Sequential Learning with Active Partial Labeling for Building Metadata

Lu Lin, Zheng Luo, Dezhi Hong, Hongning Wang
Department of Computer Science, University of Virginia
{ll5fy,zl5sv,dh5gm,hw5x}@virginia.edu

ABSTRACT

Modern buildings are instrumented with thousands of sensing and control points. The ability to automatically extract the physical context of each point, e.g., the type, location, and relationship with other points, is the key to enabling building analytics at scale. However, this process is costly as it usually requires domain expertise with a deep understanding of the building system and its point naming scheme. In this study, we aim to reduce the human effort required for mapping sensors to their context, i.e., metadata mapping. We formulate the problem as a sequential labeling process and use the conditional random field to exploit the regular and dependent structures observed in the metadata. We develop a suite of active learning strategies to adaptively select the most informative *subsequences* in point names for human labeling, which significantly reduces the inputs from domain experts. We evaluated our approach on three different buildings and observed encouraging performance in metadata mapping from the proposed solution.

KEYWORDS

Sequence Labeling, Metadata Inference, Smart Buildings

ACM Reference Format:
Lu Lin, Zheng Luo, Dezhi Hong, Hongning Wang. 2019. Sequential Learning with Active Partial Labeling for Building Metadata. In *The 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '19), November 13–14, 2019, New York, NY, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3360322.3360866

1 INTRODUCTION

The rapid development of the Internet of Things ushers in a new wave of sensors and controllers with an unprecedented level of capability and usability, which catalyzes the grand vision for intelligent buildings. However, the thousands of sensors and actuators are usually provided by different vendors who follow disparate conventions to name their points[1], and it is thus hard for third-party application and service providers to interpret the context information about these points and further utilize them.

The contextual information about a sensor point includes its type, location, etc. Such information is typically encoded as the *metadata*, which is a short text string comprised of several abbreviations, and also called *point name*. Table 1 shows three examples of metadata, and the mapped context is represented as *labels*. The first two point names use distinct formats even within the same building for the same sensor type, due to the lack of a standardized naming convention. Consequently, significant manual effort from domain experts is required on a per-building basis for metadata mapping.

Table 1: Examples of metadata and corresponding labels.

Metadata	Label
ebu3b.fschw.chwp4-alm	building, CHW, pump alarm sensor
ebu3b.chw-sys.chwp2-alm	building, CHW, pump alarm sensor
SDH.CHW1.CWP3_ALM	building, CHW, pump alarm sensor

We aim to automatically parse the point names and map them to standardized labels, while reducing the manual inspections from domain experts. Various prior works have been proposed to extract all the encoded information from the raw textual metadata [3, 6]. However, these solutions usually assume a consistent format for all the point names, which often does not hold in practice. Some learning-based solutions have been proposed for metadata mapping, but they can only classify the points into predefined types [1, 4].

In order to reduce human labeling, we seek to leverage the *regularity* and *dependency* in the name conventions for mapping metadata. In particular, vendors often follow certain structures when creating the point names. Table 1 illustrates metadata with the following structure: the chilled water system (CHW) identifier follows the building name, and is followed by the sensor type; dependency exists. As such structures usually repeat in multiple points, we seek to learn these latent structures from a small set of point names which ideally cover all structures. As a result, we formulate this metadata parsing problem as a sequence labeling task and those unknown structures as syntax for generating the sequences.

We propose a generic framework to address this sequential labeling problem, requiring minimal manual effort from domain experts. We use a Conditional Random Field (CRF) model [7] to extract the latent structures from a set of metadata. To reduce the human labor for labeling, we optimize the CRF model via active learning (AL), which only asks experts to label the most informative metadata instance in each iteration. Particularly, we propose to only solicit labels for the most informative *subsequences* of a selected point name. For example, in Figure 1, the red sub-sequence appears occasionally, and thus contains more information to infer new syntax, while the black counterpart is frequent and can be well predicted by the CRF model; we only solicit labels for the red phrase, rather than for the entire string. We call the labeling mechanism as such *partial labeling*, and this is in sharp contrast to prior works that require each entire point name to be labeled [3, 6]. We demonstrated the effectiveness of our active learning based sequential labeling approach on three metadata corpus.

