and adds a link named "TogetherHas" between  $C_m$  and  $Tmp$ .  
 The link means  $C_m$  and  $Tmp$  has a same child concept, as shown in Figure 7(b).

**Else** let  $ST$  becomes a child concept of  $Device$ , as shown in Figure 7(c).

**Return:**  $Device$

ALGORITHM 4: Device ontology expansion algorithm based on semantic similarity.

device ontology. However, there is no related and useable ontology in IoT recently. For example, there is a task to find a suitable concept in the device ontology for "operating temperature," and the result may be "humidity" if there is no suitable concept in ontology. Treating "humidity" concept as the semantic label of "operating temperature" is obviously wrong. Thus, in order to obtain correct semantic labels, "operating temperature" should be expanded into the device ontology as an ontology concept. In this paper, we propose a device ontology expansion algorithm based on semantic similarity. The main idea of this algorithm is to initialize small device ontology and to add a subtree (as shown in Figure 5) to the device ontology.

Nonfunction description contains three components: performance, interface, and working condition. The content of each component can be obtained by Algorithm 2. For example, the "working condition" concept may contain many subconcepts, such as ambient temperature, humidity, and altitude. An example of creating a subtree is shown as follows.

- (1) The root of subtree is the "working condition" concept.
- (2) The children of the root are the content of "working condition," such as ambient temperature, humidity, and altitude. They are the subconcepts of the root and the structure of a created subtree is shown in Figure 6.

The structure shown in Figures 5 and 6 can be represented by  $C: (P, S, V)$ , where  $C$  is the top concept of this structure and  $P, S$ , and  $V$  are the subconcepts of  $C$ . The specific algorithm is shown in Algorithm 4.

In Algorithm 4, the inputs are a device ontology  $Device$ , a subtree  $ST$ , and a threshold  $\delta$ . The output is the ontology  $Device$  after extension. In Step 1, semantic similarity between the top concept  $C$  in  $ST$  and each concept in  $Device$  is computed and is marked with  $S$ . The maximum  $S_m$  in  $S$  and the corresponding ontology concept  $C_m$  are found. In Step 2, if  $S_m > \delta$ , this algorithm adds the subconcepts of  $C$  under the concept  $C_m$  (as shown in Figure 7(a)). Otherwise, similar to the process in Step 1, a matching process of subconcept



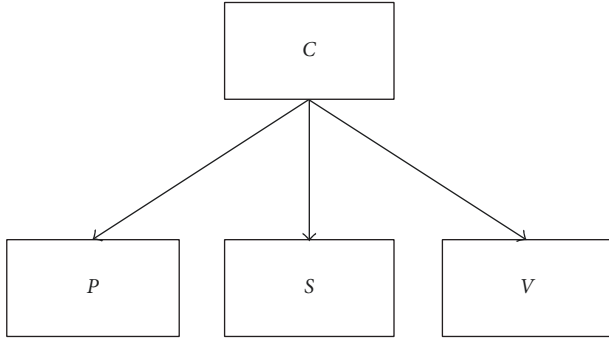


FIGURE 5: The structure of the subtree.

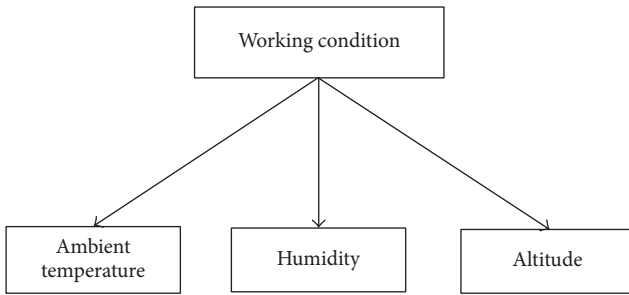


FIGURE 6: An example of the subtree in Figure 5.

(including  $P$ ,  $S$  and  $V$ ) of  $C$  is started. This algorithm supposes  $P$  match success and then links  $C_m$  and  $P$  with the “TogetherHasP” relationship (as shown in Figure 7(b)). If all concepts (including  $C$ ,  $D$ ,  $P$ , and  $V$ ) fail to match, this algorithm adds  $C$  and the subconcept of  $C$  under the top concept of  $Device$  (as shown in Figure 7(c)). Let  $r$  denote the scale of ontology  $D$ , and the time and space complexity of Algorithm 4 are  $O(r)$ .

