W3C LBD Community Group
Minutes - Call 2021-09-21

Attendees:

- Karl Hammar
- Mathias Bonduel
- Philipp Hagedorn (RUB, Germany)
- Joel Bender (Cornell University)
- Shams Ghazy (University of Nottingham, Malaysia)
- Katja Breitenfelder (Fraunhofer IBP)
- Christian Kreyenschmidt (Jade University)
- Conor Shaw
- Hazar Karadag
- Jing Ying Wong
- Jude Lubega
- Kevin Luwemba Mugumya
- Madhumitha Senthilvel (RWTH Aachen)
- Shams Khaled Elhosseny Ghazy
- Jyrki Oraskari (RWTH Aachen)
- Allan Mazimwe
- Lusubilo Singogo
- Mugumya Vincent

If you are here: see https://teams.microsoft.com/l/meetup-join/19%3ameeting_NGVkNjkyMWQtYWY1My00YjhmLTg5NmMtOWMxNWI5Y2M3YzM1%40thread.v2/0?context=%7b%22Tid%22%3a%227564bc8f-3738-4b4d-bd57-5a02ca6215fb%22%2c%22Oid%22%3a%2245b184f4-44dd-4f9c-99af-f791654418cd%22%7d instead of earlier communicated link, which appears inaccessible.

Presentation slides

Link to the slides on:
- Google Drive
- Github

Date and time

- 2021-09-21, Tuesday, 15:00-16:30@UTC/ 17:00-18:30@CEST/ 08:00-09:30@PST

Moderators

1. Karl Hammar

Agenda

1. Introduction of new/returning members
2. Presentation: Kevin Luwemba, *Relational learning on Linked Building Data for Reinforcement Learning-based building contro*l
3. Q&A

Minutes

1. Introduction of new/returning members
    a. Jude Lubega: colleague (?) of Kevin, located in Uganda. Computer scientists PhD (Reading University)
    b. Hazar Karadag: Master student at RWTH Aachen in construction robotics with Jacob Beetz. Want to learn more about Linked Data in the construction industry
2. Presentation: Kevin Luwemba, *Relational learning on Linked Building Data for Reinforcement Learning-based building contro*l
    a. Breakdown of the research context: Use case of building control
        i. Linked Building Data: complex buildings, large amount of sensors, various and heterogenous metadata schemes (labelling information related to the building), data processing is departmentalized (similar as building structure) => more standardized and formalized metadata schemas via Linked Building Data (e.g. SAREF, BOT, BRICKS, SSN, SOSA, etc)
        ii. Reinforcement learning: even larger challenges, data is not linked but siloed (makes advanced reasoning difficult), need for collaborative building control (multi-agent) but this requires data and inter-links
            1. Consequently, building agents are not adaptive and underutilize building control information (leading to less performant control actions)
            2. Existing solution: reinforcement learning > some things are very difficult to model.
                a. Two ways to approach your problem:
                    i. Model-based RL
                    ii. Model free RL
                b. experience-driven, sequential-decision making > control problem most often modelled as Markov Decision Process (data from the present is enough to predict the future)
                    i. States: temperature, humidity, etc
                    ii. Actions: increase temperature, cut off the power, charge/stop PV, etc
                    iii. Dynamics: how to change from one state to another (probability function) => choose actions

iv. Reward: feedback of the performance of the agent in the environment (trial-and-error). Function of action and new state
v. Discount factor: deal with uncertainties of forecasting future rewards

iii. Relational learning:
1. augment Linked Data in knowledge graph before going to RL agent
2. RL agent uses knowledge graph for learning.
3. Dealing with relational data is tricky compared to raw data

iv. Learning from relational building information
1. Relationships are not often used in the learning
2. Predict unknown (incomplete) nodes in a data graph => node classification task. Predicts relations to existing, known nodes
3. Link prediction task: find extra relationships
4. Link-based clustering: find communities of nodes in a graph. Can be useful when a certain node breaks a connection (e.g. a sensor breaks and a room needs to be operated still. Take actions from similar room with a sensor)
5. Raw data =1=> structured data =2=> learning algorithm =3=> model
   a. Step 1 is very complicated for graphs (incomplete, open world assumption). Can we skip this first step using feature learning?
      i. Find the features of a node automatically. Node embedding models: local and global features
         1. Anomaly detection
         2. Clustering

v. Whole pipeline
1. Serialize LBD graph from Revit
2. Enter LD in RDFlib > formalize in tensor (~ 3D matrix)
3. Training, validation and testing tensor
4. Iterative process to learn the model, optimized to optimal model (RESCAL-ALS model)
5. Push learned vectors to the agent (instead of raw data)

vi. Large number of existing RL models: translation, bilinear, neural network > tested extensively in biomedical world but not clear if it works the same for Linked Building Data. Studying the peculiarities of the models
1. Neural networks are not optimized for graphs => graph neural networks

vii. Open issues

1. Sparsity of certain nodes: will not learn appropriately. How much sparsity can be tolerated? > requirement can be formalized in SHACL shapes: e.g. each node needs +10 relations
2. Size of the graphs can become large (e.g. aggregated buildings in smart city setting) > also large size of vector needed. Requires optimization
3. SPARQL combined with more abstract query patterns, based on reasoning with RL > embed in SPARQL engine

3. Q&A
   a. [Karl] If the vector is a reduced part of the graph. Does a single vector per node a good idea or do you think multiple vectors depending on the problems are needed? [Kevin] vectors might be biased by the queried sources. Vectors are task-dependent. It's necessary to find the sweet spot in the size of vectors (not too small, not too large). A certain vector might also be reusable for other use cases
   b. [Madhu] changes in (interlinked) data models. What about integrity of certain links? History of previously created links. Does your research take this into account? [Kevin] OPM can capture history inside a knowledge graph. The RL model can try to learn based on OPM. Would need to think about it more. [Madhu] from biomedics domain? [Kevin] always retraining after dumping (a part of) past embeddings. Can be avoided with graph neural networks, reuse knowledge from the history.
   c. [Karl] does the training requires to utilize/study the semantics of modeling languages such as OWL/RDF/RDFS? [Kevin] limited/not used. Also redundant data (e.g. rdfs:label on a node). We abstract the graph
   d. [Mathias] did you used DL reasoning in your workflow, e.g. to expand the graph? [Kevin] not yet considered. What could be the use cases? [Mathias] expanding the graph at the start of the pipeline, before doing the training. Or to check for inconsistencies when you want to materialize triples back in the data graph at the end of the pipeline. For the latter, SHACL can also be useful in certain cases.

Next Call

TBD - a call with multiple short pitches (10-15min) by researchers who want to share their latest findings, ideas, challenges, etc.

> We are interested in getting suggestions from the community about potential agenda items and **Elevator Pitches** for the following calls. Please send your suggestions to the chairs or to internal-lbd@w3.org, whether you have a short presentation to bootstrap the discussion, and an approximate duration you think the discussion will last.

Previous minutes

https://www.w3.org/community/lbd/meeting-minutes/
https://github.com/w3c-lbd-cg/lbd/tree/gh-pages/minutes