OfficeGraph: A Knowledge Graph of Office Building IoT Measurements

Roderick van der Weerdt[1 [0000−0002−1125−1126]], Victor de Boer[1 [0000−0001−9079−039X]], Ronald Siebes[1 [0000−0001−8772−7904]], Ronnie Groenewold[2], and Frank van Harmelen[1 [0000−0002−7913−0048]]

[1] Vrije Universiteit Amsterdam
De Boelelaan 1105, 1081 HV, Amsterdam, The Netherlands
{r.p.vander.weerdt,v.de.boer,r.m.siebes,frank.van.harmelen}@vu.nl
[2] VolkerWessels iCity
Torenallee 20, 6003 5617BC, Eindhoven, The Netherlands
RGroenewold@volkerwessels.com

Abstract. In order to support the global energy transition, smart building management provides opportunities to increase efficiency and comfort. In practice, real-world smart buildings make use of combinations of heterogeneous IoT devices, and a need for (knowledge graph-enabled) interoperability solutions has been established. While ontologies and synthetic datasets are available, a real-world, large scale and diverse knowledge graph has so far not been available. In this paper, we present OfficeGraph, a knowledge graph expressed in the SAREF ontology containing over 14 million sensor measurements from 444 heterogeneous devices, collected over a period of 11 months, in a seven story office building. We describe the procedure of mapping original sensor measurements to RDF and how links to external linked data are established. We describe the resulting knowledge graph consisting of 90 Million RDF triples, and its structural and semantic features. Several use cases are shown of the knowledge graph: a) through various realistic data analysis use cases based on competencies identified by building managers and b) through an existing machine learning experiment where we replace the original dataset with OfficeGraph.

Keywords: Knowledge Graph · Dataset · IoT · SAREF · Sensors.

1 Introduction

Due to the increasing awareness of climate change, in combination with the rising energy costs, more interest is being shown towards sustainability and efficiency of energy usages. One area where sustainability measures can be especially effective is in office building management, where efficiency gains for large buildings have large impact, as compared to dealing with one house at a time [7].

To increase the efficiency of office building management, we can use data produced in such smart office buildings. IoT sensors have become prevalent in office buildings. Measurements from those sensors can be examined to determine

possible sustainability improvements, such as by training a machine learning system to discover behavioral and systematic patterns of occupation density during the week, CO2 levels and heating values [10].

Open datasets facilitate research opportunities and experiments. Unfortunately, large and heterogeneous datasets of sensor data from office buildings are rarely available publicly. If they are available, they are either small in size or only related to one type of measurement (for example only energy consumption [9] or movement [14]). Some experiments reported in literature are performed on proprietary datasets, restricting evaluation of systems and comparing research.

As smart office buildings typically have a multitude of sensor types, realistic open datasets will contain heterogeneous data since different sensors make different types of measurements (such as temperature, or occupancy). Additionally, a successful data model will need to be able to deal with a varied and sometimes inconsistent use of time intervals. Finally, internal links and links to external data can increase usefulness of said data. Therefore, making office building IoT sensor data available as a knowledge graph addresses these requirements. Ontologies, such as SAREF [3] are already available to represent measurements as knowledge graphs. In summation, the dataset should be a large, heterogenous, and open knowledge graph.

In this paper, we present **OfficeGraph**, a large, real world knowledge graph containing measurements taken by 444 IoT devices, over 11 months, in a seven story office building. The devices are made up of 17 different sensor models, which make measurements of many different *properties*[3]. We first discuss relevant related work, further motivating the resource presented here (Section 2). We then describe the original data and the process of converting it to a knowledge graph (Section 3). The results and ways of accessing the knowledge graph are described in Section 4. In Section 5, we demonstrate the usefulness of OfficeGraph through two realistic data analysis use cases provided by building stakeholders. There, we also define and execute a machine learning experiment on the knowledge graph.

2 Related Work

This section describes related research by providing examples of what currently available IoT datasets look like, and what kind of experiments are performed with such datasets. We compare the datasets on the three requirements defined in the introduction: size, heterogeneity and openness.

Arz von Straussenburg et al. created a dataset containing (boolean) measurements of detected movement at desks in an office space, in order to create a desk sharing space [14]. In the experiment the motion sensor measurements were used to classify a desk as occupied, creating a more accurate desk sharing platform for the office occupants. Data was recorded for eleven days. Although the dataset used is open, the measurements are not heterogeneous (only motion)

[3] In order to avoid confusion between RDF properties and the property being measured, we will specifically refer to the former as RDF properties, and address the latter as properties.

and it is not large. Motion is also measured in OfficeGraph, and contact sensors show which doors are opened. The experiment from Arz von Straussenburg et al. can be repeated with OfficeGraph by classifying an office as occupied when the door was already opened that day.