[1] A `point` is a physical sensor or controller, or a software value such as setpoint.

BuildSys '19, November 13–14, 2019, New York, NY, USA

ACM ISBN 978-1-4503-7005-9/19/11…$15.00
https://doi.org/10.1145/3360322.3360866

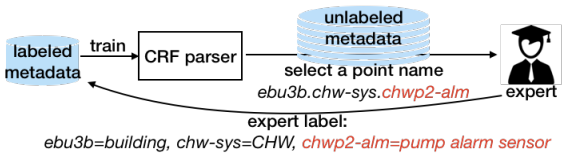

Figure 1: Framework of active sequential learning, where the CRF parser adaptively selects point names from the unlabeled dataset and asks the expert to label. The labeled point names are added to the training set for CRF update.

2 RELATED WORK

Researchers have attempted to systematically address the problem of metadata parsing. A programming language based solution is exploited to derive a set of regular expressions from labeled instances to extract sensor context [3]. Scrabble uses a CRF model to map the labels from metadata and develops a deep neural network to transfer the learned model to new buildings [6] . Active learning has been extensively explored to infer a certain aspect of the metadata. For example, Hong et al. developed a clustering-based active learning algorithm to recognize different sensor types by point names [4]. Similarly, Balaji et al. uses active learning, combined with hierarchical clustering, to learn the sensor types [1]. However, these works require fully labeled instances for training, whereas in contrast our method can learn the patterns in metadata from only partially labeled instances. This leads to further reduction in manual labels from experts. Furthermore, we consider the problem of metadata mapping as a sequence labeling problem, and develop a suite of active learning strategies to reduce the amount of manual labeling effort. For example, we utilize the testing sequences to guide our instance selection (e.g., transductive learning) and enable partial labeling in the queried instances. This better balances the amount of manual labels and the effectiveness of model training.

3 METHODOLOGY

In this section, we use conditional random fields (CRF) to solve the sequence labeling task for metadata, and develop a suite of active learning strategies with partial and transductive labeling to minimize the amount of required manual labels for model training.

3.1 Sequence Labeling with CRFs

We aim to parse the metadata of building sensors and map them to normalized labels that describe the physical context. We formulate this metadata mapping problem as a sequence labeling task: given an input string comprised of T characters as $\boldsymbol{x} = \langle x_1, \ldots, x_T \rangle$, we want to map it to a corresponding label sequence $\boldsymbol{y} = \langle y_1, \ldots, y_T \rangle$. To make it a one-to-one mapping, each character in a point name is labeled with the Beginning-Inside-Outside (BIO) scheme [9] that represents the relative position of a character in the label sequence. For the third examples in Table 1, the substring ⟨C, H, W, 1⟩ are assigned the labels ⟨B-chw, I-chw, I-chw, I-chw⟩ in BIO scheme, respectively: as "C" is at the *Beginning* of "CHW1", while the following characters are *Inside* "CHW1". The punctuation "." is tagged as the "O" which stands for *Out* of any meaningful labels.

The sequence labeling task thus learns to map the point name string to BIO labels. We realize this mapping by a linear-chain Conditional Random Fields (CRF) model [7], which is a family of probabilistic models that estimate the *conditional probability* of a label sequence $\boldsymbol{y}$ given the character sequence $\boldsymbol{x}$ by: $p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{t=1}^{T} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \boldsymbol{x}_t)\right)$, where $Z(\boldsymbol{x})$ is a normalization term over all possible combinations of $\boldsymbol{y}$, and $\Theta = \{\lambda_k\}_{k=1}^{K}$ are the model parameters that weigh the corresponding features $\{f_k\}_{k=1}^{K}$. These features describe the co-occurrence of $\boldsymbol{x_t}$, label y_t, and adjacent label y_{t-1}, which is the key to capture the syntax underlying the point naming scheme. We define a rich set of features in Table 2, where $\mathbf{1}_A$ is an indicator function that returns 1 if the condition A is true, and 0 otherwise. For example in Figure 1, when $t = 1$, $f^{(pre)}_{\text{C, I-chw}} = 1$, it translates to "the preceding character is 'C', and the label of the current character is I-chw", and $f^{(trans)}_{\text{B-chw, I-chw}} = 1$ reflects that "if the current character is labeled as B-chw, the label of next character should be I-chw".

