

PNNL-38511

Automated Tagging and Semantic Mapping of BACnet Metadata to Brick Schema for Building Automation Systems

September 2025

Milan Jain
Robert Lutes
Srinivas Katipamula



U.S. DEPARTMENT
of **ENERGY**

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062

www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: <http://www.ntis.gov>

Automated Tagging and Semantic Mapping of BACnet Metadata to Brick Schema for Building Automation Systems

September 2025

Milan Jain
Robert Lutes
Srinivas Katipamula

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Mapping BACnet point names to standardized Brick classes is essential for interoperability and analytics in building automation systems, yet manual tagging remains time-consuming and error prone. Automating this process can reduce labor costs, improve scalability across large building portfolios, and enable advanced analytics for energy efficiency and diagnostics. Previous approaches have relied on classifiers trained on curated datasets such as Mortar or on rule-based methods, both of which struggle with rare abbreviations, vendor-specific conventions, and heterogeneous building contexts. In this work, we propose an automated semantic tagging framework that integrates classifier-based normalization, embedding-driven semantic matching, and foundational language models, extended with retrieval-augmented prompting (RAP) to inject domain-specific context. We evaluate the framework across eight anonymized PNNL-operated buildings comprising more than 7,429-point names, with classifiers trained on Mortar and predictions validated against SME-curated ground truth. Results show that classifiers achieved 22.0% accuracy, foundational models (Llama3) achieved 24.1%, and RAP improved accuracy to 27.9%, confirming that contextual grounding is critical to robust inference. Building on these results, future work will explore contextual enrichment using metadata and time-series signals, expansion of training corpora, transfer learning for domain adaptation, and flexible top-k recommendations to further reduce reliance on manual tagging.

Summary

The ability to translate raw BACnet point names into standardized Brick classes is essential for enabling consistent, interoperable analytics in building automation systems. Today, this process is largely manual, requiring significant subject matter expertise and effort, which limits scalability across large building portfolios.

This work develops and evaluates an automated semantic tagging framework that reduces reliance on manual mapping. The framework combines three components: (i) classifiers trained on the Mortar dataset to normalize abbreviations, (ii) embedding-based semantic matching to align normalized names with Brick ontology classes, and (iii) foundational models (Llama3) to extend beyond the coverage of training data. We further introduce retrieval-augmented prompting (RAP), which enhances the foundational model by supplying domain-specific exemplars as contextual hints.

The framework was deployed across eight anonymized PNNL-operated buildings spanning more than 7,000-point names collected via VOLTTRON. Predictions were validated against SME-curated mappings. Classifiers achieved 28.3% accuracy, with Linear SVC performing best. The foundational model alone achieved 26.6%, showing strong generalization to rare abbreviations but limited performance without building-specific context. Incorporating RAP improved accuracy to 35.6%, representing a 9.0 percentage point gain over the baseline foundational model and a 7.3-point improvement over the best classifier.

Error analysis identified three recurring challenges: (1) coverage gaps for rare setpoints, (2) confusion between semantically similar operational states, and (3) misclassification of highly specific, site-dependent terms. RAP effectively addressed the first category by grounding inference in domain exemplars, while the other two remain active areas for improvement.

Looking ahead, future work will focus on contextual enrichment through metadata and time-series data, expansion of training corpora to capture diverse vendor practices, and modeling advances such as transfer learning, direct point-to-class mapping with foundational models, and top-k class recommendations to support SME-in-the-loop validation. Together, these directions aim to deliver scalable, accurate, and context-aware semantic tagging pipelines that can accelerate the adoption of advanced analytics in building energy management.

Acknowledgments

This research was supported by the Energy and Environment Directorate, under the Small Money Agile Research and Technology Transitions (SMARTT) Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE AC05 76RL01830.

Acronyms and Abbreviations

AI – Artificial Intelligence
BACnet – Building Automation and Control Network
BC – Brick Class
CFM – Cubic Feet per Minute
FM – Foundational Model
HP – Heat Pump
HVAC – Heating, Ventilation, and Air Conditioning
JCI – Johnson Controls
LLM – Large Language Model
MCP – Model Contextual Prompting
ML – Machine Learning
RAG – Retrieval-Augmented Generation
RAP – Retrieval-Augmented Prompting
RDF – Resource Description Framework (data model used in Mortar TTL files)
RF – Random Forest (classifier)
SME – Subject Matter Expert
SVC – Support Vector Classifier (Linear SVC)
TTL – Terse RDF Triple Language (RDF serialization format used in Mortar dataset)
VAV – Variable Air Volume
VFD – Variable Frequency Drive
XGBoost – Extreme Gradient Boosting (classifier)