Rafsanjani et al. also created a dataset of IoT device data measured in an office environment [10], in order to learn the "energy-use behaviors of the occupants". Occupancy data was used to determine which occupants where using devices in which rooms, to detect who was using what. Data was recorded over a six week period, however the data was not made public. Although the data used in the experiment is heterogeneous, it is not large and not open. OfficeGraph includes occupancy data as well, however since it only includes sensor measurements there is no energy consumption for devices that are used by occupants. Adjustments can be made to the experiment, by changing the energy consumption of devices into the thermostat settings of users, which can yield similar patterns as the original experiment.

The previous two experiments both required one or two datatypes (movement or occupancy and energy consumption), but more heterogeneous datasets have also already been created. Heo et al. created a dataset of 26 different devices, with data recorded for 144 hours [5]. They compare various data acquisition scheduling methods for the devices with the goal of keeping the required data collection goals but minimizing the traffic needed for gather it. By scheduling the data acquisition fewer data inquiries need to be performed, therefor saving energy. The dataset has been made available online as downloadable matlab files. Although the data used in the experiment is open and heterogenous, it is not large. Since OfficeGraph also contains heterogeneous data, it can similarly be used to setup a testbed with devices. This would also allow for a longer experiment, because OfficeGraph contains a longer period of measurements.

OfficeGraph is an IoT graph with measurements data from the IoT devices, but there are also different kinds of IoT datasets. An example of such a dataset is created by Ren et al. [11]. This *traffic* dataset does not focus on the measurements made by the IoT devices, but instead it focuses on the traffic that is communicated between the devices. The dataset does not contain the measurements made by the devices, but it contains the "packet headers" of the messages send by 81 devices. Besides recording the measurements being taken by devices for 112 hours, this dataset also contains the results from "34,586 experiments" with the communications between the devices, linking different parameter settings to different behaviors from the devices.

With traffic IoT device datasets experiments are performed concerning various aspects like privacy and profiling based on device behavior [2]. Event though this traffic IoT device dataset is available online, it contains different information from the IoT devices then what what is collected with OfficeGraph, which puts the emphasis on the measurements made by the IoT devices. Throughout this paper when we refer to an IoT dataset or knowledge graph this will be a IoT measurements dataset or knowledge graph.

A larger dataset that is available online is the OPSD Household data dataset [9]. It contains almost five years of energy consumption information from 68 devices spread out over 11 households in a city in southern Germany. The measurements from the residential households have been used to create a knowledge graph of 36 different devices [16]. However all the information it contains about the devices is energy consumption, limiting the applications of the dataset. Experiments with OPSD that investigate the impact of semantic enrichment on machine learning performance are described in [17]. There, two version of the same knowledge graph are compared, with and without the enrichment. In Section 5.2, we describe how we replicate this same experimental setting with OfficeGraph to demonstrate the use of the resource.

3 Converting the Source Data

In order to create the OfficeGraph we construct a pipeline that maps a collection of JSON files to RDF based on the SAREF ontology and perform various enrichments. By making changes to the mapping template described in this section the pipeline is reusable for JSON data from different devices. We did not use any general knowledge graph creation method, such as RML, because due to the size of the dataset general knowledge graph creation methods would require a lot of memory and execution time [6]. The Python scripts used to perform the mapping process can be found on GitHub[4] and were in part based on [17].

3.1 Source Data

The original collection of measurements consists of separate JSON files, one file for each device, stored in three folders which separate the devices based on manufacturer: Airwits, Calumino and Samsung. There are 19 different models in total, Airwits and Calumino both have one type of device model, with the remaining 17 device models being Samsung devices.

All devices are located in a seven story office building in the Dutch city of Eindhoven. Over 200 different companies make use of the building with an average of around 250 people working in the offices. The data was initially collected as part of the InterConnect project[5], which has as its goal to use semantic technologies to facilitate connections between smart devices. By consistently modeling the data in the shared SAREF ontology, any device only needs to map it once, instead of having to translate it to a bi-lateral format for each receiving device.

3.2 Original Data Structure

Each JSON file consists of measurement data originating from one sensor. The objects in the JSON files containing the measurements use a total of 40 different

[4] https://github.com/RoderickvanderWeerdt/OfficeGraph
[5] https://interconnectproject.eu

Table 1: Three examples of mapping templates used to create the Officegraph. Bolded variables represent the data from the JSON file, italicized variables are defined elsewhere in the mapping and regular font variables are created for this specific template.