**4.3.2. Annotation Dictionary Learning Based on Semantic Similarity.** The annotation dictionary is associated directly with the classification of nonfunction description and plays a leading role in semantic annotation in IoT. When a new sample contains some new words that are not included in the annotation dictionary, the results of semantic annotation are incorrectly using the original annotation dictionary. For example, if a new sample contains a “frequency” word which is not included in the annotation dictionary, the classification result of the “frequency” word often has a strong possibility of error. The solution is to expand the annotation dictionary before classifying. The process of this phase is similar to Algorithm 1 except the sources of the training set. The training set of this process can be obtained by Algorithm 2 or built by users.

## 5. Experiments

**5.1. Setup of Experiments.** We used three experiments to demonstrate the effectiveness of the proposed approach in this paper. The first experiment is to illustrate and analyze the annotation results of our approach. The second experiment is

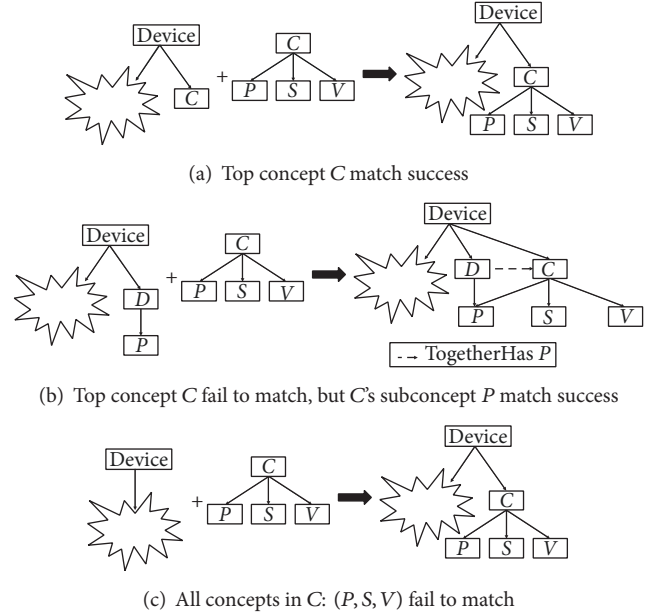


FIGURE 7

applied to indicate the influence of the experiment parameters on the annotation results of our approach. In the third experiment, we supplied a comparative experiment to evaluate our approach. IoT devices include temperature sensors, pressure sensors, RFID intelligent devices, transmitters, and current transformers. The data in this paper are the specifications of IoT devices. The experiments data contain different types of temperature sensors, pressure sensors, zero sequence current transformers, infrared gas sensors, gas measuring equipment, temperature transmitters, humidity transmitters, and so on. They are from different companies with a total 88 specifications of IoT devices. Using cross validation in the experiments, 88 datasets are divided into 8 groups and each group contains 11 datasets. Eight experiments are designed to evaluate the annotation effect of our approach and each experiment selects 7 groups of datasets as the training set while selecting 1 group of datasets as the test set. In the experiments, the text classification algorithm in this paper is Naïve Bayes algorithm and the experiment parameter  $\delta$  is assigned to 0.5.

