

セマンティック技術を用いたリアルタイムセンシングデータ 検索システムの試作

下田 功一^{1,a)} DIEFENBACH Dennis² MARET Pierre² 戸辺 義人^{3,b)}

概要: 近年, IoT(Internet of things) と呼ばれるインターネットに接続されているモノが急速に普及している。また, その結果, インターネット上に多くの異なる形式を持つデータが登場している。セマンティックウェブ(もしくはデータのウェブ)は, 共有および再利用が容易なデータを公開するための標準で構成されている。データは RDF(Resource Description Framework) として保存され, SPARQL を利用することで参照することができる。これら, セマンティックウェブの主な利点の一つは, 共通の形式の下で異なるシステムの情報を扱えることである。セマンティックウェブ技術では IoT に接続されたデバイスによって生成されたデータを扱う方法がすでに提唱されている。本研究では, それらの技術を用い, さまざまなセンサによって生成されたデータを RDF 形式で公開することに加え, QAnswer と呼ばれる RDF データ用の Question Answering システムと自然言語を使用して, さまざまなセンサによって生成されたデータをリアルタイムで照会するシステムを試作した。

キーワード: セマンティックウェブ, センサデータ, Question Answering

Abstract: The Internet of things (IoT), i.e. the ensemble of devices that are connected to the internet, is growing fast. In particular this means that new very heterogeneous data stream are appearing on the internet. The Semantic Web (or web of data) consist of a set of standards to publish data that can be shared and reused easily. Data is stored using the Resource Description Framework (RDF) and queried using SPARQL. One of the main advantages is that it allows to align information of a heterogeneous system under a common umbrella.

Semantic Web Technologies where already proposed as a way to align the data produced by devices connected to the IoT. In this work we go one step further. We first expose the data produced by different sensors into RDF format. We then use an off the shelf Question Answering system for RDF data, called QAnswer, to query in real time, using natural language, the data produced by different sensors.

Keywords: semantic web, sensor data, Question Answering

1. はじめに

近年, パソコンやスマートフォンだけでなく様々なモノがインターネットに接続され, 情報を伝達する IoT が普及し始めている。2013 年時点では 158 億個ほどであったインターネットにつながるモノは 2020 年には 400 億個まで

増大すると言われている [14]。また, IoT の一部としてセンサデータなどの情報を収集, 分析し実社会に反映させる Cyber-Physical-System(CPS) も盛んに研究が行われている [10]。こうした動きが広まりインターネットに接続できるモノが増えるということはデータを記述するメタデータも多くの形式が存在し得ることとなり情報の統一性が失われてしまう。

個人や一般家庭規模で考えた時, IoT 家電などに代表される IoT 機器を扱う際にメーカーごとに違う規格で作られているそれら进行操作するために多くのソフトウェアやスマートフォンアプリケーションが必要になり, 利用者に不便さを感じさせてしまう原因にもなる。企業や都市などのより大きな規模で考えた時, 一般での使用よりも多くのセンサや IoT 機器を使う可能性が考えられる。これらの場

¹ 青山学院大学大学院理工学研究科理工学専攻
Graduate School of Science and Engineering, Aoyama Gakuin University

² サンテティエヌ・ジャン・モネ大学
Université de Lyon, CNRS UMR 5516 Laboratoire Hubert Curien, France

³ 青山学院大学理工学部情報テクノロジー学科
Department of Integrated Information Technology, Aoyama Gakuin University

a) koichi@rcl-aoyama.jp

b) yoshito-tobe@rcl-aoyama.jp

合を想定した時、異なるメタデータを持ったデータを効率よく収集することは困難であると考えられ統一されたメタデータの必要性が伺える。

ウェブ上で、ウェブページの情報に単なる文字列としてではなく具体的な意味の情報を与えることによって利便性をあげるはW3C(World Wide Web Consortium)によって提唱されたプロジェクト、セマンティックウェブがある。セマンティックウェブでは統一されたメタデータ形式 RDF を用いてデータを機械が扱いやすい形式にし情報の利便性をあげることが目的の1つとしてある。実際に、LOD(Linked Open Data) Cloudと呼ばれるウェブ上のデータセット群には RDF 形式で扱うことができる多数のデータセットが公開されている。この RDF に対してデータを扱うためのクエリ言語 SPARQL が標準化されている。しかし、クエリの可読性を考えた時に、一般の人間が直感的に気軽に SPARQL を用いて検索を行うことは困難である。この問題を受け、LOD のデータに対して SPARQL を自然言語に自動で変換し検索できる Question answering システムが発展してきた。