JSON Header	Relevant mapping template
"device_model_description"	<*device*> saref:hasModel <**modeltype**>.
"data_room"	<*device*> s4bldg:isContainedIn <**data_room**>. <**data_room**> s4bldg:contains <*device*>. <**data_room**> rdf:type s4bldg:BuildingSpace.
"data_temp_c"	<*device*> saref:measuresProperty <*property_uri*>. <*device*> saref:makesMeasurement <meas_uri>. <meas_uri> rdf:type saref:Measurement. <meas_uri> saref:hasvalue <**value**>. <meas_uri> saref:hasTimestamp <*timestamp*>. <meas_uri> saref:isMeasuredIn om:degreeCelsius. <meas_uri> saref:relatesToProperty <*property_uri*>. <*property_uri*> rdf:type saref:Temperature.

headers. Of these, 24 were identified as containing relevant information for the OfficeGraph, by discussing with experts from the domain to determine which headers contained duplicate information, and by excluding headers that were always left empty. For each of the 24 headers, mapping rules were created that produce triples that capture the data, and structure it correctly in the graph. Three examples of such mapping rules can be seen in Table 1.

3.3 Data Model & Mapping Template

Multiple ontology standards exist for smart device information, such as SSN [1] and WoT Thing Descriptions [15]. OfficeGraph is expressed in SAREF [3], a domain standard ontology specifically created to model measurements of different IoT devices. A comparison and mapping between SAREF and SSN is made in [8].

The main structure we use from SAREF can be seen in Figure 1, for each individual device the "device template" creates triples for all consistent information about the device, such as the device type and model. For each individual measurement the "measurement template" creates triples to describe the measurement, its value, unit of measurement and timestamp. The device instance and measurement instances are connected in two ways, directly through the `saref:makesMeasurement` property, and indirectly through the `saref:Property` instance. The latter describes what has been measured, such as temperature or humidity.

In addition to the SAREF ontology we use two of its extensions. SAREF4BLDG[6], which provides classes used to describe the relation between devices and rooms,

[6] https://saref.etsi.org/saref4bldg/

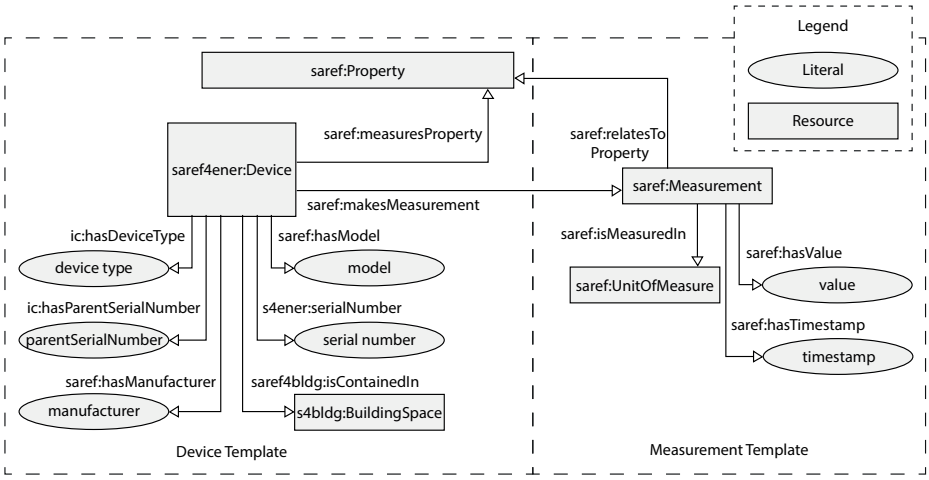


Fig. 1: Visualization of the templates used to create OfficeGraph.

and between rooms and buildings. The other extension is SAREF4ENER[7], which provides additional classes for information about the device. As suggested in the SAREF documentation we use the OM1.8 ontology to represent the units of measure of the measurements [12].

Additionally, the following new instance and six new subclasses of SAREF classes, and two new RDF properties are introduced to enable a more detailed representation of the building data:

new instance of `saref:UnitOfMeasure`:

- `ic:people` is used in the measurement of a number of people, such as a doorcounter which counts the number of passing people.

new subclasses of `saref:Property`:

- `ic:RunningTime` is used in the measurement of time that has passed, such as the time since the last movement was detected.
- `ic:Contact` is a property related to whether or not a sensor is making contact (closed window).
- `ic:BatteryLevel` is a property related to the current percentage of charge left in the device's battery level.
- `ic:CO2Level` is a property related to the current CO_2 level measured by the device.
- `ic:DeviceStatus` is a property related to current status of the device, whether it is active or not.
- `ic:thermostatHeatingSetpoint` is a property related to the current heating setpoint of a thermostat.

new RDF properties to store different information relevant to the device:

[7] https://saref.etsi.org/saref4ener/

- `ic:hasParentSerialNumber` is a property of a device, to store the Parent Serial Number, which is related to the edge device it is connected to.
- `ic:hasDeviceType` is a attribute of the device that represents the devicetype of the device.