**5.2. Experiments Evaluation.** The description of automatic semantic annotation results is shown as follows: the format of each annotation result is “<label>content</label>.” The component <label> is semantic label and its content is the URI of a concept matching from the device ontology using the method shown in step (4) in Section 4.2. For example, the content of component <label> can be “http://com.scut/owl/Ontology/#Voltage.” The content component is the key information extracted in step (1) in Section 4.2, for example, “0.38~66 KV.” The component </label> represents the end of an annotated result and its content is the same as the component <label>. An automatic semantic annotation result of our method is showed in Box 1.

```

<http://com.scut.emos/owl/Ontology/Device/#Identification>B:002</http://com.scut.emos/owl/Ontology/Device/
#Identification>
<http://com.scut.emos/owl/Ontology/Device/#Performance>
  <http://com.scut.emos/owl/Ontology/Device/#Voltage>0.38 KV~66 KV
</http://com.scut.emos/owl/Ontology/Device/#Voltage>
  <http://com.scut.emos/owl/Ontology/Device/#GridFrequency>50 Hz
</http://com.scut.emos/owl/Ontology/Device/#GridFrequency>
  <http://com.scut.emos/owl/Ontology/Device/#Start>“L1” side the second as “K1”
</http://com.scut.emos/owl/Ontology/Device/#Start>
</http://com.scut.emos/owl/Ontology/Device/#Performance>
<http://com.scut.emos/owl/Ontology/Device/#Function>
  <http://com.scut.emos/owl/Ontology/Device/#FunInput>data</http://com.scut.emos/owl/Ontology/Device/
#FunInput>
  <http://com.scut.emos/owl/Ontology/Device/#FunProfile>handling
device</http://com.scut.emos/owl/Ontology/Device/#FunProfile>
  <http://com.scut.emos/owl/Ontology/Device/#FunOutput>data</http://com.scut.emos/owl/Ontology/Device/
#FunOutput>
  <http://com.scut.emos/owl/Ontology/Device/#FunType>http://com.scut.emos/owl/Ontology/Device/#Zero
</http://com.scut.emos/owl/Ontology/Device/#FunType>
</http://com.scut.emos/owl/Ontology/Device/#Function>
<http://com.scut.emos/owl/Ontology/Device/#State>NULL</http://com.scut.emos/owl/Ontology/Device/#State>
<http://com.scut.emos/owl/Ontology/Device/#Interface>NULL</http://com.scut.emos/owl/Ontology/Device/#Interface>
<http://com.scut.emos/owl/Ontology/Device/#WorkingCondition>
  <http://com.scut.emos/owl/Ontology/Device/#AmbientTemperature> -10
</http://com.scut.emos/owl/Ontology/Device/#AmbientTemperature>
  <http://com.scut.emos/owl/Ontology/Device/#AtmosphericPressure>80~110 KPa
</http://com.scut.emos/owl/Ontology/Device/#AtmosphericPressure>
  <http://com.scut.emos/owl/Ontology/Device/#RelativeHumidity>90% (25°C) 50% (40°C)
</http://com.scut.emos/owl/Ontology/Device/#RelativeHumidity>
</http://com.scut.emos/owl/Ontology/Device/#WorkingCondition>

```

Box 1: An automatic semantic annotation result of our approach.

The contents of five properties, which are the identification, performance, function, interface, and working condition of our device description framework in IoT, are displayed in Box 1 and each property corresponding to a URI (e.g., <http://com.scut.owl/Ontology/#Performance>). The content of each property is embedded between `<label>` and `</label>`.

The goal of semantic annotation in IoT is to annotate IoT devices with semantic labels and further transform the results of semantic annotation into semantic IoT services. In this way, IoT devices can be depicted by the unified and rich semantic form and support semantic service discovery. Ontology technology is the crucial elements of semantic IoT services. The results of automatic semantic annotation can be directly transformed into ontology individuals. An annotation result of our method represented by N3 notation (<https://www.w3.org/TeamSubmission/n3/>) is shown in Box 2.

For the convenience of illustration, an ontology individual represented by N3 notation is shown in Box 2. It is named “B:002” and consists of four parts segmented by a blank line. In the first part, the first line is applied to specify that the namespace of “device” is “<http://com.scut.emos/owl/Ontology/Device/#>” and the third line is applied to indicate that “B:002” is an individual of

“Device” ontology. The next few lines are applied to illustrate the relationships the “B:002” rule has. For example, the fourth line indicates that the “B:002” rule owns the “device:hasPerformance” relationship that points to the “device:PerformanceB002” concept. The second part is applied to describe the “device:PerformanceB002” concept which has the “device:hasVoltage” relationship and the “device:hasGridFrequency” relationship. The “device:hasVoltage” relationship points to “0.38 KV~66 KV”, which means that “B:002” has a “Voltage” attribute whose value is “0.38 KV~66 KV”. While the third part is applied to describe the “device: FunctionB002” concept, the fourth part is applied to indicate the “device: WorkingConditionB002” concept.

Two evaluation indexes, precision and recall, are applied to evaluate the annotation ability of our approach. To demonstrate the effectiveness of our approach, the results of automatic semantic annotation, marked as AR, are compared with the results of manual semantic annotation, marked as MR. For each message of IoT devices’ information, such as “the voltage is 0.38–66 KV,” the format of each annotated message is “`<label>content</label>`,” which contains two components: content and label. An annotated message is correct if and only if content and label are both correct. The calculation formulas are as follows:  $P1 = A/E$ ,  $P2 = B/F$ ,

