北京郵電大學

Queen Mary
University of London

Undergraduate Project Report

2016/17

A Semantic Annotation System for Web of Things

Name:	Du Sicong
Programme:	Telecoms
Class:	2013215108
QM Student No.	130801256
BUPT Student No.	2013213144

Date 2017/5/2

Table of Contents

Abstract

Integrating Web of Things (WoT) with semantic technologies facilitates the construction of a large-scale networked knowledge infrastructure which can provide many smart applications with a strong knowledge base. However, current WoT applications work with a manual mechanism to define the data model and device annotations. This leads to the emergence of many domain-specific applications which results in the problem that many knowledge resources are isolated from each other. This paper proposes and implements a semi-automatic semantic annotation system to link domain-specific entities to a domain independent knowledge base to improve the usability and the usefulness of WoT knowledge resources. Some use cases for tests are also considered.

Key words: Web of Things (WoT), semantic annotation, entity linking, message passing, jointly reasoning

摘要 (Chinese translation of the Abstract)

将语义技术与物联网相结合，有助于构建一个大型的网络知识基础设施，为许多智能应用提供强大的知识库。然而，当前的物联网应用程序使用手动机制来定义数据模型和设备标注。在这种情况下出现了许多特定领域的物联网应用，导致了许多知识资源彼此隔离。本文提出并实现了一个半自动语义标注系统，将特定领域的实体链接到一个与领域无关的知识库，以提高物联网知识资源的可用性和有用性。一些用于测试的用例也被考虑了进来。

Chapter 1: Introduction

1.1 Motivation

Internet of Things (IoT) connects many physical devices into the cyber world of World Wide Web through Uniform Resource Identifier (URI) as unique IDs for the devices. Web of Things (WoT) focus on reusing web patterns and protocols to enhance the concept of IoT. "Web Servient" is proposed to improve the abstract model of the devices. Each device is not only a client but also a server which facilitates the construction of cyber-physical systems with higher data and resource usability.

Under the spread of the concept of WoT, many WoT platforms emerge to provide developers with convenient data service and various solution templates. BaiduTianGong, Yeelink and Amazon web service are some of the available platform resources on the web. Developers open these platforms and invoke the APIs to define the business dependent device data model and web service interfaces to develop WoT applications. Obviously, the openness of the web technologies makes the development of IoT applications more easy. Devices in the real world can be more conveniently mashed up with both web services and other devices through cyber-physical systems. However, this results in many heterogeneous and isolated data and resources. Because the data models are defined by individuals without unified standard or linking method. Thus, this is a waste of usable information resources.

The data and services of WoT applications can be regarded as web resources. The introduction of semantic technologies aims to improve the usability and usefulness of WoT resources by bridging heterogeneous data from different domain dependent data sources with the help of global domain independent knowledge base. Accordingly, the semantic annotation technology facilitates the construction of a networked knowledge infrastructure with more interoperable structural data sources.

There are already some efforts to adopt semantic web technologies in WoT applications including Semantic Sensor Web, SENSEI and SPITFIRE. By annotating the physical devices with computer

comprehensible vocabulary such as Semantic Sensor Network (SSN) ontology model, it is possible to share data and knowledge with other applications and platforms. However, these efforts still have limitations. The computer understandable vocabularies such as RDF linked ontology are based on domain-specific knowledge. The semantics extraction from raw data needs domain experts to accomplish in manual way. This lowers the interoperability of data and knowledge sharing among different systems in different domains. They do not consider the possibility of a growing knowledge base linked to global domain independent Linked Open Data (LOD), for example, DBpedia, Yago, FOAF and etc. Since Key-Value pairs is a common form to annotate the attributes of the WoT devices, I take the format of JSON in an example to explain a possible concrete situation in which the data interoperability is lowered. In Figure 1, there are two schema from two different domains in JSON format. In schema 1, the developer annotates the type "location" with "location". However, in schema 2, another developer annotates the type "location" with "deploy". Computer does not know the two annotations refer to the same type. Thus, by linking domestic entities to the entities in domain independent knowledge base such as DBpedia, computer can understand the relations among different data sources among different domains which supports further semantic searching and integration.

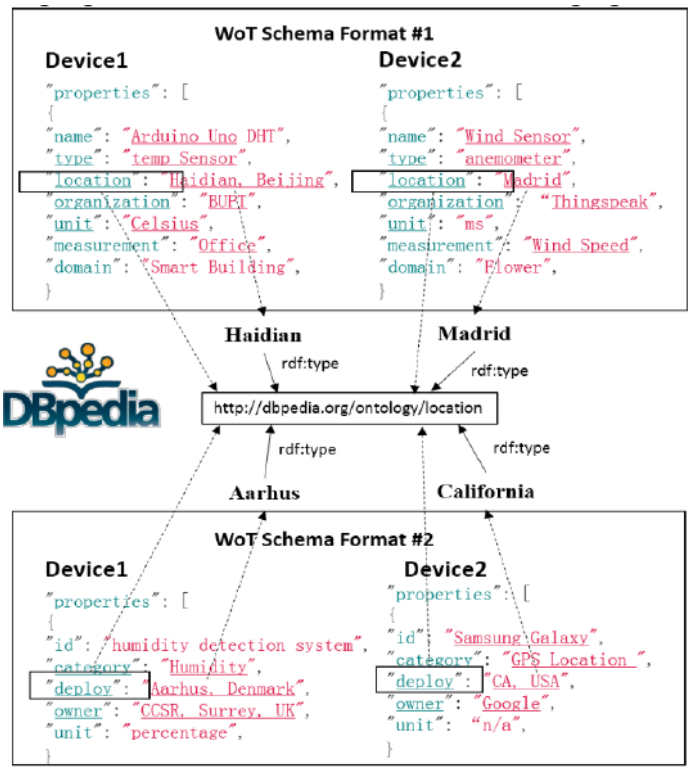


Figure 1 Linked WoT data with semantic annotations of keys, values and relations

1.2 Achievement

A manual annotation tool is implemented for user to manually construct the local knowledge graph. The construction of the local knowledge base is subject to a predefined ontology model. The output after manual annotation is a domain specific knowledge graph.

This paper proposes a factor graphic model to link the domain-specific WoT metadata to a domain independent knowledge base (KB). And an entity linking algorithm based on message passing is also proposed to disambiguate the domestic metadata. These techniques are used to implement the automatic annotation tool.

Some possible use cases for test and application are studied. These use cases are about the usage of the knowledge graph generated after manual and automatic annotation.

1.3 Technique context

The web application is implemented based on JSP techniques. The framework of Spring + Spring-MVC + Mybatis is adopted. Because these Java-based techniques have the following advantages:

•Java is an object oriented language. Object oriented is a natural abstraction of the real world entity model. Any entity in the physical world can be seen as an object. These objects can interact through messages. In addition, any entity in the physical world can be attached to a class of things. An object is an instance of a class. An object-oriented programming language is object-centred and message-driven. Object-oriented programming language can be regarded in a perspective that program is object plus message. Hence, it is easy to model the entities in the real world to the classes in programmes.

• Java is a platform-independent language. Applications written in Java can run on different hardware and software platforms without modification. Java mainly relies on the Java Virtual Machine (JVM) at the target code level to achieve platform independence. JVM is an abstract machine attached to the specific operating system. JVM has a set of virtual machine instructions with its own stacks and register groups. But the JVM is usually implemented on software rather than on hardware. Platform independence of Java has greatly accelerated and promoted the development of software products.

•Java supports multi-thread programming. Java supports multithreading in two ways. On the one hand, the Java environment itself is multithreaded. Several system threads are responsible for the useless cells recycling, system maintenance and other system-level operations. On the other hand, Java language built-in multi-threaded controlling can greatly simplify the development of multi-threaded applications. Java provides a class Thread, which is responsible for starting to run, terminating the thread, and checking the thread status. Java threads also include a set of synchronization primitives. These primitives are responsible for concurrency control of threads. Using Java's multi-thread programming interfaces, developers can easily develop the applications supporting multi-thread, improving program executing efficiency.

The storage methods in the implementation are as follows:

•Jena-tdb. This is a file storage system with the query language SPARQL. The input and the output of this module are both in OWL format. The data is RDF linked in Jena-tdb. Jena-tdb has built-in tools to support semantic reasoning.

•Mysql. Mysql is a relational database system based on table forms or schemas. We can conveniently apply the factor graphic model with the help of Mysql.

•Neo4j. Meo4j is a graph database which can visibly display the structure of the RDF linked data with labels and relations in a vivid graph form.

1.4 Report Structure

The rest of this paper is arranged in this way. Chapter 2 focuses on some background knowledge of semantic Web of Things. The detailed description of the annotation system and all the involved algorithms are illustrated in Chapter 3 as well as the details about the application implementation. Chapter 4 is about the experiment results and performance evaluation from various aspects of the annotation system. And Chapter 5 concludes the whole paper and predicts some possible subsequent future work after the efforts of this paper.