3.4 Enrichment

In addition to OfficeGraph we also create three enrichments, that can be used in combination with OfficeGraph.

Devices in Room Not all devices recorded the name of the room in which they were located in the original data. Additional information was retrieved from the office building to add triples for most devices in which room they were located, and in which "service zone" they are located (which is used for maintenance on the devices). Furthermore, we added for each room and service zone on which floor they were located and that each floor is part of the same building.

Wikidata days OfficeGraph is linked to the Wikidata graph, by matching the timestamps to the Wikidata concepts of their corresponding dates. This allows federated queries to be performed, combining information from OfficeGraph and Wikidata. For example, we can query the graphs to determine on which day a measurement is taken.

Graph Learning Enrichment Previous research has shown that certain semantic enrichments to a knowledge graph can be beneficial for the machine learning on a graph process [17]. These same enrichments are made available for OfficeGraph, in separate files. The enrichments are:

- *Sequence links*, RDF properties that link to the previous and next measurement, taken chronologically.
- *Rounded values*, URI entities of the measurement values, rounded to function as a bucket for all similar values.
- *Timestamp buckets*, This enrichment is slightly different compared to the original semantic enrichment. Where the original was only a URI entity of the timestamp, this time it also serves as a bucket to collect all measurements taken within the same hour.

The results of the mapping process will be discussed in the next section.

4 Description of OfficeGraph

The OfficeGraph consists of 89,599,577 triples describing 14,930,478 measurements measured by 444 devices. Measurements were taken over a period of 11 months, starting March 1st 2022 and ending January 31st 2023. The resulting turtle files have a uncompressed size of 4.5 GB.

The measurements are represented with 11 different properties, which are all shown in Table 2. Additionally, the table shows which properties are measured by which device model and what the distribution of models is for the devices.

Table 2: Distribution of the models from the devices, and properties measured by those device models.

	Device property type											
Device model	ic:BatteryLevel	ic:CO2Level	ic:Contact	ic:DeviceStatus	ic:RunningTime	ic:thermostatHeating...	saref:Humidity	saref:Motion	saref:Occupancy	saref:Power	saref:Temperature	**Number of devices**
Aeon Home Energy Meter				1								1
Fibaro Smoke Sensor	3			3							1	3
Hex Doorcounter Ver 1.0					14				14			14
Qubino Dimmer				1								1
R5		227					227				227	227
SmartPower Outlet				20						13		20
SmartSense Button	3			3							1	3
SmartSense Moisture Sensor	8			8							1	8
SmartSense Motion Sensor	36			36				24			24	36
SmartSense Multi Sensor	93		51	93							62	95
Z-Wave Basic Smoke Alarm	2			2								2
Z-Wave Door/Window Sensor	6		2	6								6
Z-Wave Metering Switch				2								2
Z-Wave Radiator Thermostat	1			1								1
Z-Wave Range Extender				1								1
Z-Wave Switch Secure				4								4
Z-Wave Temp/Light Sensor	2			2								2
Z-Wave Water/Temp/Light Sensor	4			4							2	4
Zigbee Thermostat	14			14		11					11	14
Totals	172	227	53	201	14	11	227	24	14	13	329	444

4.1 Timepoints

The number of timepoints at which measurements were taken differs greatly between devices, as can be seen in Figure 2. We identified two causes of this difference: (1) some devices take measurements every time a change in the measurement is detected (Samsung and Calumino devices). Which also means devices detecting many changes will generate more measurements than the devices detecting fewer changes, for example a people counter in a busy hallway will make more measurements then a door sensor in a one person office. Other devices (Airwits devices) will make measurements at a given interval, producing a

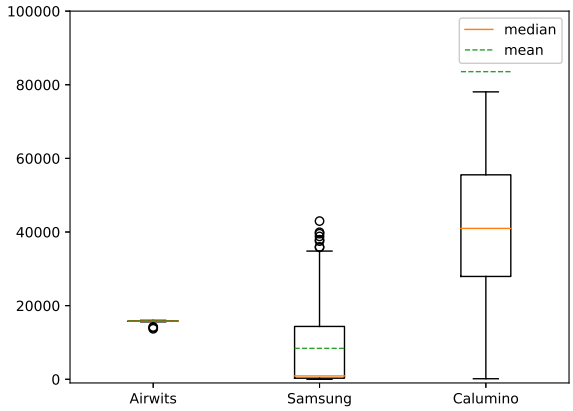


Fig. 2: Boxplots of the number of points in time in which a device has made a measurement. Different update settings result in different distributions of timepoints. One Calumino device outlier is not shown for readability, because it made measurements at 600,000 points in time.

very similar amount of timepoints. (2) Some devices were turned off, or had a low battery, for a period of time, resulting in fewer overall measurements.