セマンティックウェブ技術を利用して IoT で生成された異なるデータを整理することはすでに提案されており [7]、特定のデータモデルはすでに作成され標準化されている [4]。本研究では、自然言語を使用してセンサによって生成されたデータにアクセスするために、Question Answering システムを利用したソリューションを提案する。

以下、2章では関連研究及び技術、3章ではシステムの実装、4章では評価、5章では結論を記す。

2. セマンティック技術

2.1 Resource Description Framework

RDF は W3C により標準化されている、ウェブ上のデータを記述するための枠組みである。ウェブ上で公開されているオープンな RDF のデータセット群 LOD^{*1}上でデータを記述するための形式として使われている。RDF は主語 (subject) 述語 (predicate) 目的語 (object) により構成されておりこれらを合わせて triple と呼ぶ。また、主語と述語は Uniform Resource Identifier (URI) を持たないといけない制約がある。

RDF を用いてセンサのデータを扱う研究も行われており、スマートシティの開発に役立てようとしている動きがある [13][9][11]。本研究でもセンサのデータを扱う形式として RDF を用いる。

2.2 wikidata

wikidata は多言語で構成された構造化データベースで

あり LOD のデータベースの1つである。wikipedia などのウィキメディアプロジェクトで使用される情報源を提供することを目的としている。wikidata 上では項目を一位の番号によって区別しており RDF における object, subject に相当するデータは接頭辞 Q を持ち、Q1 は universe となっている。RDF の predicate に相当するデータは接頭辞 P を持ち P828 は "cause" となっている。また、主語 "universe" は述語 "cause" を保持しており、その目的語として "Big Bang" (Q323) を有する、[12]。この triple は '宇宙はビッグバンにより誕生した' といった関係を記している。wikidata からは RDF を記述する turtle 形式でデータを取得でき [3]、本研究では battery, temperature などのセンサやプロパティに関する項目を wikidata から取得しデータベースに保存することで各 object の説明文を取得した。

2.3 Semantic Sensor Network Ontology

Ontology とは特定のドメインのデータをモデル化し説明するための用語とプロパティを含む RDF のデータセットである。SSN (Semantic Sensor Network) Ontology はセンサとその周りの環境を RDF で記述するための Ontology として開発された [2]。SSN は基本的なクラスとプロパティに SOSA (Sensor, Observation, Sample, Actuator) を含めている。SSN と SOSA は衛星画像、産業と家庭のインフラ、社会センシング、市民科学、観測主導など、幅広いアプリケーションとユースケースをサポートすることができる。ontology は <http://www.w3.org/ns/sosa/> に記されており predicate として hasSimpleResult, hasFeatureOfInterest などを保持している。本研究ではセンサから取得した RDF の記述を SOSA を基に行った。

2.4 QAnswer

QAnswer は SPARQL クエリを自然言語に変換し LOD 上のデータを検索する Question Answering システムの1つである [6][5]。Jean Monnet 大学が開発している。ユーザが質問をすると、適切と思われる答えを LOD 上から検索し、提示してくれる。wikidata, openstreetmap, dbpedia など多くのデータセットを扱うことができ言語も英語、中国語、アラビア語など多数の言語が使える点が優れている。

図1ではデータベースに wikidata を選択し 'What is the population of Japan?' という質問をした結果である。QAnswer はこの質問の答えを wikidata より取得しており、'127,110,047' という日本の人口を表す結果を示している。これは wikidata に

```
'SELECT DISTINCT ?o1 where {  
<http://www.wikidata.org/entity/Q17>  
<http://www.wikidata.org/prop/direct/P1082>  
?o1 . } limit 1000'
```

という SPARQL query を送信した時と同じ結果を示し

*1 <https://lod-cloud.net/>

ており Q17 は Japan,P1082 は poplation を示しているの
 で主語 Japan, 述語 poplation とした時の目的語を表示して
 いる. 表示された答えが間違っていた時には他の候補から
 正しい結果を探すことができフィードバックを送ることで
 ユーザの手によって精度を向上させることもできる.