Chapter 2: Background

2.1 Web of Things (WoT)

The Web of Things (WoT) allows objects in the real world to connect to the cyber world, such as World Wide Web. It can be adopted to refer to relevant software architectural styles, programming patterns and approaches. Just as what Web (lies at Application layer) is to the Internet (lies at Network layer), the Web of Things serves as an Application Layer approach to simplify the development of Internet of Things (IoT) applications.

Instead of proposing a brand new standard, Web of Things reuses widely spread Web standards used in

- programmable Web, e.g. REST, HTTP, JSON
- semantic Web, e.g. JSON-LD, Microdata
- real-time Web, e.g. Websocket
- social Web, e.g. OAuth, social networks

2.2 Semantic web

The Semantic Web is an extension of the Web following the standards by the World Wide Web Consortium (W3C). It is an imagination towards the future network. Now, the semantic web behaves as one of the features of the age of Web 3.0. In short, semantic web is a type of intelligent web. It can interpret not only words and concepts, but also the logic relationship among them which can make communications more efficient and valuable.

The core of semantic web is to add machine comprehensible semantic metadata to the documents in the web such as HTML and XML. Thus, the whole Internet becomes a common medium for information exchange.

The evolution history of semantic web is depicted in Figure 2 with comparison to the timeline of XML world.

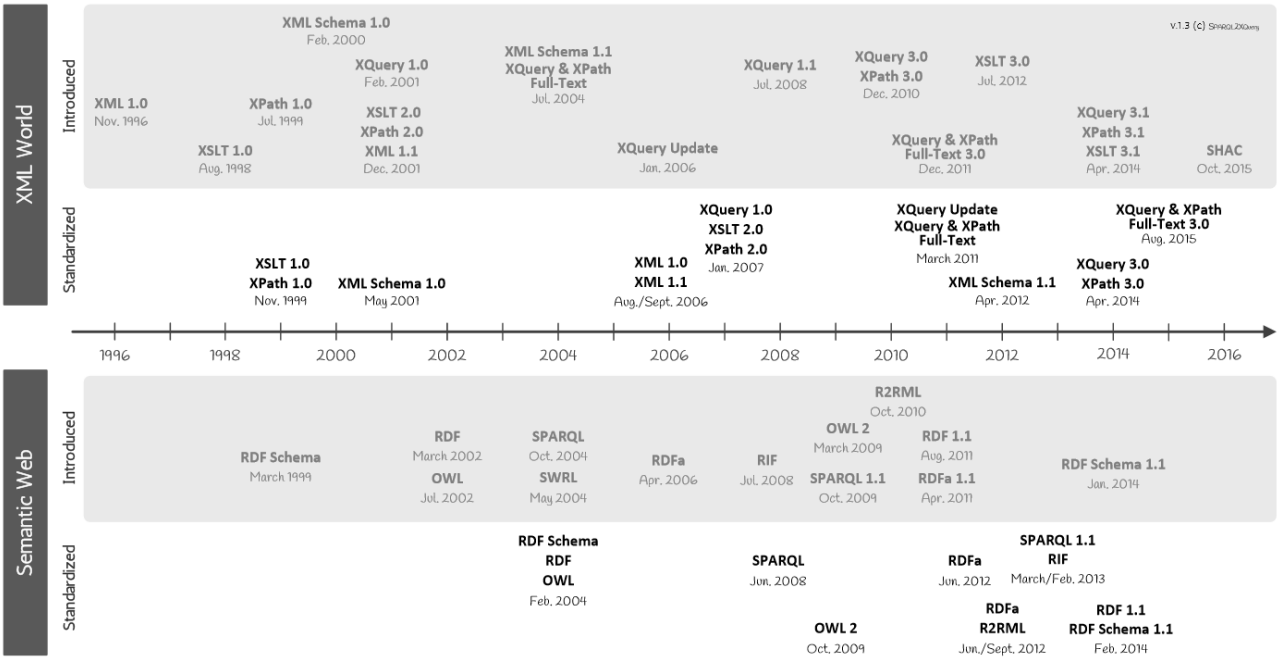


Figure 2 Evolution history of XML and semantic web standards[1]

2.2.1 Resource Description Framework (RDF)

RDF is a W3C standard used to describe cyber resources on the Web such as title, author, updated date, content and copyright of webpages. RDF is written based on XML techniques which is called RDF/XML. With the use of XML, RDF information can easily be exchanged among machines with different operating systems and application languages. And Resource Description Framework also provides models and grammars with respect to data. It serves as a general method to describe data so that it can be read and understood by computers. Thus, individual teams can use it and exchange information.

Accordingly, RDF is designed to be an important part of semantic web activities which facilitates the interactions among computers rather than presentations to persons in webpages. The purposes of Semantic Web Vision proposed by W3C are:

- Information on the Web has accurate meanings

- Web information can be understood and processed by computers
- Computers can integrate information from Web

Moreover, RDF is a directed, labeled graph data format for representing information in the web. It is often used to represent personal information, social networks, metadata about digital artifacts among other things, as well as to provide a method to integrate many different sources of information.

2.2.2 Web Ontology Language（OWL）

The term “ontology” comes from philosophy. The theory of ontology aims at studying various entities in the universe and how they are connected. When it comes to Web, ontology is an accurate description about web information and the relations among web information.

OWL is a kind of language used to process information on the web which is also a W3C standards. It is constructed on top of RDF using XML. The same as RDF, OWL is designed to be interpreted by computers rather than read by persons. And OWL is also an important component of “Semantic Web Vision”. OWL consists of sub-languages:

- OWL Lite
- OWL DL (Description Logic, including OWL Lite)
- OWL Full (including OWL DL)

OWL has many similar aspects with RDF. However, compared to RDF, OWL has a stronger ability of machine interpretation. It possesses larger vocabulary than RDF does.

2.2.3 Simple Protocol and RDF Query Language (SPARQL)

SPARQL is a data access protocol and a query language developed and defined for RDF data model.

It can be used for any information resources which can be represented by RDF. It is now a W3C recommended standard. As a protocol, SPARQL is a way of communication between parties that run SPARQL queries. It defines a way of invoking the service and bindings of a transport protocol for that goal. And it is also a standard query language and data access language in the form of expressive query against the RDF data model. Especially, SPARQL is compatible with the graph patterns when conducting queries.

To be short, SPARQL connects two web technologies together, Web 2.0 and semantic web. It is very promising to be the major network database query language and data access standard in the future.

2.3 Related work

This section is about some state of the art on using semantic technology to annotate sensory data and link entities for WoT applications. Some methods implement the annotation by linked data and ontological knowledge engineering and some others adopts machine learning methods to extract knowledge from existing data sources.

2.3.1 Semantic Sensor Web (SSW)[2]

The semantic sensor web (SSW) proposes a method to annotate sensory data using three dimensions which are spatial, temporal and thematic semantic metadata. This method reuses specifications proposed by Open Geospatial Consortium (OGC) and Sensor Web Enablement (SWE) and tries to extend the specifications using semantic technologies to construct an enhanced framework to describe and access the sensory data.

2.3.2 Linked sensor middleware (LSM) [3]

Linked sensor middleware (LSM) implements many functionality modules, for instances,

- wrappers to collect and publish real-time data
- a web interface for data annotation and visualization
- a Simple Protocol and RDF Query Language (SPARQL) for operation on entities and ontology

It facilitates the integration among different data sources. And with the help of LSM, the modularization of WoT applications can be improved in a great extent.

2.3.3 Knowledge Extraction and Knowledge Base Construction

Wang et al.[4] uses a novel pattern-based Hidden Markov model (pHMM) to model time series which tries to reveal a global picture of the system dynamics that generates the time series data. Ganz et al.[5] proposes an approach to automatically acquire the semantic knowledge from sensor data. They apply an enhanced k-means clustering method and a statistic model to extract and link relevant concepts in the form of a topical ontology. The City Pulse framework aims at pattern discovery in real-time cases and analysis involving prediction. It facilitates the construction of business services and use application in the background of smart city. It aggregates, interprets and extracts meaningful knowledge from large amount of heterogeneous data streams and provides real-time adaptive urban reasoning services.

Some other studies are not directly applied to IoT knowledge base construction. But, the methods they proposed can be used to semantic annotation and extraction on structural metadata of devices. For examples, Limaye et al.[6] and Mulwad et al.[7] both used probabilistic graphical model. And they both took a thought called message passing into consideration into their algorithm.