@prefix device: <http://com.scut.emos/owl/Ontology/Device/#>	
device: B:002	
a	device:Device
device:hasPerformance	device:PerformanceB002
device:hasFunction	device:FunctionB002
device:hasState	NULL
device:hasInterface	NULL
Device:hasWorkingCondition	device:WorkingConditionB002
device:PerformanceB002	
device:hasVoltage	“0.38 KV~66 KV”
device:hasGridFrequency	“50 Hz”
device:FunctionB002	
device:hasFunInput	“data”
device:hasFunProfile	“handling device”
device:hasFunOutput	“data”
device:hasFunType	NULL
device:WorkingConditionB002	
device:hasAmbientTemperature	“-10”
device:hasAtmosphericPressure	“80~110 KPa”
device:hasRelativeHumidity	“90% (25°C) 50% (40°C)”

Box 2: A result of semantic annotation represented by N3 notation.

TABLE 2: The average of precision and recall in each experiment.

Experiment	1	2	3	4	5	6	7	8
P1	0.848	0.902	0.867	0.889	0.853	0.854	0.879	0.883
P2	0.874	0.912	0.879	0.901	0.868	0.870	0.863	0.908
C1	0.906	0.910	0.891	0.922	0.896	0.903	0.883	0.711
C2	0.901	0.887	0.886	0.917	0.896	0.884	0.898	0.927

$C1 = A/C$ , and  $C2 = B/D$ , where  $P1$  and  $P2$ , respectively, represent the precision of content and label components in AR,  $C1$ , and  $C2$ , respectively, mean the recall of content and label components in AR. The quantity of correct content component and correct label component in AR is, respectively, denoted as  $A$  and  $B$ , and  $E$  and  $F$ , respectively, represent the total amount of content and label components in AR, while  $C$  and  $D$ , respectively, mean the total number of content and label components in BR.

Each device specification corresponds to a four-tuple ( $P1$ ,  $P2$ ,  $C1$ , and  $C2$ ), and the average of four indexes in each experiment is calculated. The results are shown in Table 2.

The combined precision  $P$  and recall  $C$  are computed according to Table 1 by the calculating formulas

$$P = \alpha P1 + (1 - \alpha) P2 \quad (1)$$

$$C = \beta C1 + (1 - \beta) C2,$$

where  $\alpha$  and  $\beta$  are weight and can be set according to users' specific requirements. In this paper, we set  $\alpha = 0.5$  and  $\beta = 0.5$ . The combined results are shown in Table 3.

The precision and recall of  $i$ th group of datasets are marked as  $P_i$  and  $C_i$ , respectively. The average precision  $P_z$  and the average recall  $C_z$  of our approach are calculated by computing arithmetic average according to the combined

TABLE 3: The combined precision and recall of each experiment.

Experiment	1	2	3	4	5	6	7	8
$P$	0.861	0.907	0.873	0.896	0.860	0.862	0.840	0.900
$C$	0.903	0.898	0.889	0.920	0.896	0.894	0.891	0.919

TABLE 4: The average precision  $P_z$  and recall  $C_z$ .

Index	Value
$P_z$	0.8743
$C_z$	0.9012

precise and recall in Table 3. The calculating formula is shown as follows:

$$P_z = \sum_{i=1}^N \frac{P_i}{N} \quad (2)$$

$$C_z = \sum_{i=1}^N \frac{C_i}{N},$$

where  $N$  is the number of the groups of cross validation experiments. In this experiment,  $N$  is set 8. The computing results are given in Table 4.

TABLE 5: The results of experiment parameters analysis.

Index	$\delta$										
	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99
$P_1$	0.848	0.867	0.852	0.860	0.850	0.864	0.863	0.857	0.858	0.873	0.873
$P_2$	0.867	0.885	0.872	0.878	0.871	0.884	0.885	0.877	0.880	0.895	0.895
$C_1$	0.915	0.897	0.904	0.908	0.900	0.903	0.899	0.898	0.890	0.894	0.890
$C_2$	0.899	0.894	0.899	0.904	0.897	0.899	0.900	0.893	0.888	0.892	0.890
$P_z$	0.858	0.876	0.862	0.869	0.860	0.874	0.876	0.867	0.870	0.884	0.884
$C_z$	0.907	0.896	0.901	0.906	0.898	0.901	0.899	0.895	0.889	0.893	0.890
$F$ -measure	0.882	0.885	0.881	0.887	0.879	0.888	0.886	0.880	0.879	0.888	0.886

Table 4 shows that the average precision and recall of our approach are 87.43% and 90.12%,  $F$ -measure that combines precision and recall is defined as

$$F = \frac{2PC}{P+C}. \quad (3)$$

Actually,  $F$ -measure is the geometric average of precision and recall. The larger the  $F$ -measures are, the better the results of semantic annotation are. The  $F$ -measure of our approach is 0.8876, which means that our approach can correctly annotate 88.76% of IoT devices' information. This experiment demonstrates that our approach has great precision, recall, and  $F$ -measure. It also proves that our approach is an efficient and effective method for semantic annotation of IoT devices.