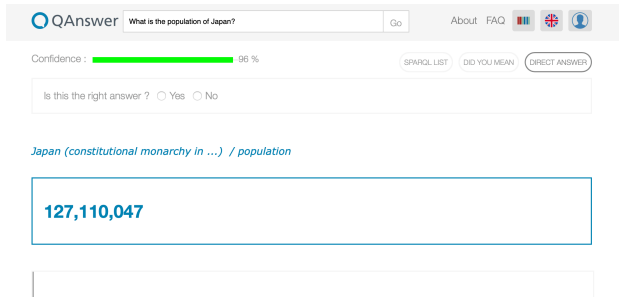


図 1 QAnswer での日本の人口に関する質問

3. 実装

この章ではセンサデータを RDF 形式に変換し, 自然言
 語で問い合わせるシステムの構成を説明する.

本システムは pysense, iPhone, センサデータを受け取る
 HTTP サーバ, RDF データベース fuseki, QAnswer によっ
 て構成する (図 2).

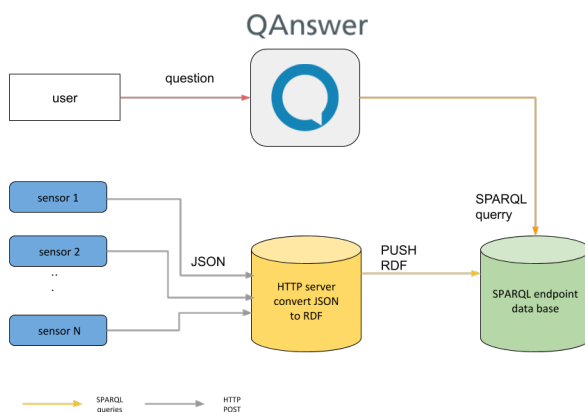


図 2 システム概要

3.1 センサ

センサとして Pycom が開発するセンサシールド Pysense
 とボード Lopy4, Apple が開発する iPhone SE を用いた.

図 3 のように Pysense と Lopy4 はセットで用いられ,
 Pysense には以下のセンサが搭載されている.

- 加速度計
- 湿度計
- 光センサ
- 気圧計

● 温度計

これらは特定の部屋に設置し, 部屋の温度や湿度などの
 環境を測るのに用いる. Lopy4 を Wi-Fi アクセスポイント
 に接続し, JSON 形式でセンサデータと MAC アドレス,
 部屋番号を HTTP サーバに POST した. iPhone では
 AppStore でリリースされているアプリ, SensorLog 上で
 iPhone のセンサを用いて以下の値を計測した.

- 加速度
- 歩数
- 位置情報
- 方位
- 充電残量

これらも Lopy4 同様に JSON 形式で HTTP サーバに POST
 した.



図 3 pysense と lopy4

これら 2 つのセンサから送られる情報は JSON データで
 はあるが異なる形式で記述されている.

3.2 HTTP サーバ

Python を用いて HTTP サーバを作成した. 役割は, a)
 センサが送信した JSON を受け取ること, b)JSON を RDF
 へと変換すること, c) 時間情報の取得, d)RDF をデー
 タベースへと送信すること, e)heatindex の計算である.