Chapter 3: Design and Implementation

3.1 System framework description

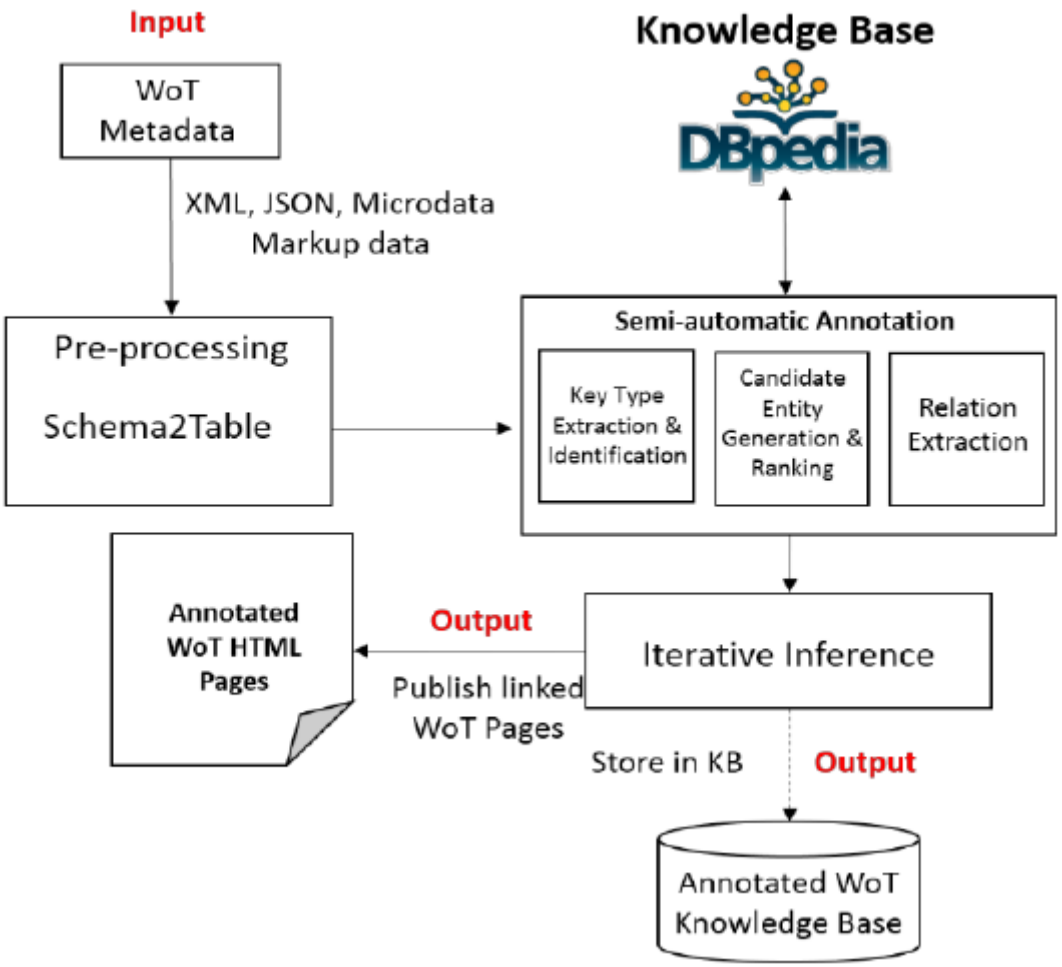


Figure 3 Annotation System on WoT metadata based on entity ranking and message passing

Figure 3 shows the overall structure of my annotation system based on entity linking algorithm and the massage passing model. The system consists of two main parts which are manual annotating and automatic annotating respectively. The manual parts aim at collecting WoT metadata from web front end with HTML and transfers them into tables and constructs the domestic domain-specific knowledge base. The automatic annotation part focus on linking the domain-specific knowledge base to the global domain independent knowledge base and providing a functionality of disambiguation based on entity linking algorithm with message passing model.

When the local knowledge graph is constructed according to the user inputs through HTML, the entity linking module begins to work. First the Candidate Generation module queries from DBpedia and gets a candidate set returned from DBpedia for each entity or cell value in the table. And then svm-ranking which is a third party supporting module will re-rank the candidate sets for each entity. The candidate sets for classes and relations between classes can be generated from the candidates of cell values and their corresponding types in DBpedia. After the candidate generation, the module of iterative jointly inference will extend the local knowledge graph through linking it to the global domain independent knowledge base. The extended knowledge graph is exactly the target of the whole annotation system.

3.2 Manual annotation tool

Manual annotation tool aims at transferring the user input into tables and RDF linked data. User input annotations for his devices through web front end (HTML). And the manual annotation system will store them into MySQL as tables and into Jena-tdb as RDF linked data with their official operation APIs.

The user input is restricted to a predefined ontology model. The classes in the top level are shown in **Figure 4** with the help of the software protégé. The ontology model is designed for the application scene of smart building. This ontology design takes the temperature anomaly diagnosis and automatic controlling into consideration which facilitates the future usage of the whole knowledge graph.

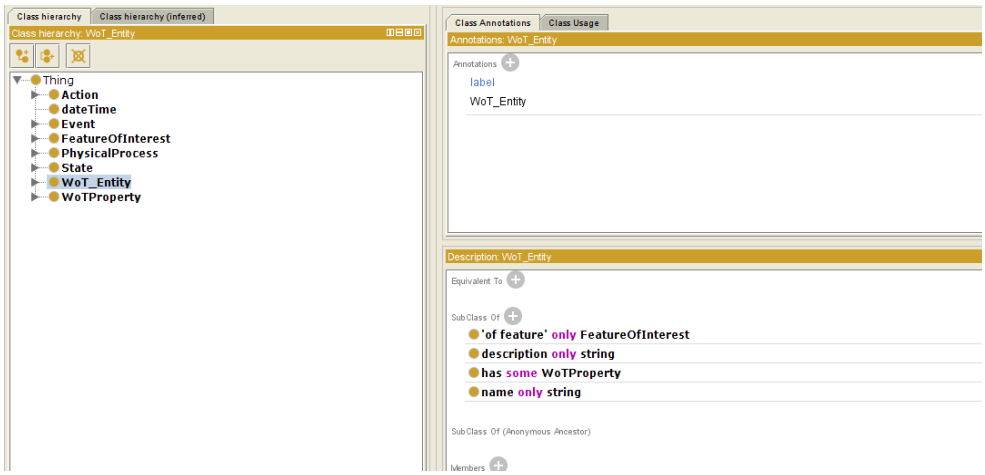


Figure 4 Top level classes in the predefined ontology

The class WoT_Entity is the most basic one. The devices added by users are the subclasses of this class. A device has a name and corresponding origin description made by the user. WoT_Entity has a series of properties which are the subclasses of the top class called WoTProperty. And a class named as FeatureOfInterest is also included to describe the WoT_Entity more properly. All the subclasses of WoT_Entity, WoTProperty and FeatureOfInterest are shown in **Figure 5.** We can see that there are two basic types of the class Device which are Sensor and Actuator respectively. FeatureOfInterest is used to describe what type of function an entity of WoT_Entity is about such as IndoorAirQuality or IndoorTemperature. There are many subclasses under the class WoTProperty. These subclasses are adopted to describe the information or properties about the subclass of WoT_Entity such as the location of the device, environmental parameter to detect, the unit of the sensory value and etc. In another way, these properties are the target annotations which are originally attached to the device manually by the user himself.

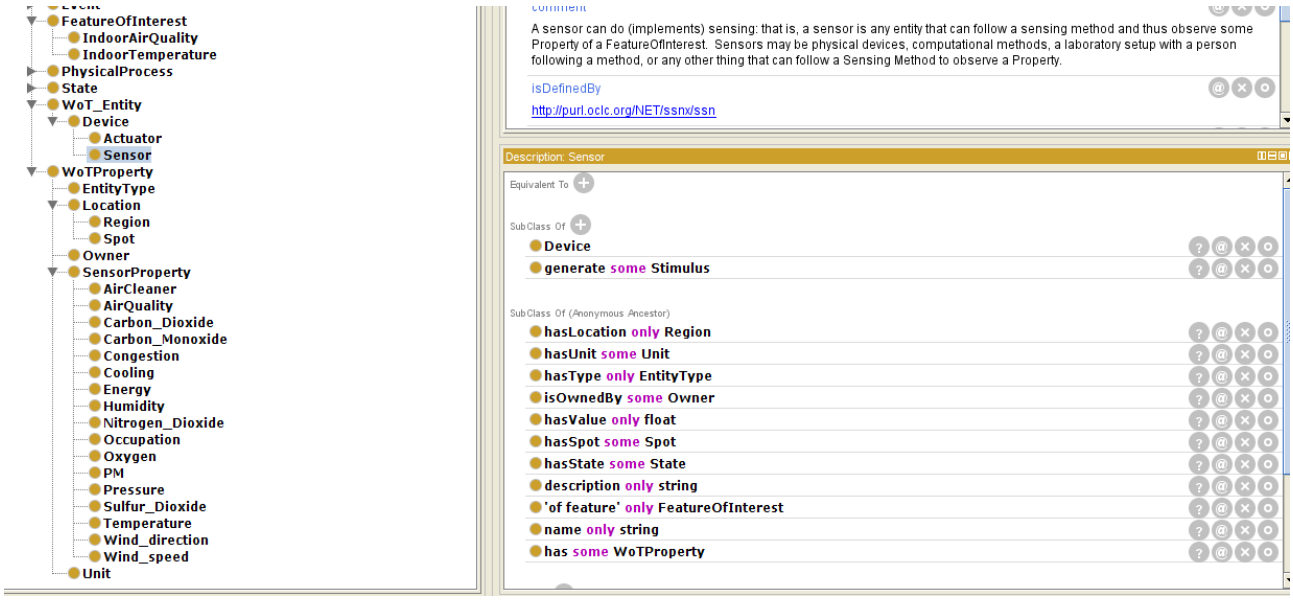


Figure 5 Subclass hierarchy structure of WoT_Entity, WoTProperty and FeatureOfInterest

The other classes are for the functionalities of anomaly diagnosis and automatic controlling. The subclass details for these class are shown in **Figure 6**. The class State is used to annotate a device's present working status which can help to find whether there are some anomalies in the environment of interest. And Anomaly and ObservedCause are subclasses of Stimulus which is a subclass of

Event. When an anomaly is observed, the causes resulting the anomaly need to be found out. With respect to each observed cause, actuators need to take actions to change the environment to remove the anomaly. TurnDown and TurnUp are the two subclasses of the class Action. The class PhysicalProcess which consists of NegativeCorrelationProcess and PositiveCorrelationProcess is used describe the relation between the actuators and the corresponding environmental property of interest. The class dataTime is used to record the present time when needed to be attached to an event.

