Poster: Scrabble: Converting Unstructured Metadata into Brick for Many Buildings

Jason Koh[1], Bharathan Balaji[2], Dhiman Sengupta[1],
Julian McAuley[1], Yuvraj Agarwal[3], Rajesh Gupta[1]
[1]University of California - San Diego, [2]University of California - Los Angeles, [3]Carnegie Mellon University
{jbkoh,dsengupt,jmcauley,rgupta}@ucsd.edu,bbalaji@ucla.edu,yuvraj.agarwal@cs.cmu.edu

ABSTRACT

Buildings traditionally consist of vertically integrated subsystems installed by multiple vendors in different times without common understanding of the entire system. It results in unstructured metadata of thousands of data points, which third part vendors who seek to deploy applications like fault diagnosis need to map into a common schema. This mapping process requires deep domain expertise in both the schema and buildings with significant manhours. Our framework, *Scrabble*, significantly reduces effort of mapping multiple buildings by introducing a two-stages active learning mechanism that exploits the structure present in a standard schema, Brick, and learns from buildings that have already been mapped to the schema. Scrabble maps characters of metadata into intermediate representation (IR) using conditional random fields and then to labels with a modified classifier chain. Introducing IR enables reusing the learned model for other buildings. Our model requires only minimal input from domain experts for mapping. We have evaluated Scrabble reduces 60 % of samples to achieve 95 % accuracy covering more labels with 2.54 times higher macro F1 at compared to a naive baseline.

1 INTRODUCTION

While many metadata schemata such as IFC[1], Project Haystack[2] and Brick [1] have been proposed to provide a common understanding of building resources, their adoptions are very slow because it requires significant human effort and deep domain expertise for the target building. A building manager is usually given raw string metadata of data points from a target Building Management System (BMS), which she needs to map to a target schema. An example in Fig. 1 is parsing that "RM-3.ZNT" given by a BMS is a *zone temperature sensor* located in *room 3*. The Brick Metadata example follows a formal syntax, Turtle[3]. The task is especially difficult as the raw metadata scarcely follow a standard even in a building or buildings in the same campus.

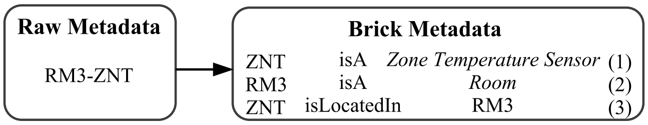

Figure 1: Example metadata normalization to Brick. The raw metadata is unstructured and only a human expert can read it as in Brick metadata.

Though there is works proposed to reduce human effort in normalizing metadata [2], it lacks in the context of multiple buildings but for a building. Large effort is unavoidable to convert multiple buildings from the scratch. We thus propose a framework, *Scrabble*, exploiting known mapping of some buildings to reduce effort in normalizing target buildings. Scrabble uses two-stages machine learning model to maximize reusability with the structure of Brick and an observation that the meaning of a character sequence is consistent across different buildings. We evaluated Scrabble reduces 60 % of samples to achieve 95 % accuracy covering more labels with 2.54 times higher macro F1 at best compared to a baseline.

2 BRICK

We briefly introduce Brick as Scrabble utilizes its structure. Brick [1] is a metadata schema capable of representing all necessary vocabularies and relationships for building applications. It consists of Tags, TagSets and relationships among Tagsets. Tags constitute TagSets like *temperature* and *sensor* constitute *temperature sensor*. TagSets are organized in a tree-like hierarchy as *temperature sensor* is a subclass of *sensor*. As in Fig. 1, an actual entity in Brick is associated with a TagSet ((1) and (2)) and an entity can have relationships with other entities like using *isLocatedIn* (3).

3 SCRABBLE

Fig. 2. summarizes the entire framework. The model consists of two stages: from characters of raw metadata to Brick Tags as intermediate representation (IR); and from a set of Brick Tags to TagSets, which are actual labels to identify. The separation would improve the reusability of the learned model as two same character sequences commonly mean the same thing. For example, "ZN" stands for zone in many contexts such as ZNT for a zone temperature sensor and ZN-101 for Zone-101. Once Scrabble learns a model from the set of examples in given source buildings, it iteratively asks domain experts to provide examples for a target building to cover raw metadata patterns unknown in existing samples.

In the first stage, we learn the model mapping characters to Brick Tags with conditional random fields (CRF) [3]. CRF is a statistical model widely used for sequential learning. In sequential

[1] IFC Add2 Release, http://www.buildingsmart-tech.org/specifications/ifc-releases/ifc4-add2", date accessed: 2017-09-06

[2] Project Haystack: http://project-haystack.org/, date accessed: 2016-09-06.