Table 2: The features defined for the sequence labeling task.

Feature	Syntax
$f^{(cur)}_{i,j} = \mathbf{1}_{x_t=i, y_t=j}$	current token is i, current label is j
$f^{(pre)}_{i,j} = \mathbf{1}_{x_{t-1}=i, y_t=j}$	previous token is i, current label is j
$f^{(nxt)}_{i,j} = \mathbf{1}_{x_{t+1}=i, y_t=j}$	next token is i, current label is j
$f^{(BOS)}_{j} = \mathbf{1}_{y_0=j}$	beginning of sequence with label j
$f^{(EOS)}_{j} = \mathbf{1}_{y_T=j}$	end of sequence with label j
$f^{(trans)}_{j_1,j_2} = \mathbf{1}_{y_{t-1}=j_1, y_t=j_2}$	previous label is j_1, current label is j_2

The CRF's parameters Θ are learned in a supervised manner [7]. Given the training set of metadata $X = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$ with labels $Y = \{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^N\}$, the weights Θ are estimated by maximizing the conditional likelihood: $\Theta = \text{argmax}_\Theta \sum_{n=1}^{N} \log p_\Theta(\boldsymbol{y}^n|\boldsymbol{x}^n)$. After training, we use the learned model to predict labels of new point name instances via the Viterbi algorithm.

3.2 Active Learning for CRFs

We train the CRF with an active learning way, which aims to achieve high prediction accuracy while minimizing the labeling effort. In particular, the CRF model is initialized with only a few labeled instances, and is then updated on the fly by querying human annotators for the most informative instances from an unlabeled dataset.

In order to select a sequence for labeling, it is critical to measure the *informativeness* of each instance. We denote the most informative instance as $\boldsymbol{x}^*$, which is selected from the unlabeled dataset $X_{\mathcal{U}}$ by an evaluation function $\phi(\boldsymbol{x})$ under a CRF model Θ. Various formulations of $\phi(\boldsymbol{x})$ have been proposed [10], among which we exploit Token Entropy (TE) as the base measurement, given by:

$$\phi^{TE}_{\Theta}(\boldsymbol{x}) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{|\mathcal{J}|} p_\Theta(y_t = j|\boldsymbol{x}) \log p_\Theta(y_t = j|\boldsymbol{x}), \quad (1)$$

where $\mathcal{J}$ is the label set, and $p_\Theta(y_t = j|\boldsymbol{x})$ is the marginal probability of the label $j \in \mathcal{J}$ for x_t, given by the current model. Intuitively, more confident predictions would have lower TE values.

Full Labeling v.s. Partial Labeling. Standard active learning solutions solicit the labels for each entire selected sequence, which we refer to as *full labeling*. The labeling effort can be further reduced by conducting more selective annotation: We acquire expert's annotation only for the most informative subsequences, which we

Algorithm 1 Active sequence labeling with test set

Input: labeled set X, unlabeled set $X_{\mathcal{U}}$, test set $X_{\mathcal{T}}$

while cost for human labeling is within budget **do**
  $\Theta \leftarrow \text{train}(X)$
  // Choose top M informative test sequences from $X_{\mathcal{T}}$
  $X'_{\mathcal{T}} \leftarrow \text{argmax}^{M}_{\boldsymbol{x} \in X_{\mathcal{T}}} \phi_{\Theta}(\boldsymbol{x})$
  // Choose the most informative sequence in $X_{\mathcal{U}}$
  $\boldsymbol{x}^* \leftarrow \text{argmax}_{\boldsymbol{x} \in X_{\mathcal{U}}} \phi_{\Theta}(\boldsymbol{x}) \times (sim(\boldsymbol{x}, X'_{\mathcal{T}}))^{\beta}$
  $X \leftarrow X \cup \langle \boldsymbol{x}^*, \text{label}(\boldsymbol{x}^*) \rangle, X_{\mathcal{U}} \leftarrow X_{\mathcal{U}} - \boldsymbol{x}^*$
return x