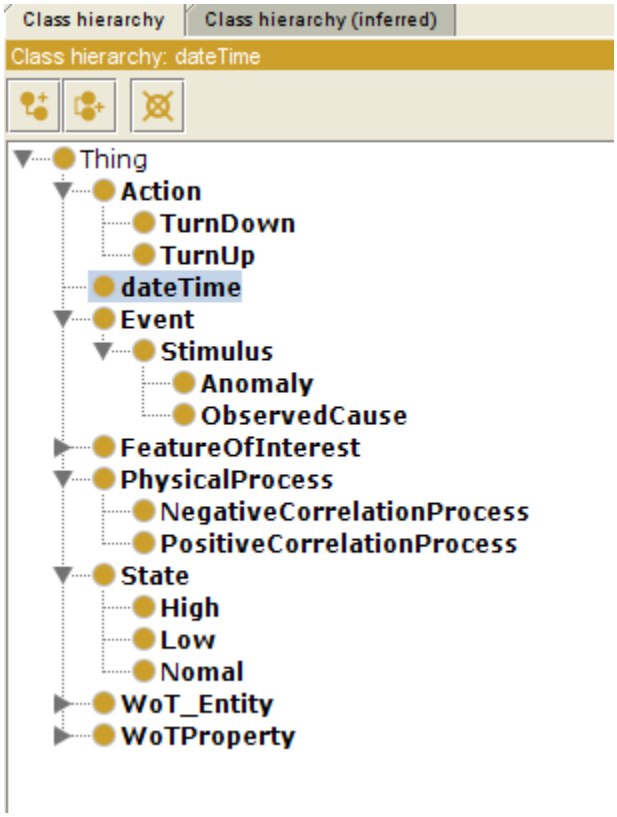


Figure 6 Subclass details for the rest classes

With the help of this ontology model, the annotation system can construct a local structural domain dependent knowledge graph which can improve the usability of the application data. In the sense of engineering project, the RDF linked data is the knowledge graph wanted.

3.3 Automatic annotation tool

Automatic annotation tool aims at linking the local knowledge graph to the global domain independent knowledge base. This operation is actually to link a local entity to a global entity to help computers to interpret what exactly the local entity means. The work flow of the automatic annotation tool is shown in **Figure 7.**

Figure 7 Work flow for automatic annotation tool

Candidate generation consists of Query and Re-rank phase for each cell values and a process for generation for candidates for column headers (classes) and relations between types.

For each cell value, the query module generates a ranked list of possible assignments from DBpedia. The ranking criterion adopted by DBpedia is entity popularity score. DBpedia provides web online query service, so we can get the service at its URL. The sqarql query commands for a cell text is shown in Figure 8.

```
Input: sparql statement
select DISTINCT ?s1
    ( sql:rnk_scale ( <LONG::IRI_RANK> ( ?s1 ) ) ) as ?rank
        where
          {
              ?s1 rdfs:label ?o1 .
              ?o1 bif:contains "( QueryString)" .( e.g.    Humidity )
              Filter regex (str(?s1),"resource").
          }
Output: All matching instance from KB
(e.g. Humidity ,Absolute_Humidity,Relative_Humidity...)
```

Figure 8 sparql query commands to DBpedia

The cell value which corresponds to an entity is used as the query string. And then DBpedia will return a ranked list subject to the query conditions. The query string should be fuzzily matched under the relation “rdfs:label” and the candidate entity should be of the type “resource”.

We expect the candidate set to be listed according to the possibility in which what the actual content in the cell could be. But the ranking result does not fully meet our requirement. For example, I take “eu” as the query string. In the result list, the first is “http://dbpedia.org/resource/Euroscepticism” and the second is “http://dbpedia.org/resource/Unitary_patent”. http://dbpedia.org/resource/EU_(European_Union) which is the actual target we want is out of the top 100.

To improve the confidence of the ranking score, I use a SVM ranking classifier[8] to re-rank the list returned by DBpedia. The SVM ranking uses string similarity and entity popularity as metrics to evaluate how possible an entity is what we want. The explanation of the two concepts are shown as follows.

•String similarity

The SVM ranking classifier compares the syntactic features between the query string and the entities in the candidate sets. Several metrics are considered in its ranking mechanism:

- Levenshtein Distance
- string length
- Dice score

Several 0-1 checks are also adopted by the SVM ranking module to see whether the entity equals to the query string or fully contains the query string. The SVM ranking tool checks whether all words in the query string are matched in the original order in the candidate entities, too.

•Entity popularity

Entity popularity is a kind of prior probability feature. It has already been integrated in DBpedia.

The score of entity popularity is listed along with the candidate entities returned from DBpedia. It can be helpful when entities have similar names. This score is generated according to how frequently the target entity is referenced by other entities. That is, the more references to this entity, the higher the score is.

In principle, the candidate set of types and relations are generated from the corresponding entity sets. The candidate set for type is exactly the union of all the candidate sets of type for corresponding candidate entity. And the candidate sets of the relations between each pair of types can also be returned from DBpedia.

For each candidate entity, I sent a new query to DBpedia and DBpedia can return a set of possible types. This query takes the entity as the subject and "rdf:type" as the predicate. The objects returned are of our interest. As for the relations between types or keys, I did another query to DBpedia which takes one type as subject and another type as object. DBpedia can return a set of possible predicates for the pair of types. The predicates are exactly the relations we want.

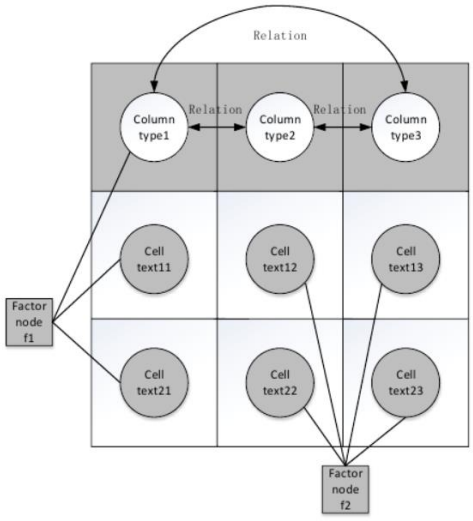


Figure 9 factor graph of a given table

In the stage of jointly reasoning, this paper first maps the input table into a factor graph model

which is a kind of probability graph model. The factor graph for a given table is shown in Figure 9. The circular grey nodes are entity nodes who holds attribute values corresponding to every fields. Each row can be regarded as a device instance. The circular white nodes represent the types of all entities in corresponding columns. The schema keys or attribute names are mapped to the values of these type nodes as column headers. These two kinds of nodes of column headers and cell values are variable nodes for the following iterative message passing algorithm. And the square grey nodes are factor nodes in the message passing algorithm. Factor node f1 works as functions of every variable nodes in each column to generate the most possible column type. Factor node f2 works as functions of every pair of cell values in each row to genetate the most possible relation between column types.

The iterative message passing algorithm takes the top candidate value from the list output by re-ranking module as the initial value for each cell entity. In each iteration, variable nodes transfer their values to the factor nodes they connected firstly which means that values of the variable nodes are the paragrams of the connected factor nodes. And then, factor nodes f1 and f2 run to generate the most possible value for each column type and relation between each pair of column types respectively. After that, the factor nodes will pass the result to every variable nodes they connect with an “change” or “unchange” message. The variable nodes will reassign their values accoding to the messages they received. This process will iterate until no change happens or the upper limitation is for the whole iteration is reached. And the details of how factor nodes f1 and f2 work are illustrated in the following two sections.

To draw a short conclusion, the values of column types and relations between each pair of column types are generated from the assignments of all the cell entities. And column types and the relations between columns also work as constraints to disambiguate the values of cell entity. The iterative constraints finally link the domain-specific knowledge base to the global domain independent knowledge base. Therefore, in this way, we can know that some different description of keys in different schemas refer to the same entity in the domain independent knowledge base. Thus, the ambiguation disappears and some further operations such as semantic search or integration among different systems can be possible.