4.2 Graph Structure Metrics

The indegree and outdegree of a knowledge graph provide information concerning the amount of RDF properties of each entities. The outdegree is the number of RDF properties an entity has, the indegree of an entity is the number of entities it is an object of. The indegree and outdegree of a graph can have big effects on algorithms that use graph traversal, such as RDF2Vec [13].

In Figure 3 we see the indegree and outdegree of OfficeGraph. To compare it with other large knowledge graphs we use the in and outdegree as recorded by Duan et al. [4], where the authors describe multiple characteristics about large knowledge graphs, such as DBpedia or Barton. The average indegree (5.6) and outdegree (6.0) of OfficeGraph is similar to the other knowledge graphs, as is the distribution of the indegree of the entities. However, when we compare the number of entities with an outdegree higher then 10^4, the other knowledge graphs only have two entities with such a high outdegree, while OfficeGraph has hundreds. These high outdegree entities represent the devices and the high outdegree is due to the high number of measurements related to the devices.

4.3 Enrichment

Devices in Room For 340 devices we are able to add additional room information, which results in a graph containing 2,426 triples.

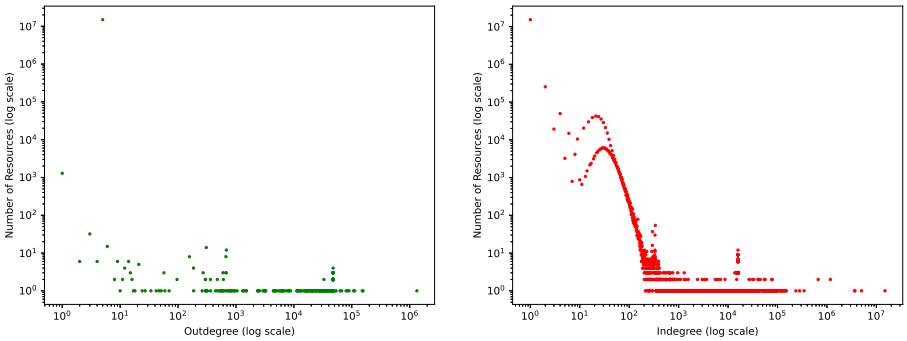


Fig. 3: The indegree and outdegree of entities in OfficeGraph, the axis use logarithmic scale.

Wikidata days The Wikidata days enrichment adds 8,088 triples, one triple for every hour in the OfficeGraph. We chose to make the links to Wikidata only for the timebuckets in the graph learning enrichment, since the timestamps in OfficeGraph are literals.

Graph Learning Enrichment When the enrichment is performed with all devices the graph learning enrichment adds a combined total of 89,581,980 triples. Six new RDF properties are added for each measurement in the OfficeGraph, except for each first and last (chronological) measurement of each device, because those do not have previous or next measurements to link to.

4.4 Accessing the KG

OfficeGraph is accessible under the Creative Commons Attribution 4.0 International license, in three ways: as RDF files on GitHub, a snapshot on Zenodo and through a SPARQL endpoint.

RDF files on GitHub A zipped version of OfficeGraph is available on GitHub[8]. Instead of one file containing the entire knowledge graph the zipped folder contains a separate file for each individual device. Each file contains all the measurements made by the device. The *devices in room* enrichment is included in a separate file.

Zenodo snapshot The same zip file that is available on Github is also made available on Zenodo at: https://zenodo.org/records/10245815.

SPARQL endpoint A Cliopatria [18] server has been set up at https://data.interconnect.labs.vu.nl to store the data, and expose it through a SPARQL endpoint. SPARQL queries can be used to retrieve information from the graph. The *devices in room* enrichment is included in the datastore.

[8] https://github.com/RoderickvanderWeerdt/OfficeGraph

5 Using OfficeGraph

In this section, we demonstrate the usefulness of OfficeGraph through two realistic use cases: 1) through a data analysis task for building management and 2) by performing a machine learning experiment with the OfficeGraph.