Lopy4,iPhone から送られてきたデータをリアルタイム
 で SSNO に基づく RDF 形式に変換し, データベースの
 endpoint へと SPARQL query で送信した.

heatindex は温度と湿度より求められる温熱指標の一つ
 である [1]. 以下の式 (1) により算出される.

$$0.4 * (Temperature + Humidity) + 4.8 \quad (1)$$

3.2.1 RDF

センサから得た JSON データを SSNO に基づいて RDF
 へと変換した. 図 4 は SSNO に基づいて書かれた実際の
 RDF で turtle 形式で記述されている. 図 5 はその構造を
 表したナレッジグラフである. センサから送られてきた情

報は predicate として resultTime, hasSimpleResult, hasFeatureOfInterest, observedProperty, madeBySensor を持つ observation として変換される。

resultTime の目的語 time は時間であり http サーバ がデータを取得した時間が記されている。hasFeatureOfInterest の目的語 result は結果であり、センサから取得された値が格納される。例えば, battery の場合は 0.73 といった残量が示される。hasFeatureOfInterest の目的語 user は result の観察対象でありバッテリーの場合は iPhone, 温度センサの場合は観察されている部屋となる。observedProperty の目的語 property は観察される性質である。battery の場合は battery 自体であり URI を wikidata の battery(Q267298) から取得している。madeBySensor の目的語は sensor であり, デバイスに搭載されている具体的なセンサを記す。user, property, device を除く observation 自体と time,result はセンサから送られてくるたびに新たなデータがとして更新されるのでその都度, 新しいデータがデータベースに追加される仕組みとなっている。

```

@prefix ns1: <http://www.w3.org/ns/sosa/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://example.org/Observation/1562492019.742482koichiB> a
ns1:Observation ;
  ns1:hasFeatureOfInterest <http://example.org/
koichiDevice> ;
  ns1:hasSimpleResult "0.730000 Per"^^<http://w3id.org/
lindt/custom_datatypes/ucum> ;
  ns1:madeBySensor <http://example.org/koichi/
batteryMeasure> ;
  ns1:observedProperty <http://www.wikidata.org/entity/
Q267298> ;
  ns1:resultTime
"2019-07-07T11:33:39.742525+02:00"^^xsd:dateTime .

<http://example.org/koichi> a ns1:FeatureOfInterest ;
  rdfs:label "koichi"@en .

<http://example.org/koichiDevice> a ns1:FeatureOfInterest,
ns1:Platform ;
  rdfs:label "koichi's iPhone"@en ;
  ns1:hosts <http://example.org/koichi/batteryMeasure> .

<http://www.wikidata.org/entity/Q267298> rdfs:label
"battery"@en ;
  ns1:isObservedBy <http://example.org/koichi/
batteryMeasure> .

<http://example.org/koichi/batteryMeasure> a ns1:Sensor ;
  rdfs:label "battery measure"@en ;
  ns1:isHostedBy <http://example.org/koichiDevice> ;
  ns1:observes <http://www.wikidata.org/entity/Q267298> .
    
```

図 4 turtle 形式で記述された RDF

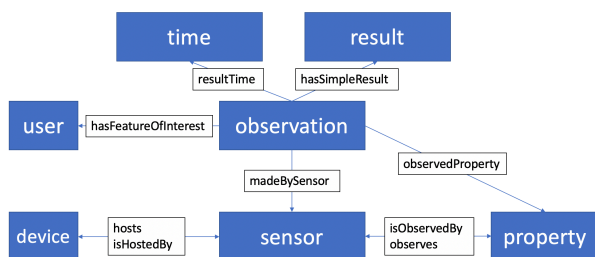


図 5 上述の RDF のナレッジグラフ

3.3 データベース

データベースとして RDF を保存することができる Fuseki を用いた [8]. endpoint を使うことにより, SPARQL でデータのアップロード, 外部からの検索を行うことができる。データベースはセンサから送られたデータ。SOSA のネームスペースから取得した class, property データ, wikidata から取得した property データによって構成されている。

3.4 QAnswer

データベースからのデータの検索に自然言語を用いることができる QAnswer を用いた。QAnswer では人が行なった質問を受けデータベースの endpoint に対し適切な答えの候補と思われる複数の SPARQLquery を送りその結果を受信する。それらの候補のうちから最も適切であると判断されたものが最初に結果として表示される。