For each current assignment of the entity in one column, I set a query to DBpedia as mentioned in section 3.2.3. And then, the factor node f1 will count, in the column, how many times each type appears as the score of this candidate type. The top one with the highest score is exactly what we want. When two candidates have the same score, this algorithm considers the granularity as the second metric. This is to see how many instances belong to the target type. The less instances a type has, the more specific the type is. Of course, want the one with a minimum granularity.

There is a problem here worth mentioning. The type with the top score may not be actually the one we want. It just seems to be the most possible one according to reasoning. So, setting a threshold is a better choice. When the top score is under the threshold, treat it as with a low confidence, and do not annotate the column with this type.

When variable nodes receive the message from the factor nodes. It will reassign its value according to the guide in the received message. If changed, actually, it will connect to a new class coming from the domain independent knowledge base.

The algorithm for relations reasoning is similar to that for type reasoning. One difference is that the factor node f2 takes the current assignments of two different cell entities in the same row. And then the message should be returned to both variable nodes. Another difference is that the relation between two columns can be bi-directional which means one can be the subject with the other to be the object and vice versa.

As for other details, the implementation is a web application technically. This web application uses DBpedia to be the domain independent knowledge base. The tool for the re-ranking phase is an open source SVM ranking classifier. And, the whole web application is based on Java language with Spring + Spring MVC + Mybatis framework. And the application uses Jena-tdb to store the RDF linked data and Mysql to store mapped tables from user inputs.

Chapter 4: Results and Use cases study

4.1 Experiment results

In this section, the results after annotation are shown as knowledge graph with the help of Neo4j.

4.1.1 Manual annotation tool

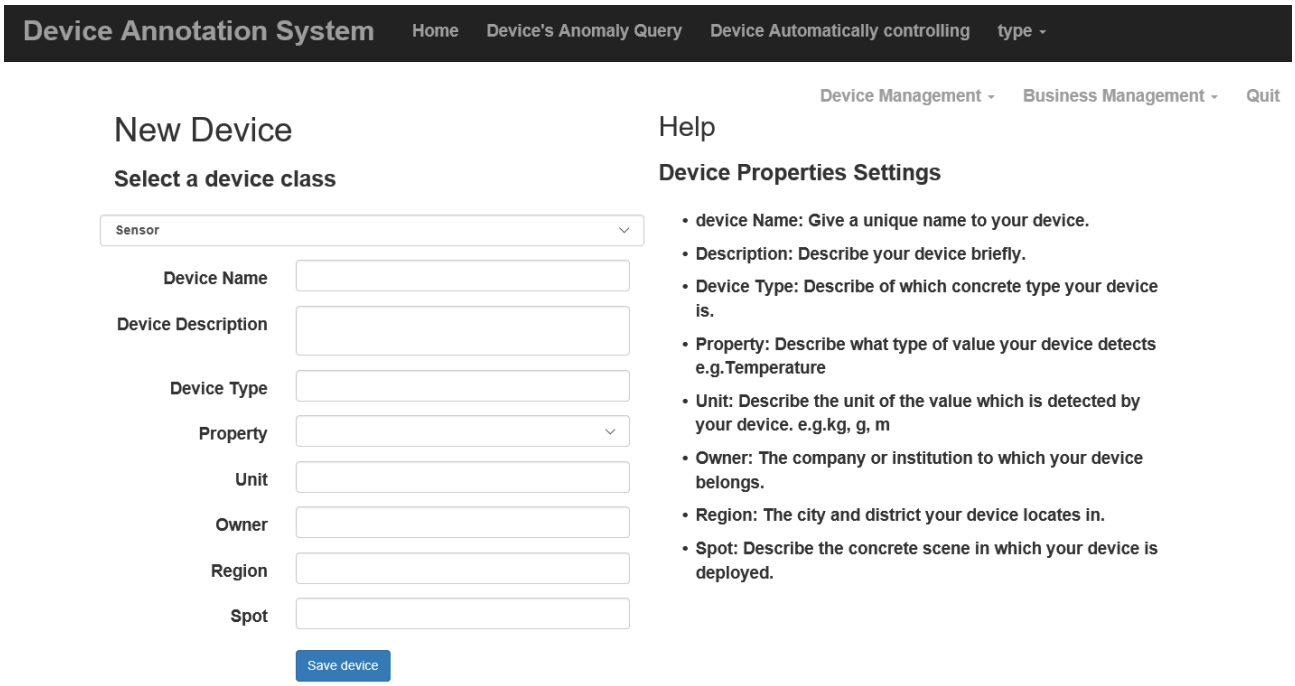


Figure 10 manual annotation input page

The input page for manual annotation subsystem is shown in **Figure 10**. User inputs various annotations into the boxes. And the application will store the information into the storage system. The initial local knowledge base is shown in **Figure 11**. The initial knowledge graph is constructed according to the predifined ontology model. After manual annotation, the domestic knowledge base is shown in **Figure 12**.