**5.3. Analysis of Experiment Parameters.** In this paper, Algorithms 3 and 4 are related to semantic similarity which contains a threshold  $\delta$ . In Algorithm 3, the parameter  $\delta$  is applied to select semantic labels from the device ontology. It is easy to get an error and meaningless semantic label (this wrong information may be rather trouble in service discovery than null value) when  $\delta$  is set too low. Few appropriate semantic labels are found when  $\delta$  is set too high. In Algorithm 4,  $\delta$  is applied to ontology concept matching. Unrelated concepts are easy to be matched successfully when  $\delta$  is set ridiculously low, while related concepts are matched unsuccessfully when  $\delta$  is set ridiculously high. Thus, it is extremely important to set an appropriate value of the parameter  $\delta$ .

In this section, we carry out an experiment to analyze the influence of the parameter  $\delta$  on semantic annotation results. The parameter  $\delta$  has been set from 0.01 to 0.99. After cross validation and the evaluation of semantic annotation results using the indexes provided in Section 5.2, we obtain the experiment results as shown in Table 5.

Table 5 displays that the influence of different values of parameter  $\delta$  on the results is not serious, and the fluctuation range of the results is in the range of 10%. The  $F$ -measure of our approach floats around 0.885. There are two reasons that cause those situations. Firstly, the device ontology that is applied to semantic label selection is large enough after training and expansion, so that most of words or phrases can accurately choose semantic labels with a high semantic similarity that near 1.0. Thus, difference values of parameter  $\delta$  cannot obviously affect the semantic annotation results.

Secondly, the process of semantic label selection and ontology concept matching is to select ontology concepts that have maximum semantic similarity with corresponding words or phrases. Those weaken the influence of parameter  $\delta$  on the results to an extent.

**5.4. Method Comparison.** In this section, our experimental evaluation aims to show the performance of our approach. The evaluation is achieved by comparing our method with General Architecture of Text Engineering (GATE) framework. GATE is open source software that has ability of solving almost text processing problems, including semantic annotation and information extraction named entity recognition. A Nearly-New IE System (ANNIE) (<https://gate.ac.uk/sale/tao/splitch6.html#x9-1200006>) which has processing resources of sentence splitter, POS Tagger, and JAPE transducer is an information extraction system in GATE. JAPE (<https://gate.ac.uk/sale/tao/splitch8.html#x12-2070008>) is a language to define rules for information extraction and allows users to recognize regular expressions in annotation on text. GATE provides a rule-based automatic semantic annotation method and will extract the relevant information according to the extraction rules defined by users. Those extraction rules are described by JAPE.

The experiment was conducted as follows. Firstly, a lot of necessary extraction rules are described by JAPE to define the information that expects to be extracted from devices' information. Secondly, all JAPE documents defined by users are added to GATE for information extraction. Besides, ontology concepts of the device ontology are selected to annotate the results of information extraction. Then, we obtain the results of automatic semantic annotation using GATE. Finally, all the two approaches are competitive in aspects of precision, recall, and  $F$ -measure. The results returned in this comparative experiment are achieved and shown in Figure 8.

As illustrated in Figure 8, both of two approaches are comparative in aspects of precision, recall, and  $F$ -measure. Our approach obviously performs better than GATE in terms of precision and  $F$ -measure. Nevertheless, GATE has a better performance with respect to recall. The average content recall  $C_1$  of GATE arrives beyond 92% and the average label recall  $C_2$  of GATE achieves even above 96%. The detailed causes of this result are as follows: (1) GATE is a semantic annotation method based on predefined rules and there are