refer to as *partial labeling*. This is motivated by the observation that the model is usually confident about frequently occurring subsequences, and its uncertainty mostly lies in those infrequent subsequences. To perform partial labeling, we extend the informativeness measurement TE to subsequence level:

$$\phi^{TE}_{\Theta}(\boldsymbol{x}_{t:t+w}) = -\frac{1}{w} \sum_{t'=t}^{t+w} \sum_{j=1}^{|\mathcal{J}|} p_{\Theta}(y_{t'} = j|\boldsymbol{x}) \log p_{\Theta}(y_{t'} = j|\boldsymbol{x}),$$

where the subsequence starts at index t over a window size w, and there will be $T - w + 1$ subsequences for a sequence of length T. Among all the subsequences obtained from unlabeled instances, we select the most informative one for human labeling, and use the model to label the rest of this sequence. This reduces human annotations from T to w in each step, compared with fully labeling. A smaller w requires less human effort but may introduce more erroneous labels to the remaining tokens, as fewer human labels would yield a less accurate model. On the contrary, a larger w provides more accurate training labels but increases the manual labeling cost. We will later discuss about this trade-off in our evaluation.

Inductive Labeling v.s. Transductive Labeling. Prior works using TE do not utilize any observations from the test instances [10]. Such mechanism that completely isolates test set from learning process is called *inductive labeling*. However, in our problem, we do have access to the test set, e.g., the points to be labeled in the target building. We revise the sample selection strategy with respect to *transductive learning*, where we select the most similar unlabeled instances to the most informative test cases. The intuition is that similar point names may encode similar syntax, and we can foresee the naming patterns of the target test set by choosing the unlabeled instances in a transductive manner, so as to make the learning process highly targeted, hence greedy and efficient. The *similarity* between one instance in the unlabeled set $\boldsymbol{x} \in X_{\mathcal{U}}$ and a set of instances from the test set $X'_{\mathcal{T}} \in X_{\mathcal{T}}$ is measured by:

$$sim(\boldsymbol{x}, X'_{\mathcal{T}}) = \frac{1}{|X'_{\mathcal{T}}|} \sum_{\boldsymbol{x}' \in X'_{\mathcal{T}}} cosine\big(\pi(\boldsymbol{x}), \pi(\boldsymbol{x}')\big). \quad (2)$$

The cosine similarity between two sequences is calculated based on their k-mers feature π [8], which is the set of all the contiguous subsequences of length-k in the input sequence.

Algorithm 1 summarizes our active sequence labeling process: We first choose M most informative instances from the test set using $\phi_{\Theta}(\boldsymbol{x})$, then select a new sequence from the unlabeled set based on the product of $\phi_{\Theta}(\boldsymbol{x})$ and $sim(\boldsymbol{x}, X'_{\mathcal{T}})$. The coefficient β controls the relative importance of the similarity to test set, which allows us to better balance the information from test set.

Table 3: Details of evaluation buildings.

Building	Location	Year	#Points	Metadata Length
EBU3B	San Diego, CA	2004	1074	7 ~ 35
SDH	Berkeley, CA	2009	2551	7 ~ 31
IBM	Dublin, Ireland	2011	1366	10 ~ 33

4 EVALUATION

In this section, we evaluate the proposed transductive active partial sequence labeling method on the metadata of three buildings.

4.1 Experimental Setup

4.1.1 Dataset. As all the methods for evaluation can produce labels for each entire metadata string, we obtain a subset of three buildings that has the character-level full labels from the studies of Brick [2] and Plaster [5], as detailed in Table 3.

4.1.2 Metric. To evaluate the predictive performance of the model, we measure the *phrase-level* accuracy, where a *phrase* is a group of characters that represent the same entity, e.g., "chw1" as a whole is labeled as "CHW". A phrase is considered as correctly labeled if and only if each token's BIO label is correct.