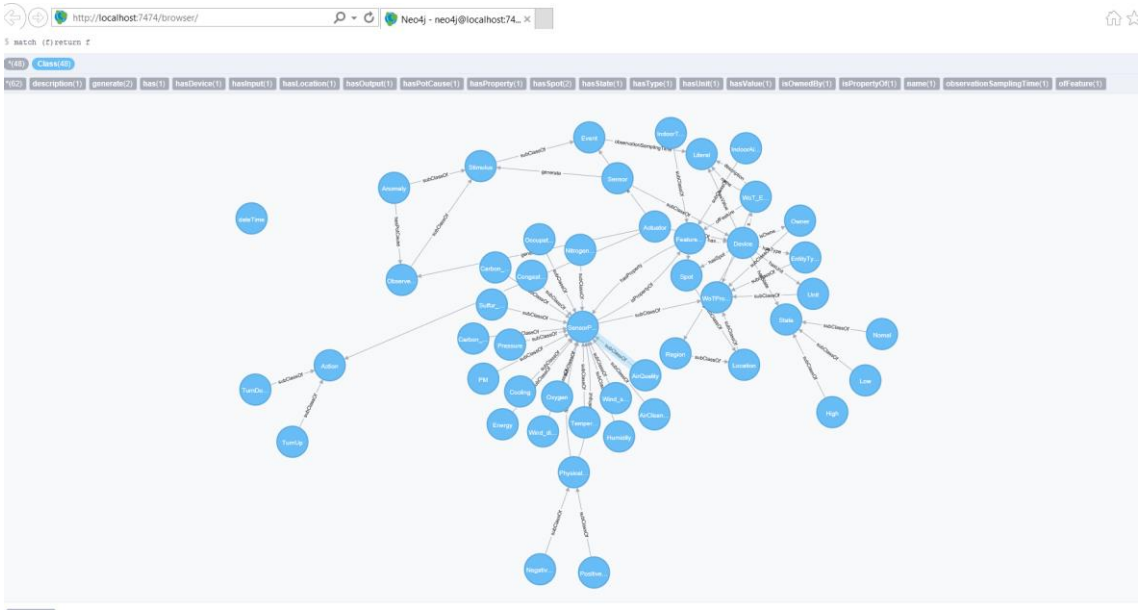


Figure 11 initial local knowledge base

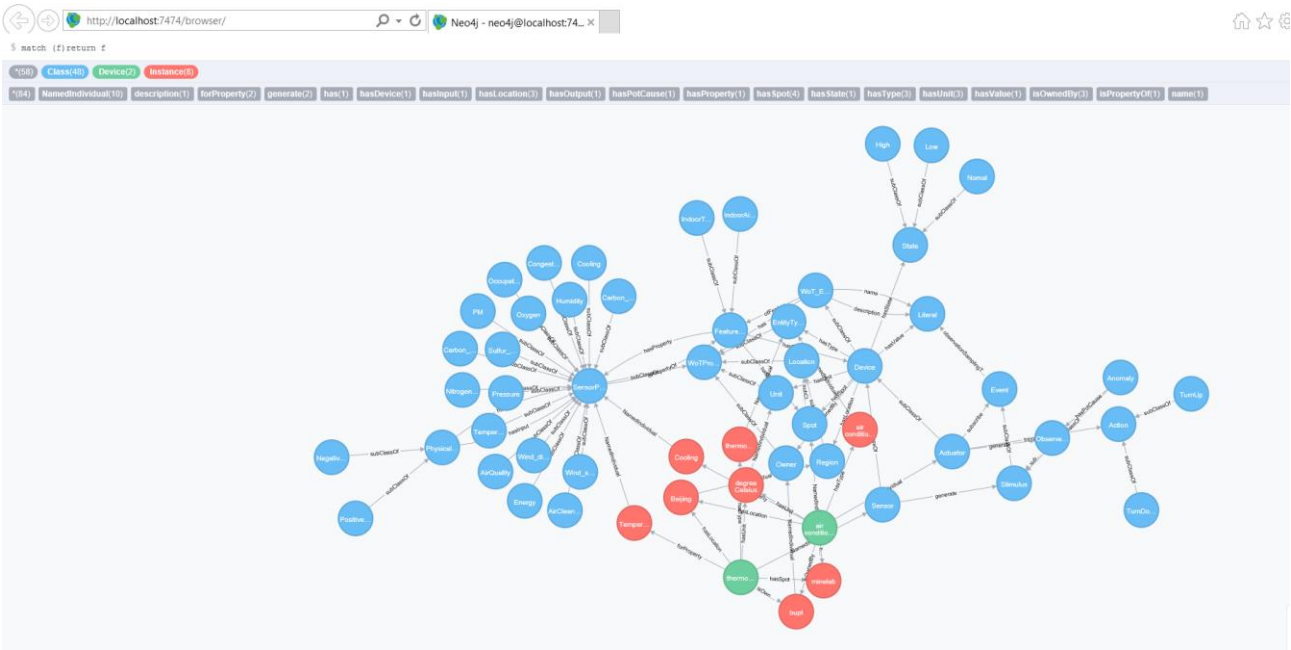


Figure 12 local knowledge base after manual annotation

4.1.2 Automatic annotation tool

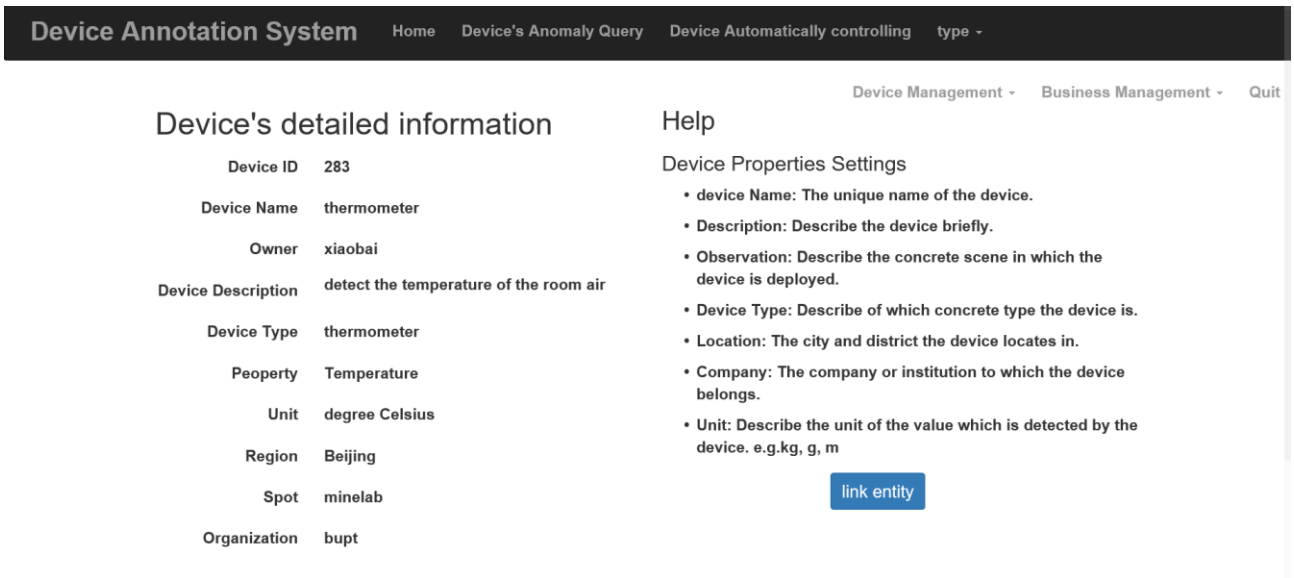


Figure 13 entrance point page for automatic annotation

The entrance point for the automatic annotation is shown in **Figure 13**. After entering the details of target device, click the “link entity” button. And then the web application will run the entity linking module. The results are shown in **Figure 14**. The blue nodes are the initial classes constructed according to the predefined ontology model. The green nodes are devices input by the user. The red nodes are the annotations for the devices added manually. The pink nodes are the entities from DBpedia which have connections with local entities. In another way, the local knowledge graph is extended by the entity linking algorithm with the help of DBpedia.

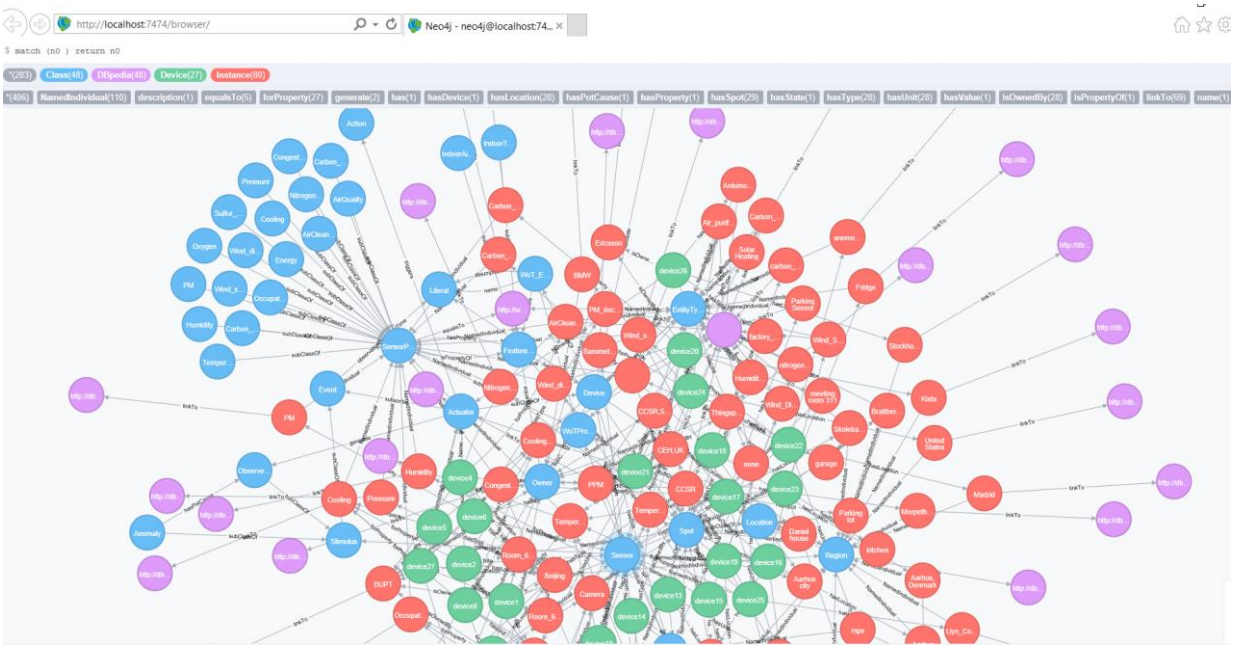


Figure 14 local knowledge base after automatic annotation

4.1.3 Performance

There is a notable phenomenon worth paying some attention. The iteration of the iterative message passing converges very quickly. After the first two iterations, there is no variable node to be changed. Two reasons may account for it. One is the contribution of the re-rank module. The re-rank module tries to move the target assignment to the top of the list with higher accuracy. The other reason is that the number of the relations between columns may be little. The more relations there exist, the more constraints upon one variable node need to be considered. Thus, less relations may result in less iterations.

And another thing is that the time consumed for each annotation table varies considerably with respect to the number of the rows and the columns in that table. It is reasonable that more time is needed if there are more rows and columns since more nodes and messages need to be processed.

4.2 Use cases study

The final output of the whole annotation system is a RDF linked knowledge graph taking a domain independent knowledge base into consideration. The usage of this knowledge graph is of great importance. In this section, I will propose several use cases for the purpose of guidance for the usage of the knowledge graph. The use cases are illustrated in the form of the story board as shown from **Table 1** to Table 3.

A typical use case of the annotation system is smart building. Many sensors and actuators are deployed in the building in advance. User needs to manually annotate all the devices complying with a predefined ontology model through front end (HTML). The web application then can run entity linking algorithm to link local knowledge base to DBpedia. In the use case of smart building, anomaly diagnosis from sensory data and automatic controlling to change the environment can also be integrated into the annotation system based on annotations reasoning.

Table 1 Anomaly diagnosis

Story name	Temperature sensing	Story ID	01
As a	smart home user		
I want	to receive warning when the temperature turns abnormally high or low and know about the causes for the anomaly.		
So that	I can take actions to make the temperature stay back in the normal interval.		

Acceptance Criteria • The anomaly and the causes must be correct. • Some alerting signal must be receivable (e.g. alarm in the house, alarm on the cellphone). • The causes must be shown to the host (e.g. log, short messages).
Notes • Regular checking and reasoning mechanism are needed. • The threshold of the normal temperature and the anomaly should be considered carefully.

I ran a demo for the application scene of temperature sensing. And the result is shown in **Figure 15**. The system captured a temperature high anomaly and the causes are found from device2, device5 and device3.