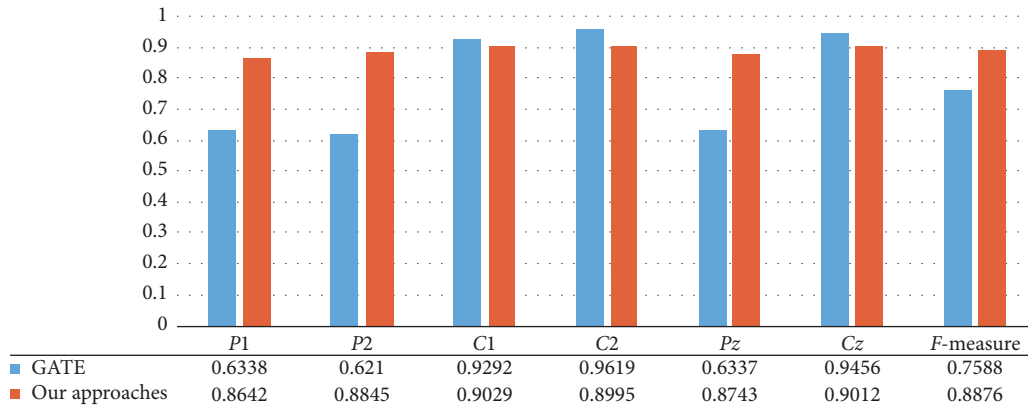


FIGURE 8: The performance of our approach and GATE.

some intercrossing relationships between rules. The error ratio of semantic annotation of GATE will extremely increase along with the growth of the rules and the intercrossing relationships among them. Moreover, the error ratio has a negative impact on the precision index. However, based on machine learning, our approach possesses excellent scalability and overcomes the limitations of rule-based methods. It is extremely robust with the increase of IoT devices. (2) As a rule-based semantic annotation method, the GATE can almost extract all the accurate information from IoT devices' information, so that GATE performs better in aspects of recall.

## 6. Conclusions

With the rapid growth in the number of IoT devices, manual and semiautomatic methods of semantic annotation can hardly meet the increasing requirements due to inefficiency. In this paper, we propose a device-oriented automatic semantic annotation method for information of IoT devices. The method can automatically extract key information, divide information, expand the device ontology, and match concepts in the device ontology. Although there are a number of semantic annotation methods, few of them focus on the information of IoT devices and deal with the automation of semantic annotation. The main contribution of our work consists of four parts: (1) considering the characteristics of IoT devices, we put forward a devices description framework to describe IoT devices; (2) we propose the process of automatic semantic annotation which consists of five steps; (3) we introduce a series of algorithms in the annotation process including annotation dictionary generating and matching algorithm and the algorithm for ontology concept matching; (4) taking the scalability into consideration, we propose an algorithm for device ontology extension based on semantic similarity to expand the device ontology and present an algorithm for annotation dictionary extension. The experiments show that our method for automatic semantic annotation is effective and outperforms the rule-based method, GATE. Although our method of automatic semantic annotation is also appropriate for general IoT entities and lays a foundation

for IoT service discovery, there is still no principled approach for automatic service encapsulation. In our future work, we will focus on the method of encapsulating the semantic annotated information of IoT devices into semantic IoT services for efficient service discovery.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This paper is supported by the Engineering and Technology Research Center of Guangdong Province for Logistics Supply Chain and Internet of Things (Project no. GDDST[2016]176); the 3th strategic rising industry program of Guangdong Province (Project no. 2012556003); International Cooperation Special Program for platform (Project no. 2012J510018); the Key Lab of Cloud Computing and Big Data in Guangzhou (Project no. SITGZ[2013]268-6); Engineering & Technology Research Center of Guangdong Province for Big Data Intelligent Processing (Project no. GDDST[2013]1513-1-11); IoT home wireless router system and RFID (Project no. GDEID2012IS054); the Promotion of the Industrialization of Family Information Platform (Project no. 2013B090200055).

## References

- [1] I. Peña-López, "ITU Internet report 2005: the internet of things," 2005.
- [2] M. Junling, J. Xueqin, and L. Hongqi, "Research on Semantic Architecture and Semantic Technology of IoT," *Research and Development*, vol. 8, no. 5, pp. 26–31, 2014.
- [3] Q. Xu, P. Ren, H. Song, and Q. Du, "Security enhancement for IoT communications exposed to eavesdroppers with uncertain locations," *IEEE Access*, vol. 4, pp. 2840–2853, 2016.
- [4] Z. Lv, T. Yin, X. Zhang, H. Song, and G. Chen, "Virtual reality smart city based on WebVRGIS," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1015–1024, 2016.