4.1.3 Baselines. We compare *transductive* and *partial* labeling strategies with two active learning based baselines: `TE` given by Eq.(1), and information density based TE (`denTE`) [10] which weights TE of $\boldsymbol{x}$ by its similarity to all the unlabeled instances in $X_{\mathcal{U}}$:

$$\phi^{DenTE}_{\Theta}(\boldsymbol{x}) = \phi_{\Theta}(\boldsymbol{x}) \times \Big(\frac{1}{|X_{\mathcal{U}}|} \sum_{\boldsymbol{x}' \in X_{\mathcal{U}}} cosine(\pi(\boldsymbol{x}), \pi(\boldsymbol{x}'))\Big)^{\beta}, \quad (3)$$

where the computation of cosine similarity follows Eq.(2). Instead of comparing the unlabeled instance with test cases, this method only compares it within the unlabeled set and finds the centroid as the most representative instance to query for human label.

4.2 Results and Analysis

For each dataset, we fix 15 labeled instances to initialize the CRF model, and perform 8-fold cross-validation: each fold serves as test set in turn, and the rest as the unlabeled set from which active learning strategies select instances to update the model. We report the phrase-level accuracy in Figure 2, where x-axis is the number of BIO labels solicited from human, which reflects the level of human burden for labeling, and y-axis is the phrase-level accuracy averaged over 8-folds. Specifically, we set $M = 100$ for transductive learning, and $\beta = 1.0$ for both `denTE` and transductive learning, $w = \{11, 8, 19\}$ for partial labeling in Building EBU3B, SDH, and IBM, respectively. The source code and data used in our experiments are readily available online [2].

Generally, the best CRF learned by partial and transductive active learning (`transTE-part`) converges fast and achieves a 94% accuracy on Building EBU3B with only 650 human labels ($\approx$ 29 full point names with average length of 22), while it uses 600 labels to achieve a 95% accuracy on Building SDH and a 90% accuracy on Building IBM. This demonstrates the fast convergence of active learning for CRF training. Comparing our proposed partial labeling (`TE-part`) with `TE`, we can observe obvious improvement on both convergence rate and accuracy. It validates our intuition: partial

[2] https://github.com/Louise-LuLin/active-partial-labeling

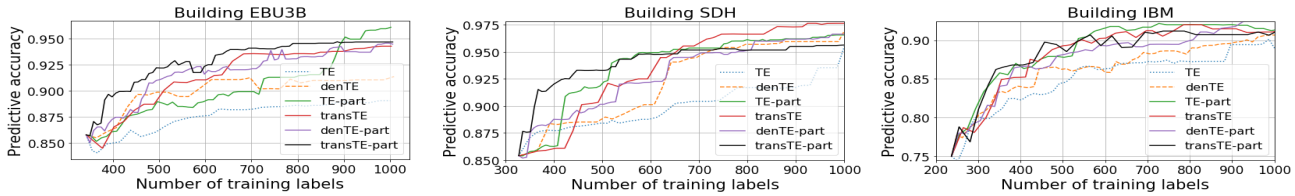

Figure 2: The performance of CRF for labeling point name strings on three test buildings: Partial labeling and tranductive labeling help improve the learning efficiency and accuracy of the active learning process.

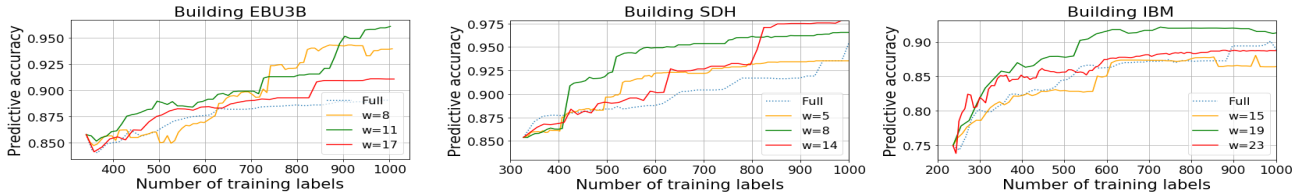

Figure 3: The effect of window size w on partial labeling: A small w may harm the accuracy while a large w will degrade the strategy to full labeling.