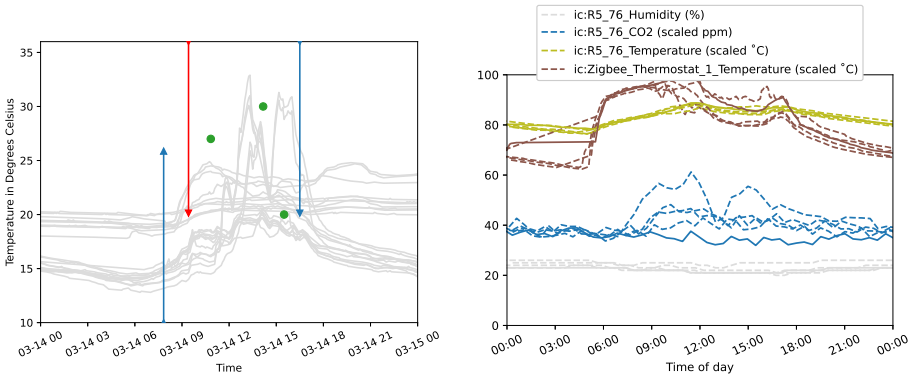
5.1 Building Management Data Analytics

The OfficeGraph can be used to highlight situations where automatization can be of use. By showing that certain situations occur it can be used as an argument for why an automatization can be beneficial. Two competency questions were created in collaboration with the building owners, to assure they are relevant for the office building.

Thermostat and Window status The first question relates to occupant behavior: "Is the thermostat turned down when the windows are opened in that same room?". To answer this question we queried the graph for contact sensors, thermostat settings and temperature values, from devices that are located in the same area. The results for one day in one room of these queries are visualized in Figure 4a, the temperature (grey lines) is measured by multiple devices, each differently colored arrow is a specific window opening (arrow up) and closing (arrow down), and the dots are the thermostat temperature settings. We consider "turning the thermostat down when the windows are opened" to have occurred when the thermostat is turned to a lower value then the current temperature within 30 minutes after a windows has opened. Using the results from the queries, we can conclude that the answer to the competency question is: no, because the thermostat is never lowered (within 30 minutes) in an office when an window is opened.

Occupation and office climate The second question relates to the office climate: "Is there a noticeable effect of occupants on the climate of office rooms?". We answer this question by querying the graph for humidity, CO_2 and temperature measurements, over five days: four weekdays and a Saturday. The results for one office are presented in Figure 4b. The values have been scaled 0-100 to fit in one figure. From the figure we see that in the morning all temperature measurements consistently rise, and lower in the evening, regardless of which day it is. The humidity is not effected by time of day, nor by which day of the week it is. However, the CO_2 values only rise in the morning during the workdays, and stay (relatively) constant during the weekend. Therefor we can answer the question: yes, the office climate is affected by occupants, specifically the CO_2 values.

Jupyter notebooks are available[8] that show the code used to create the figures and answer the competency questions.



(a) Windows opening (arrow up) and closing (arrow down) with temperature and thermostat setpoints over one day

(b) Scaled measurement changes over four day in one room, dashed lines are workdays, solid line is Saturday

Fig. 4: Plots of measurements from OfficeGraph.

5.2 Machine Learning on OfficeGraph

In this section we demonstrate how OfficeGraph can be used for machine learning experiments. We perform experiments re-using the learning task and learning approach from [17] on OfficeGraph. The goal of the original experiment was to compare the effect of different IoT knowledge graph enrichments on the effectiveness of the embedding method (RDF2Vec) and representativeness of the resulting embeddings.

Experimental setup Figure 5 depicts the pipeline used in the experiment. The goal of the experiment is to examine the effect of semantic enrichment on the quality of embeddings learned from a knowledge graph. In Step 1 of the pipeline we have two knowledge graphs, one without the enrichment (Basic Graph) and one with the enrichment (Enriched Graph). The enrichment is described in Section 3.4. In Step 2 of the pipeline we train a model to learn embedding representations for both knowledge graphs. Step 3 uses the embeddings to train two classification models. By comparing the accuracy of the classifications we determine whether the semantic enrichment had a (positive) effect on the quality of the embeddings.

The classifier in the original experiment predicted whether the outside temperature at a point in time was warm or cold. This label was created by sorting the timestamps from high to low based on the outside temperature, and labeling the first 50% as warm, and the last 50% as cold. This time the results of the classification model will be based on how well it predicts whether or not a given

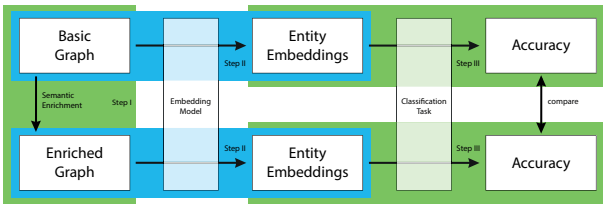


Fig. 5: Experimental pipeline. Original image taken from [17].

point in time is a working day (Monday-Friday) between working hours (9-5). All code used in this section is made available through GitHub[9].