4. 評価

4.1 環境

8 時間の間, 2 個の pysense を部屋に設置し 10 秒ごとにデータを送信, また, iPhone からセンサのデータを 10 秒ごとに送信した。それぞれのセンサの種類, 計測の対象は以下の表 1 の通りである。それらと, SOSA のネームスペースから取得した class, property データ, wikidata から取得した property の RDF データを含む 572843 個の triple を持つデータセットを作成した。

表 1 デバイスとセンサと対象

devices	sensors	interest
iPhone	accelerometer battery location compass pedometer	koichi
pysense Lopy4	accelerometer humidity light pressure temperature	F107 F108

4.2 方法

QAnswer を用いてデータベースに問い合わせをした時, データベースが全ての query を取得し, レスポンスを全て返すまでの時間を計測した。また実際に被験者を 1 名用意し QAnswer を通して検索すると SPARQL を直接打ち込み検索した時の結果を得るまでの時間を比較した。検索するワードとして 'koichi iPhone battery', 'F107 temperature', 'battery', 'what is humidity ?' を選択した。'koichi iPhone battery' は koichi というユーザに所有されている iPhone のバッテリー残量を問うための検索ワードである。

また、これは'What is the battery of koichi's iphone?'と問うのと同義である。'F107 temperature'はF107という部屋番号の部屋の気温を問うための検索ワードである。また、これは'the tempratire in room F107 is what?'と問うのと同義である。これらの質問の正しい答えが得られた時、それぞれの時系列データが一覧で表示される。'battery'、'what is humidity'はbattery, humidityといったそれぞれのpropertyの説明を問う質問であり、正しい答えが得られた時、wikipedia内のそれぞれの説明が結果として得られる。

表 2 質問と SPARQL query の対応

質問	SPARQL query
koichi iPhone battery (What is the battery of koichi's iphone?)	SELECT DISTINCT ?o1 where { ?s1 ?p1<http://example.org/koichiDevice>. ?s1 ?p2<http://www.wikidata.org/entity/Q267298>. ?s1<http://www.w3.org/ns/sosa/hasSimpleResult> ?o1. } limit 1000
F107 temp- erature (the temp- rature in room F107 is what?)	SELECT DISTINCT ?o1 where { ?s1 ?p1<http://example.org/F107>. ?s1 ?p2<http://www.wikidata.org/entity/Q11466>. ?s1<http://www.w3.org/ns/sosa/hasSimpleResult> ?o1. } limit 1000
battery	SELECT ?s0 where { VALUES ?s0 {<http://www.wikidata.org/entity/ Q267298>} }
what is humidity	SELECT ?s0 where { VALUES ?s0 {<http://www.wikidata.org/entity/ Q180600>} }

4.3 結果

結果として表 3 に示す通り SPARQL クエリを受け取り結果をデータベースが検索し終えるまでに'koichi iPhone battery'では341ms, 'F107 temperature'では371ms, 'battery'では58ms, 'what is humidity?'では53msかかった。図 6 は実際に'F107 temperature'を検索した時の結果で取得された室温の値が表示されている。

また、被験者が検索にかかった時間は表 4 に示す通り自然言語を用いた時の方が速かった。

表 3 データベースが送信までにかけた時間

question	time(ms)
koichi iPhone battery	341
F107 temperature	371
battery	58
what is humidity ?	53