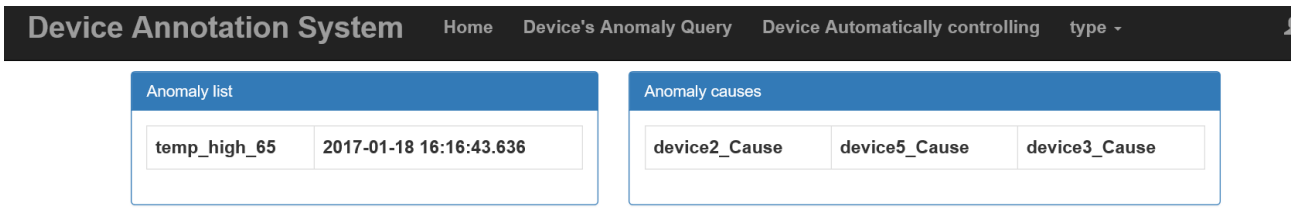


Figure 15 The result for a temperature anomaly

Table 2 is about a use case of automatic temperature controlling with respect to the anomaly captured in **Figure 15**.

Table 2 Automatic controlling

Story name	Temperature adapting	Story ID 02
As a	smart home user	
I want	to control the temperature to stay back in the comfortable range when temperature anomaly occurs.	
So that	I can keep the room comfortable in terms of temperature.	

Acceptance Criteria • Actuators must be successfully run. • What actuators have been run and what type (e.g. turn up) of the operation must be logged. • The anomaly should be cancelled and the result must be also logged.
Notes •The correspondence between anomaly causes and actuators and how actuators influence the environment must be clear. •What the system should do when anomaly still exists after running the actuators is of great importance. It should be a common self-adapting mechanism.

The result is shown in **Figure 16** with the help of a software called prot ég é which is used to display

and edit the ontology. We can see that device5 works as an actuator to cool the room. It subscribes the temp_high_65 event and triggers an action named device5_action_up. In terms of simulation in the software level, the automatic controlling is successful.

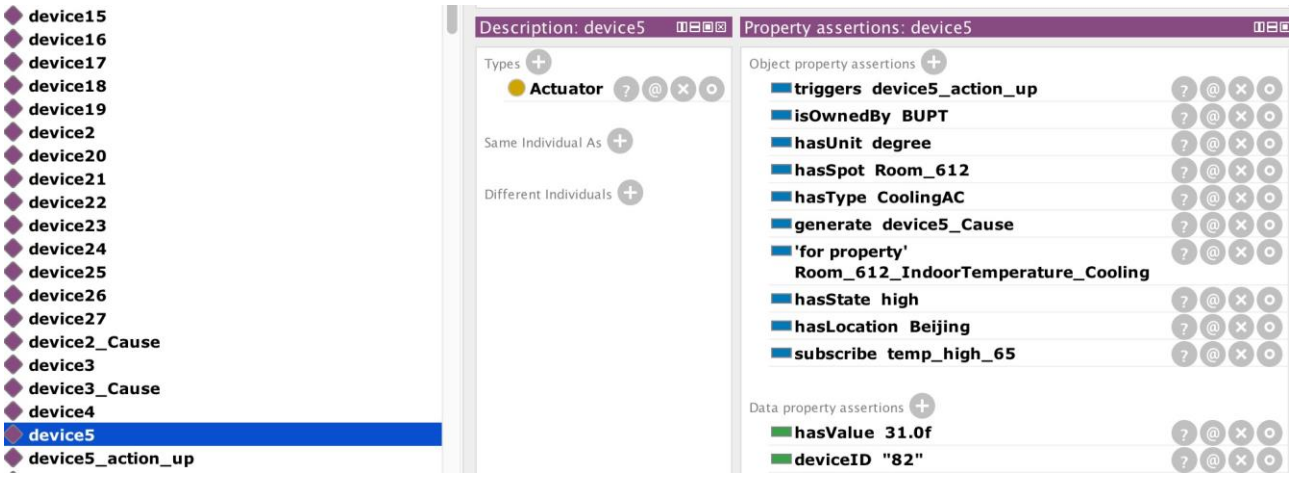


Figure 16 The result for the temperature automatic controlling

Semantic searching is also a promising application based on the networked knowledge infrastructure. Since the local knowledge graph can be linked DBpedia by the annotation system, sharing the knowledge among domain-specific applications is possible.

Table 3 Semantic searching

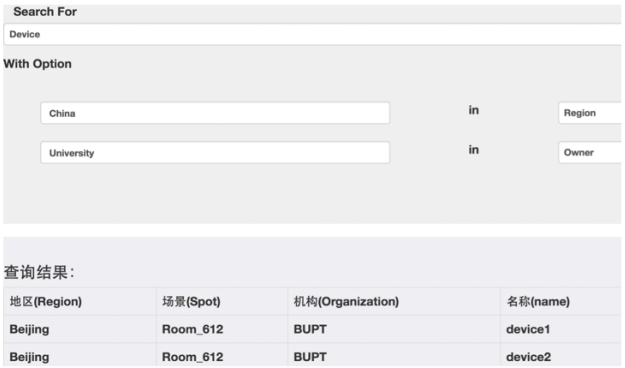
Story name	Target searching	Story ID	03
As a	WoT application developer		
I want	to search a knowledge graph belonging to another application for some useful information.		

So that I can implement some kinds of reasoning function modules in my application.

Acceptance Criteria • A public interface must be opened to other application developers. • The result set must support multiple formats (e.g. JSON, XML, RDF). • The functionality module must support searching proper relations given particular entities. • The functionality module must support searching proper entities given particular relations.
Notes • This functionality module should be abstracted as a web service.

I ran a semantic searching demo using SPARQL based on the knowledge graph generated by the semantic annotation system. The query sentences and the result list are shown in **Figure 17**.

```
SELECT DISTINCT ?deviceID
WHERE {
?device swot:deviceID ?deviceID.
?device dul:hasLocation ?loc_local.
?loc_local swot:linkTo ?loc_el.
BIND(URI(?loc_el) as ?loc_el_uri).
?loc_local a swot:Region.
        SERVICE <http://dbpedia.org/sparql> {
                ?loc_el_uri ?rel ?loc_db.
                FILTER regex(str(?loc_db),"China")
        }
}
```



查询结果：

地区(Region)	场景(Spot)	机构(Organization)	名称(name)
Beijing	Room_612	BUPT	device1
Beijing	Room_612	BUPT	device2

Figure 17 Query for automatic searching and the corresponding results

Chapter 5: Conclusion and Further Work

5.1 Conclusion

The semi-automatic sematic annotation system contains both the manual and automatic annotation tools. The manual annotation tool is used to construct a domain dependent structural knowledge base. The automatic annotation tool aims at link the local knowledge base built by manual annotation module to DBpedia which is a domain independent knowledge. The automatic annotation tool uses an entity linking (EL) algorithm based on iterative message passing using factor graph model. The EL algorithm adopts semantic techniques for WoT applications to bridge the local domain specific knowledge base to the global domain independent knowledge base. Then, ambiguities among different schemas towards the same type disappear since different local annotations are linked to the same type in the global domain independent knowledge base. Thus, a growing networked knowledge infrastructure can be achieved. And some use case study is also included in this paper.

5.2 Future work

Based on the knowledge graph base generated by the annotation system, some further functions can be integrated and improved into the web application such as anomaly diagnosis, automatic controlling and semantic searching. In terms of the functionality of anomaly diagnosis and automatic controlling, the design of the ontology model can be improved. The mechanism in the ontology used in my demos is just simple as a schema of observation-cause-action. The functionality of automatic controlling may adopt the service model according to Nicolas Seydoux et al [9]. When it comes to semantic searching, the functionality could be encapsulated as a web service with a well-defined API or a user interface.

References

[1] http://www.dblab.ntua.gr/~bikakis/XML%20and%20Semantic%20Web%20W3C%20Standards%20Timeline-History.pdf

[2] Villalonga, C.; Bauer, M.; Huang, V.; Bernat, J.; Barnaghi, P. Modeling of sensor data and context for the real world internet. In Proceedings of the 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mannheim, Germany, 29 March–2 April 2010.

[3] Patni, H.; Henson, C.; Sheth, A. Linked Sensor Data. In Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS 2010), Chicago, IL, USA, 17 May–21 May 2010.

[4] Wang, P.; Wang, H.; Wang, W. Finding semantics in time series. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (SIGMOD' 11), Athens, Greece, 12–16 June 2011; pp. 385–396.

[5] Ganz, F.; Barnaghi, P.; Carrez, F. Automated Semantic Knowledge Acquisition from Sensor Data. IEEE Syst. J. 2014, 10, 1214–1225.

[6] Limaye, G.; Sarawagi, S.; Chakrabarti, S. Annotating and searching web tables using entities, types and relationships. Proc. VLDB Endow. 2010, 3, 1338–1347.

[7] Mulwad, V.; Finin, T.; Joshi, A. Semantic message passing for generating linked data from tables. In The Semantic Web–ISWC 2013; Springer: Cham, Switzerland, 2013; pp. 363–378.

[8] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Acknowledgement

I would like to give my great gratitude here to my supervisor and a senior fellow apprentice in my laboratory who are really nice persons. They give much help and guidance to me when I meet problems. I learned a lot from them with this project.