Pre-processing Before OfficeGraph is ready to be used in the pipeline we perform some pre-processing. Only a subset of OfficeGraph is used in order to lower the required compute power to train the embeddings. The subset is created by selecting only the devices located on the 7th floor, which leaves us with 13 devices and a knowledge graph with 3.9 million triples.

We use the graph learning enrichment (as described in 3.4) in order to use the timestamp uris that bucket the timestamps and allow us to use these uris as entities as input for the pipeline.

We create the *entity file*, the document containing the entities in the graph we want to create an embedding for, and its prediction target (a class or value) by querying the graph.

Classification Task The classification task we use for the experiment predicts whether a timestamp bucket occurs within working hours. To create the entity file we query the graph to retrieve all timestamp buckets, and use a script to classify each timebucket as occurring within workhours if the day is a weekday, and the hour is between 9 and 17.

Implementation Since the goal of the experiment is not to get cutting edge results, but to perform the experiment to demonstrate the usability of Office-Graph we use the same preset hyperparameters that the original experiment used. Which means that the RDF2Vec embeddings are made with 25 random walks of length 2. And the classifier is a MLP with one hidden layer of 512 ReLU nodes. The original paper provides additional information [17].

The labels used in the original experiment were made by splitting the dataset in two even halves of warm and cold outside temperatures, therefor when the classifier would exclusively predict one of the classes, the accuracy would be 50%. This is used as a baseline to determine whether the classifier is learning something from the embeddings. For the new experiment the baseline is similarly always predicting the most prevalent class, which in this case is predicting that every hour is not a working hour. This results in a baseline accuracy of 76%.

[9] https://github.com/RoderickvanderWeerdt/semantic-enrichment-of-IoT-graphs

Table 3: Results of the experiment with OfficeGraph, compared with the results of the original experiment with OPSD.

dataset	baseline	basic graph	enriched graph
Classification (accuracy)			
OPSD[17]	50%	49.8%	**80.7%**
OfficeGraph	76.1%	73.0%	**85.1%**
Value Prediction (MAE)			
OPSD[10]	6.3	6.6	**6.1**
OfficeGraph	2.03	2.01	**1.87**

Results The results of the experiment can be seen in Table 3. Both the original experiment and the new experiment display the same behavior, the basic graphs score close to the baseline and the enriched graphs score significantly above it.

Value Prediction Task The experiment can also be performed with a different machine learning task, such as value prediction[10]. For this we replace the accuracy comparison with a mean absolute error (MAE) comparison to evaluate the resulting prediction from the MLP.

The entity file for the value prediction task is created by querying the graph based on the property we want to predict and return all timestamp URIs and values pairs. In the situation where multiple measurements were taken in the timespan of one timestamp bucket, we take the average of those measurements.

Implementation The MLP structure from the classification experiment is reused, but with a mean squared error loss function instead of the cross entropy loss function. The other change is the evaluation metric. MAE is the average difference between the prediction and the target value. Therefore, when using accuracy a higher value represents a more similar prediction, but with MAE a lower value represents a more similar prediction instead.

In the original experiment the outside temperature was predicted, here we predict the temperature measurements of one specific device. The outside temperature has a bigger range of measurements, with a minimum temperature of -11°C, maximum temperature of 28°C and standard deviation of 7.5, compared to a minimum temperature of 15.5°C, maximum temperature of 27°C and standard deviation of 2.4 with the new experiments. Therefor the results are expected to differ more then with the classification experiments, however, we still expect the overall behavior, where the model trained with the enriched outperforms the model trained with the basic graph, to occur.

As a baseline we use the average temperature over the entire dataset as the predicted value. This was the same baseline used in the original experiment.

[10] OPSD results for the value prediction task were not part of the experiments described in [17], but are presented here to compare results.

Results Table 3 shows the results from the value prediction experiment. As with the classification experiment we see that the experiment with the enriched graph outperforms the baseline and basic graph. Because the difference between the MAE results are closer then the classifier results we report the results of a significance test. We performed a t-test which showed that the results from the enriched graph are significantly different from the results with the basic graph ($p < 0.002$). The results with the baseline are not significantly different from the results with the basic graph ($p > 0.5$).

When we compare the results of the new experiment with the original experiment (which used the OPSD dataset) we see that in both cases the enriched graph provides the most similar predictions. The MAE is in all cases much higher for the original experiment, as was expected, due to the bigger variance and range of the outside temperature that is predicted in the original experiment.

6 Conclusion

In this paper we presented OfficeGraph, a knowledge graph containing 11 months of heterogeneous measurements from 444 IoT devices. We described the mapping process, how it applies to this dataset, and made the code available to be adapted and reused for other datasets.