Table 4: Trace of Partial Labeling Active learning.

Accuracy:	0.86	0.88	0.91	0.93
Selected:	ebu#b.fschw.chwp#-dly	ebu#b.chw-sys.chwp#-alm	ebu#b.chwp#-vfd.voltage	ebu#b.chwp-sys.chw-lead
Worst predicted:	ebu#b.chw-sys.cwdp-sp	ebu#b.chwp#-vfd.voltage	ebu#b.chwp-sys.chw-lead	ebu#b.fschw.chwp#-vfd
	ebu#b.chw-sys.chwp#-alm	ebu#b.fschw.chwp-db	ebu#b.chwp#-vfd.hoa-sts	ebu#b.chw-sys.cwdp-sp

labeling eliminates redundant sub-sequences that the model is already confident about. Comparing transductive labeling (`transTE`) with `TE` and `denTE`, we find a fast increasing accuracy which suggests that using test set can indeed guide the model to target for the test cases. `transTE` gains more improvement in the later stage when the model starts to converge on Building SDH, which shows the potential of transductive learning when the informativeness across instances is less differentiable and targeted training becomes more important.

As in our partial labeling solution, w controls the balance of human cost and data accuracy. We evaluate how the model performs when we use different window size w for partial labeling. The results are summarized in Figure 3, where "Full" indicates full labeling, and we shall note that it is equivalent to the first stage that uses CRF to obtain BIO tags in [6]. The value of w varies for different buildings since their point names have different lengths. Comparing with the w that achieves the highest accuracy (i.e. 11, 8, 19 for Building EBU2B, SDH, IBM), we find that oftentimes a smaller w may degrade the learning by introducing erroneous labels from inaccurate model, but a larger w may degrade the strategy to full labeling. We also inspect the quality of the selected subsequences. Table 4 presents the selected subsequence for labeling in a few iterations. We see that examples that best resemble poorly-predicted test cases are selected for labeling by our strategy, and that the subsequences are mostly the phrases for indicating the sensor type, which bears the most diversity; this demonstrates the effectiveness of the proposed partial labeling active learning.

5 CONCLUSION

In this study, we tackle the problem of automated metadata mapping in sensor point names. We formulate it as a sequential labeling problem and develop a suite of active learning strategies to adaptively select the most informative subsequences for model training, in order to minimize the human labeling effort. We evaluated our approach in three commercial buildings, and the results demonstrate the effectiveness of partial labeling and transductive labeling for active learning. As future work, we plan to address partial labeling in the way of structured learning, e.g., learning to search for the optimal subsequence for labeling, and find a proper solution to handle errors in model training introduced by its own prediction.

ACKNOWLEDGMENTS

We thank our shepherd, Clayton Miller, and the reviewers for helpful comments. This work was supported by National Science Foundation IIS-1718216 and Department of Energy DE-EE0008227.

REFERENCES

[1] Bharathan Balaji and et al. 2015. Zodiac: Organizing large deployment of sensors to create reusable applications for buildings. In *BuildSys*. ACM, 13–22.
[2] Bharathan Balaji and et al. 2016. Brick: Towards a unified metadata schema for buildings. In *BuildSys*. ACM, 41–50.
[3] Arka A Bhattacharya and et al. 2015. Automated metadata construction to support portable building applications. In *BuildSys*. ACM, 3–12.
[4] Dezhi Hong, Hongning Wang, and Kamin Whitehouse. 2015. Clustering-based active learning on sensor type classification in buildings. In *CIKM*.
[5] Jason Koh and et al. 2018. Plaster: An integration, benchmark, and development framework for metadata normalization methods. In *BuildSys*. ACM, 1–10.
[6] Jason Koh and et al. 2018. Scrabble: transferrable semi-automated semantic metadata normalization using intermediate representation. In *BuildSys*.
[7] John D. Lafferty and et al. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*. 282–289.
[8] Christina S Leslie and et al. 2004. Mismatch string kernels for discriminative protein classification. *Bioinformatics* 20, 4 (2004), 467–476.
[9] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*. 147–155.
[10] Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*.