[3] http://www.w3.org/TR/turtle/

BuildSys '17, November 8–9, 2017, Delft, Netherlands
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5544-5/17/11.
https://doi.org/10.1145/3137133.3141448

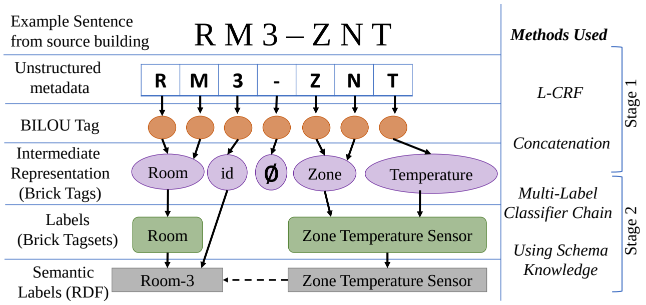

Figure 2: Scrabble Overview. It uses two-stages machine learning model that can be reused across different buildings.

learning, character's position contains important information about the actual label. Thus a character's label and location are together encoded as BILOU (Begin, Inside, Last, Out, Unit) tags. For example, R in RM is in the beginning location of RM for *Room*. Thus, the R's BILOU tag is Room-B. Identified BILOU tags are concatenated to represent actual Brick Tag (e.g., continuous Room-B and Room-L indicate *room* together.) From this stage, we can infer Tags that a raw sentence contains as the sentence, "RM3-ZNT", contains *room*, *zone* and *temperature*.

In the second stage, we map sets of Tags to TagSets, which is multi-label classification problem. It is often not straightforward to infer what a set of Tagsets represents due to a few reasons: i) some Tags are missing in the original sentence (e.g., no *sensor* in the example), ii) Tags frequency distributions for the same set of TagSets vary across different buildings, and iii) IR from the first stage may contain noise labels. Thus, we learn the mapping from samples when it is not in the currently learned model. However, there are too many possible TagSets (1000) to learn while we have a limited number of samples. For example, there can be a unique label with only one example. The label is statistically difficult to learn a model and such problem is called learning from imbalanced data. We exploit several methods[4] to mitigate the problems: we i) synthetically generate samples from Brick Tagsets, ii) introduce a modified Classifier Chain [4] exploiting Brick TagSet hierarchy to learn more stably, and iii) oversample true samples for each label. With the methods, we can learn the mapping from imbalanced samples.

After a model is learned, there will be still untouched examples in the target building. For example, *humidity sensor* in a target building may not exist in the source buildings. We pick unexplored examples based on how much the raw sentence has been utilized for the current mapping. "humid" in raw metadata may have not been used for TagSeet mapping because its label is unknown at the source building. We ask such examples to a domain expert until the mapping becomes complete.

4 EVALUATION

We evaluate Scrabble's learning speed over a baseline as shown in Fig. 3. It tracks accuracy and macro F_1 over the number of given samples for both Scrabble and the baseline. We devise a naive baseline as there is no existing frameworks with exactly same goal as ours. In the baseline, all raw sentences are tokenized with continuous alphabets and vectorized through Bag of Words (BoW) method. The vectors are then labeled with Brick TagSets and we learn Random Forest classifier from the mapping. Learning samples are randomly selected. As in both cases, accuracies increase rapidly at the early steps because there are a few label sets whose numbers are dominant compared to the other types of labels. E.g., there are many *zone temperature sensor*, so if we happen to learn it, many targets are covered. Still, Scrabble shows better or similar results than the baseline. Macro F_1 measures coverage of types of labels that the model learned. It shows bigger differences than accuracy as Scrabble can effectively select labels not learned yet by reusing learned examples.

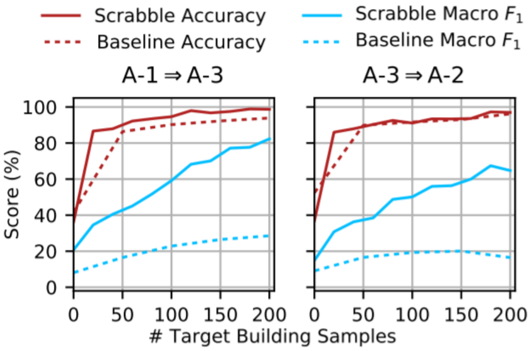

Figure 3: Comparison of learning speeds between buildings. A-1, A-2 and A-3 are different buildings in the same campus and a building left to the arrow is a source building and the other is a target building. Scrabble has a better learning rate over sample numbers with similar or better accuracy and better macro F_1 compared to the baseline.

5 CONCLUSION AND FUTURE WORK

We have initially proposed Scrabble as a framework to reuse knowledge in buildings to normalize a new buildings' metadata. Metadata normalization is important for wide adoption of smart applications in buildings and interoperability across systems. We still need to devise a better sample selection mechanism to increase learning speed more drastically. We will compare the result to more sophisticated baselines. We also envision to apply Scrabble to generic IoT domains such as water systems and environment observation systems where interoperability is critical.

REFERENCES

[1] Bharathan Balaji, Arka Bhattacharya, Gabriel Fierro, Jingkun Gao, Joshua Gluck, Dezhi Hong, Aslak Johansen, Jason Koh, Joern Ploennigs, Yuvraj Agarwal, et al. 2016. Brick: Towards a unified metadata schema for buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM, 41–50.

[2] Arka A Bhattacharya, Dezhi Hong, David Culler, Jorge Ortiz, Kamin Whitehouse, and Eugene Wu. 2015. Automated metadata construction to support portable building applications. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. ACM, 3–12.

[3] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[4] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2009. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases* (2009), 254–269.

[4] Details of each method will be presented at the poster.