Specifications of OfficeGraph are provided in terms of specific traits: 1) the properties measured by each device, 2) time points: the amount of times a devices records a measurement, 3) outdegree: the high outdegree of the device entities, 4) specific enrichments that can be added to core data of OfficeGraph.

OfficeGraph is accessible in three ways: downloadable via Github or Zenodo, and it can be queried through a SPARQL endpoint.

In order to demonstrate the usability of OfficeGraph, we described how it can be used with python (notebooks), with SPARQL queries and with machine learning experiments.

OfficeGraph is a benchmark set that can be used for future office data experiments, allowing for more representative experiments for sustainability and efficiency of energy usages.

Acknowledgements. This work is part of the InterConnect project (interconnectproject.eu/) which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 857237.

References

1. Compton, M., Barnaghi, P., Bermudez, L., GarcíA-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al.: The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. Journal of Web Semantics **17**, 25–32 (2012)

2. Dadkhah, S., Mahdikhani, H., Danso, P.K., Zohourian, A., Truong, K.A., Ghorbani, A.A.: Towards the development of a realistic multidimensional iot profiling dataset. In: 2022 19th Annual International Conference on Privacy, Security & Trust (PST). pp. 1–11 (2022)
3. Daniele, L., den Hartog, F., Roes, J.: Created in Close Interaction with the Industry: the Smart Appliances REFerence (SAREF) Ontology. In: International Workshop Formal Ontologies Meet Industries. pp. 100–112. Springer (2015)
4. Duan, S., Kementsietsidis, A., Srinivas, K., Udrea, O.: Apples and oranges: a comparison of rdf benchmarks and real rdf datasets. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. pp. 145–156 (2011)
5. Heo, S., Song, S., Kim, B., Kim, H.: Sharing-aware data acquisition scheduling for multiple rules in the iot. In: 2020 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). pp. 43–55. IEEE (2020)
6. Iglesias, E., Jozashoori, S., Vidal, M.E.: Scaling up knowledge graph creation to large and heterogeneous data sources. Journal of Web Semantics **75**, 100755 (2023)
7. Jafarpur, P., Berardi, U.: Effects of climate changes on building energy demand and thermal comfort in canadian office buildings adopting different temperature setpoints. Journal of Building Engineering **42**, 102725 (2021)
8. Moreira, J., Daniele, L., Pires, L.F., van Sinderen, M., Wasielewska, K., Szmeja, P., Pawlowski, W., Ganzha, M., Paprzycki, M.: Towards iot platforms' integration semantic translations between w3c ssn and etsi saref. In: SEMANTICS workshops (2017)
9. Open Power System Data: Data Package Household Data. Version 2020-04-15 https://data.open-power-system-data.org/household_data/2020-04-15/. (Primary data from various sources, for a complete list see URL). (2020)
10. Rafsanjani, H.N., Ghahramani, A.: Towards utilizing internet of things (iot) devices for understanding individual occupants' energy usage of personal and shared appliances in office buildings. Journal of Building Engineering **27**, 100948 (2020)
11. Ren, J., Dubois, D.J., Choffnes, D., Mandalari, A.M., Kolcun, R., Haddadi, H.: Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In: Proceedings of the Internet Measurement Conference. pp. 267–279 (2019)
12. Rijgersberg, H., Van Assem, M., Top, J.: Ontology of units of measure and related concepts. Semantic Web **4**(1), 3–13 (2013)
13. Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., Paulheim, H.: Rdf2vec: Rdf graph embeddings and their applications. Semantic Web **10**(4), 721–752 (2019)
14. Arz von Straussenburg, A.F., Blazevic, M., Riehle, D.M.: Measuring the actual office workspace utilization in a desk sharing environment based on iot sensors. In: International Conference on Design Science Research in Information Systems and Technology. pp. 69–83. Springer (2023)
15. W3C: Web of Things (WoT) Thing Description (2020), https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/
16. van der Weerdt, R., de Boer, V., Daniele, L., Nouwt, B., Siebes, R.: Making heterogeneous smart home data interoperable with the SAREF ontology. Int. J. of Metadata, Semantics and Ontologies **15**(4), 280–293 (2021)
17. van der Weerdt, R., de Boer, V., Daniele, L., Siebes, R., van Harmelen, F.: Evaluating the effect of semantic enrichment on entity embeddings of iot knowledge graphs. Proceedings of the 1st International Workshop on Semantic Web on Constrained Things at ESWC 2023 **3412** (2023)
18. Wielemaker, J., Beek, W., Hildebrand, M., Van Ossenbruggen, J.: Cliopatria: a swi-prolog infrastructure for the semantic web. Semantic Web **7**(5), 529–541 (2016)