Appendix

北京邮电大学
本科毕业设计（论文）任务书
Project Specification Form

学院 School	International School	**专业 Programme**	Telecommunications with Management (H6N2)	**班级Class**	2013215108
学生姓名 Name	DU Sicong	**学号 BUPT student no**	2013213144	**学号 QM student no**	130801256
设计（论文）编号 Project No.	RC_3144				
设计（论文）题目 Project Title	A Semantic Annotation System for Web of Things				
论文题目（中文）	一个物联网的语义标注系统				
题目分类 Scope	Research	Computer Software	Hardware and Software		

主要任务及目标Main tasks and target：	By
Task 1: Survey on Web of Things (WoT), semantic web and entity linking (EL) technologies, as well as designing the annotation framework	15 February 2017
Task 2: Designing manual annotation tools for smart things with a predefined WoT ontology	31 March 2017
Task 3: Designing automatic annotation tools for smart things with EL model via linking entities to a commonsense knowledge base	31 March 2017
Task 4: Implementation of the framework and evalutions	24 April 2017

Measurable outcomes
1) A manual annotaiotn subsystem with predefined WoT ontology
2) An automatic annotation subsystem with EL model
3) A proof of concept demos and evaluations of the whole framework

主要内容Project description：

Web of Things (WoT) framework integrates smart devices into Web by resusing and extending Web standards. It unifies the syntactic representations of physical objects via web pattern, which facilitates the integrations and mashups of heterogeneous data and web services. However, the lack of unified representation markup tools and methods at semantic layer hinders the interoperability, integration and scalable search of things.This project aims at designing and implementing a system to semi-automatically annotate WoT resources with both knowledge engineering method (top-down) and machine learning method (bottom-up). The system could generate and maintain a knowledge base for IoT domain with linked sensor data.

Project outline

Firstly, I will survey and learn the related knowledge including Web of Things(WoT), semantic web , entity linking(EL) technologies and any other necessary knowledge. The learning process may be needed at any iteration in the engineering. After that , I will take steps to learn about the principles on common system or tools design. With the help of that, I am going to achieve the design and implementation work following the guidance of software engineering. Although there is not a team for this project. I think that some methods of software engineering are nice choice.

What I expect to have working at the mid-term oral

I want to achieve the draft design work of the manual and automatic tools for smart things and implement the two tools with some simple draft program. And the algorithm on entity linking(EL) is going to be implemented.

北京邮电大学

BBC6521 Project 毕业设计 2016/17

Early-term Progress Report

初期进度报告

学院 School	International School	专业 Programme	Telecommunication Engineering with Management	班级 Class	2013215108
学生姓名 Student Name	Sicong Du	BUPT 学号 BUPT Student No.	2013213144	QM 学号 QM Student No.	130801256
设计（论文）编号 Project No.	RC_3144	电子邮件 Email	Sicongdu@bupt.edu.cn		
设计（论文）题目 Project Title	A Semantic Annotation System for Web of Things				

已完成工作：
Finished Work：

1.I did some survey work on the Web of Things(WoT), semantic web and entity linking(EL) technologies through Baidu .

2.I got a knowledge of Java web including Java, JSP , HTML, XML, JSON, RDF, graph database and some relative things through http://www.w3school.com.cn/ , http://www.runoob.com/ and some relevant books .

3.I have learned about the basic functions and framework of a annotation system.

是否符合进度？ On schedule as per GANTT chart?	[YES]

下一步：
Next steps:

1. Design a manual annotation tool for smart things with a predefined WoT ontology.
2. Implement the subsystem of the manual annotation tool.
3. Starting to learn relative algorithms to implement the functions of a automatic annotation tool.

北京邮电大学

BBC6521 Project 毕业设计 2016/17

Mid-term Progress Report

中期进展情况报告

学院 School	International School	专业 Programme	Telecommunication Engineering with Management	班级 Class	2013215108
学生姓名 Student Name	Du Sicong	BUPT 学号 BUPT Student No.	2013213144	QM 学号 QM Student No.	130801256
设计（论文）编号 Project No.	RC_3144	电子邮件 Email	Sicongdu@bupt.edu.cn		
设计（论文）题目 Project Title	A Semantic Annotation System for Web of Things				
毕业设计（论文）进展情况，字数一般不少于 1000 字 The progress on the project. Total number of words is no less than 1000					
目标任务: Targets set at project initiation: (must be the same as "What I expect to have working at the mid-term oral" in the Spec) I want to achieve the draft design work of the manual and automatic tools for smart things and implement the two tools with some simple draft program. And the algorithm on entity linking(EL) is going to be implemented.					
是否完成目标 Targets met?	[YES]				

目前已完成任务 Finished Work:

1.I have achieved the manual annotation tool for smart things.

I implement the tool as a web application through the jsp technique including Spring framework. And I use two approach to store my data. One is Mysql which is a database for future entity link implementation to communicate with DBpedia which is a public knowledge base. The other one is Jena-TDB, whose input and output are both OWL format. Jena-TDB is used to carry out some reasoning operation in the future for which some existing libraries can be used. The function of data persistence and query is implemented.

2.And I have also accomplished some simple draft or supporting programs for the automatic annotation tool. But the core module covering entity linking algorithm is left to be implemented in the next step.

3.I have read several papers and got a draft model of entity linking algorithm. And I have also got some idea for scenery verification of my annotation system.

● A model called semantic message passing can be used in my algorithm of entity link (EL). The core idea is illustrated below.

At the beginning, we can get a table from our database.

A Semantic Annotation System for Web of Things

i. Preprocessing Phase

An input table first goes through a preprocessing phase with modules to handle a number of pragmatic issues, such as sampling rows from large tables and recognizing and expanding acronyms and stylized literal values (e.g., phone numbers) and so on.

ii. Query and Rank

a. Generating and ranking candidates for cell values. The objective cell value is used as query string and When we send a query request for a cell value or a column header to DBpedia, the content of the cell value or the column header is used as query string and the contents of the column header and other row values are used as context when querying DBpedia.

b. Generating candidates for columns. Initial candidate classes for a column are generated from its cell values, each of which has a set of candidate entities, which in turn have a set of DBpedia classes.

c. Generating candidate relations between columns. We generate candidate relations for every pair of columns in the table, based on the cell value pairs in the respective columns.

iii. Joint Interference

Once the initial sets of candidate assignments are generated, the joint inference module assigns values to columns and row cell values and identifies relations between the table columns. The result is a representation of the meaning of the table as a whole. In the article, it takes an algorithm called Semantic Message Passing. All the table cells and column headers (variable nodes) send their contents to all the factor nodes they connect to. The factor node is actually a function. There are two types of factor nodes. One calculates the column header and row cell value agreement. The other calculates the relations between pairs of columns. These two functions adapt a mechanism of voting and scoring to generate the most possible values and pass the results to each variable nodes connecting to it for the next updating process. This process is iterated until convergence. And there is still a limit to the number of iteration to prevent the iteration running endlessly.

- Anomaly diagnosis and automatically controlling on the smart devices in the smart building are promising applications of my annotation system. It is possible for me to design a simple scene on anomaly diagnosis for verification.

尚需完成的任务 Work to do:

1. Implement the entity linking algorithm to complete the automatic annotation tool.
2. Integrate the subsystems.
3. Design an application scene on the anomaly diagnosis to verify the whole system.
4. Evaluate on the whole system.

能否按期完成设计（论文） Can finish the project on time or not:	[YES]

存在问题 Problems:

1. In the model of semantic message passing mentioned above, the preprocessing phase covers some redundant work. Because DBpedia is a relatively robust knowledge base. That some acronyms in our query string is OK, for example.
2. In the Query and Rank phase, DBpedia won't return ranked list of entities and classes. So, I need some other solution.

A Semantic Annotation System for Web of Things

拟采取的办法 Solutions:

1. I will skip the preprocessing phase since DBpedia is robust.
2. As for the ranking problem, I will take the SVM-rank (Support Vector Machine for Ranking) as a tool. The website can be found by http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

最终论文结构 Structure of the final report:

Abstract

A short overview of the whole report.

Chapter 1: Introduction

1. Purpose and main tasks of the project
2. Overview of the whole semantic annotation system
3. Summary of the methodologies
4. Key results and the novelty of the project

Chapter 2: Background

1. State of the art of the annotation
2. Problem statement of previous studies and my research goal

Chapter 3: Design and Implementation

1. Detailed description of the system model and the algorithms
2. Proof of the web application for annotation
3. Description and proof of the anomaly diagnosis for scenery verification

Chapter 5: Results and Discussion

Experiment results and performance evaluation of the core modules of the system.

Chapter 6: Conclusions and Future Work

1. A clear summary of the project
2. Potential further possibilities to improve the system and of more applications

References

Acknowledgements

Risk Assessment

Environmental impact assessment

日期 Date:	Mar 9th 2017

Risk Assessment

There is no physical safety risk since the semantic annotation system is a web application which only concerns software issues.

Environmental Impact Assessment

There is no negative environmental impact since the semantic annotation system is a web application which only concerns software issues.