表 4 被験者が検索にかけた時間

自然言語	36 秒
SPARQL	404 秒

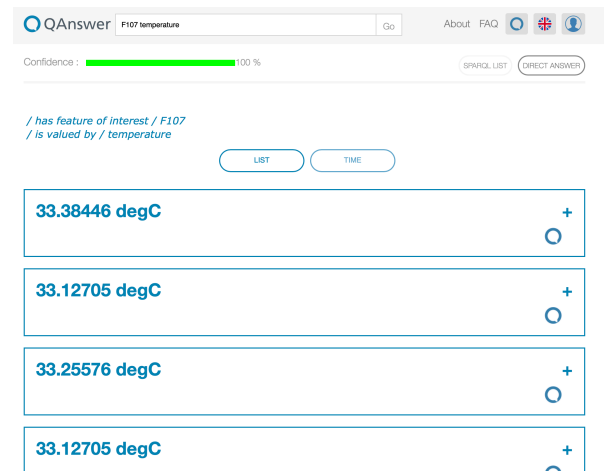


図 6 検索結果

5. むすび

セマンティック技術を用いたリアルタイムセンシングデータ検索システムの試作を行った。結果、HTTP サーバ、データベース、QAnswer を通じてセンサデータの検索を行うことに成功した。データの検索速度や精度はデータ数やデータの種類が増えれば落ちることも考えられるのでセンサの数や計測時間を増やしても検証する必要があると考えられる。

今後は、より多くのセンサが統合されたときのアーキテクチャの動作を分析したい。また、代替となるアーキテクチャを探したい。一つの選択肢として標準化されることに近い Thing Description (reference) ontology を使用することが考えられる。

センサデータを表現するためにセマンティックウェブ技術を使用するという考え自体はもとより存在したが、私たちが知る限り、セマンティック層を通して自然言語を使用してセンサによって生成されたデータにアクセスするアーキテクチャを初めて提案した。このアーキテクチャにより、SSN オントロジーを使用して生成されたデータを調整するだけで、新しいセンサを統合することができるように

なった。

活用方法として、工場や店舗などでこのシステムを活用できることが想定できる。また、スマートフォンでのデータの送信を活用することで参加型センシングによりデータセットを作成することもできると考えられる。

参考文献

- [1] V. E. Angouridakis and T. J. Makrogiannis. The discomfort-index in thessaloniki, greece. 26:53–59, 1982.
- [2] Haller Armin, Janowicz Krzysztof, Cox Simon, Danh Le Phuoc, Taylor Kerry, and Lefrancois Maxime. Semantic sensor network ontology. 2017.
- [3] David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. Rdf 1.1 turtle. *World Wide Web Consortium*, 2014.
- [4] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, et al. The ssn ontology of the w3c semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, 17:25–32, 2012.
- [5] Dennis Diefenbach, Pedro Henrique Migliatti, Omar Qawasmeh, Vincent Lully, Kamal Singh, and Pierre Maret. Qanswer: A question answering prototype bridging the gap between a considerable part of the lod cloud and end-users. In *The World Wide Web Conference*, pages 3507–3510. ACM, 2019.
- [6] Dennis Diefenbach, Kamal Singh, and Pierre Maret. On the scalability of the qa system wdaqua-core1. pages 76–81, 2018.
- [7] Antonio J Jara, Alex C Olivieri, Yann Bocchi, Markus Jung, Wolfgang Kastner, and Antonio F Skarmeta. Semantic web of things: an analysis of the application semantics for the iot moving towards the iot convergence. *International Journal of Web and Grid Services*, 10(2-3):244–272, 2014.
- [8] Apache Jena. Fuseki: serving rdf data over http, 2014.
- [9] Liangcun Jiang, Werner Kuhn, and Peng Yue. An interoperable approach for sensor web provenance. In *2017 6th International Conference on Agro-Geoinformatics*, pages 1–6. IEEE, 2017.
- [10] Edward A Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 2008.
- [11] Maroua Masmoudi, Sana Ben Abdallah Ben Lamine, Hajer Baazaoui Zghal, Mohamed Hedi Karray, and Bernard Archimède. An ontology-based monitoring system for multi-source environmental observations. *Procedia Computer Science*, 126:1865–1874, 2018.
- [12] Eric Miller. An introduction to the resource description framework. *Bulletin of the American Society for Information Science and Technology*, 25(1):15–19, 1998.
- [13] Oscar Hernandez Uribe, Matilde Santos, Maria C Garcia-Alegre, and Domingo Guinea. A context-awareness architecture for managing thermal energy in an nzeb building. In *2015 IEEE First International Smart Cities Conference (ISC2)*, pages 1–6. IEEE, 2015.
- [14] 総務省. 情報通信白書平成 30 年版. 2018.