- [5] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the internet of things: early progress and back to the future," *International Journal on Semantic Web and Information Systems*, vol. 8, no. 1, pp. 1–21, 2012.
- [6] D. Rong, *The Research on Automatic Semantic Annotation Methods*, Lanzhou University of Technology, Lanzhou, China, 2012.
- [7] F. Chen, C. Lu, H. Wu, and M. Li, "A semantic similarity measure integrating multiple conceptual relationships for web service discovery," *Expert Systems with Applications*, vol. 67, pp. 19–31, 2017.
- [8] F.-G. Liu, C. Peng, and Y. Lin, "Design and implementation of semantic web service clustering algorithm," in *Proceedings of the 12th International Conference on Machine Learning and Cybernetics (ICMLC '13)*, pp. 1747–1751, Tianjin, China, July 2013.
- [9] S. Javanmardi, M. Shojafar, S. Shariatmadari, and S. S. Ahrabi, "FR trust: a fuzzy reputation-based model for trust management in semantic P2P grids," *International Journal of Grid and Utility Computing*, vol. 6, no. 1, pp. 57–66, 2015.
- [10] C. De Maio, G. Fenza, M. Gallo, V. Loia, and S. Senatore, "Formal and relational concept analysis for fuzzy-based automatic semantic annotation," *Applied Intelligence*, vol. 40, no. 1, pp. 154–177, 2014.
- [11] D. Evans, *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*, CISCO, San Jose, Calif, USA, 2011.
- [12] P. G. V. Naranjo, M. Shojafar, L. Vaca-Cardenas, C. Canali, R. Lancellotti, and E. Baccarelli, "Big data over SmartGrid-a fog computing perspective," in *Proceedings of the SOFTCOM Workshop*, pp. 1–6, November, 2016.
- [13] P. Barnaghi, W. Wang, L. Dong, and C. Wang, "A linked-data model for semantic sensor streams," in *Proceedings of the IEEE and Internet of Things (iThings/CPSCoM), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, Green Computing and Communications (GreenCom '13)*, pp. 468–475, Beijing, China, August 2013.
- [14] S. Kolozali, M. Bermudez-Edo, D. Puschmann, F. Ganz, and P. Barnaghi, "A knowledge-based approach for real-time IoT data stream annotation and processing," in *Proceedings of the 2014 IEEE International Conference on Internet of Things, iThings 2014, Collocated with 2014 IEEE International Conference on Cyber, Physical and Social Computing, CPSCoM 2014 and 2014 IEEE International Conference on Green Computing and Communications, GreenCom 2014*, pp. 215–222, twN, September 2014.
- [15] W. Wei and P. Barnaghi, "Semantic annotation and reasoning for sensor data," in *Smart Sensing and Context*, vol. 5741 of *Lecture Notes in Computer Science*, pp. 66–76, Springer, Berlin, Germany, 2009.
- [16] P. Chenyi, *Service-oriented entity semantic annotation in internet of things [M.S. thesis]*, South China University of Technology, Guangzhou, China, 2015.
- [17] J. Bing, *Research on semantic-based service architecture and key algorithms for the internet of things [Ph.D. thesis]*, Jilin University, Changchun, China, 2013.
- [18] Z. Ming, *Research on several key issues in internet of things applications [Ph.D. thesis]*, Beijing University of Posts and Telecommunications, Beijing, China, 2014.
- [19] E. Charton, M. Gagnon, and B. Ozell, "Automatic semantic web annotation of named entities," in *Advances in Artificial Intelligence*, vol. 6657 of *Lecture Notes in Comput. Sci.*, pp. 74–85, Springer, Berlin, Germany, 2011.
- [20] G. Diallo, M. Simonet, and A. Simonet, "An approach to automatic ontology-based annotation of biomedical texts," *Lecture Notes in Computer Science*, vol. 4031, pp. 1024–1033, 2006.
- [21] P. A. Kogut and W. S. Holmes III, "AeroDAML: applying information extraction to generate daml annotations from web pages," in *Proceedings of the 1st International Conference on Knowledge Capture (K-CAP '01)*, ACM Press, Victoria, Canada, 2001.
- [22] B. Kettler, J. Starz, W. Miller, and P. Haglich, "A template-based markup tool for semantic web content," *Lecture Notes in Computer Science*, vol. 3729, pp. 446–460, 2005.
- [23] M. Vargas-Vera, E. Motta, J. Domingue et al., "MnM: a tool for automatic support on semantic markup," KMi Technical Report, 2003.