Contents

Abstract ii

Summary iii

Acknowledgments iv

Acronyms and Abbreviations v

1.0 Introduction 1

2.0 Background 3

 2.1 Automated Metadata Extraction and Normalization 3

 2.2 Semantic Standards for Building Metadata 4

 2.3 Automated Conversion to Standards 4

 2.4 Use of Foundational Models 5

3.0 Methodology 7

 3.1 Automated Mapping 7

 3.2 Automated Tagging 8

 3.2.1 AI-Based Point Normalization 9

 3.2.2 Semantic Matching 13

4.0 Case Study 18

5.0 Results and Discussion 20

 5.1 Trained Classifier with Semantic Matching 20

 5.2 Key Findings on Model Limitations 21

 5.3 Foundational Model with Semantic Matching 22

 5.4 Context-Aware Inference with Retrieval-Augmented Prompting 23

6.0 Discussion and Future Work 25

7.0 Conclusion 27

8.0 References 28

Figures

Figure 1. System Architecture 7

Tables

Table 1. Deployment Summary 18

Table 2. Classifier Accuracy in Mapping BACnet Point Names to Brick Classes
Across Eight PNNL Buildings 20

Table 3.	Comparison of classifier (Linear SVC) and foundational model (Llama3) performance on PNNL buildings	22
Table 4.	Results of semantic tagging using Linear SVC, Llama3, and Llama3 with retrieval-augmented prompting (RAP) across eight anonymized buildings.	23

1.0 Introduction

Commercial buildings are increasingly equipped with building automation systems (BAS), which rely on building management systems (BMS) to monitor and control equipment. According to the U.S. Energy Information Administration, there are approximately 5.9 million commercial buildings in the United States. Many medium-to-large facilities deploy BMS with thousands of sensor and control points per building, often exceeding 10,000 in large sites, generating dense operational metadata. While these data streams enable applications such as fault detection and diagnostics (FDD), demand-response, and grid-interactive efficient buildings (GEBs), a persistent barrier remains – metadata inconsistencies. Point names in most deployments follow ad hoc, vendor-specific conventions with no adherence to semantic standards, making cross-building analytics, scalable application deployment, and interoperability costly and error-prone.

To address these challenges, several semantic standards have been proposed, including Brick (Balaji et al. 2016), Project Haystack (“Project Haystack” 2025) and ASHRAE 223P (“ASHRAE Standard 223 User Documentation” 2025). These schemas aim to formalize the representation of building entities, their properties, and relationships, thereby providing a foundation for interoperable applications. Among these, Brick has emerged as a leading candidate due to its ontology-based design, ability to capture spatial and functional hierarchies, and increasing adoption in research and pilot deployments. Yet, a critical bottleneck remains: the manual mapping of unstructured BACnet point names to Brick schemas. This process often requires weeks of effort from building engineers and does not scale to large or diverse building portfolios.

Recent advances in artificial intelligence (AI) offer new opportunities to overcome this bottleneck. Techniques from machine learning (ML), natural language processing (NLP), and foundational large language models (LLMs) provide the ability to parse abbreviations, normalize heterogeneous naming conventions, and infer semantic classes. For example, Zodiac (Balaji et al. 2015) combined hierarchical clustering with random forest classifiers to group and label sensor metadata, significantly reducing manual labeling effort compared to purely regex-based methods. Building Adapter (Hong, Wang, Ortiz, et al. 2015) extended this approach by combining clustering of point names with classifiers trained on sensor time-series data, enabling knowledge transfer across buildings. More recent work such as BrickLLM (Perini et al. 2025) has explored the use of large language models for directly mapping point names to Brick classes, showing promise in capturing semantic relationships without handcrafted rules. Other studies have leveraged natural language techniques like bag-of-words (Bhattacharya et al. 2015) or k-mer embeddings (Hong, Wang, and Whitehouse 2015) to transform point names into feature vectors for supervised classification.

Despite these advances, many prior approaches fall short when confronted with unseen abbreviations, semantically overlapping point types (e.g., distinguishing `Occupancy_Status` from `Occupancy_Command`), or highly specific vendor conventions that do not appear in training corpora. This limitation arises because most methods either rely on predetermined regular expressions, which struggle to capture variability across sites, or on machine learning techniques—such as clustering, supervised classifiers, or active learning—that are only as effective as the training datasets provided. When the training corpus lacks coverage of certain naming conventions or abbreviations, these models tend to misclassify or overgeneralize, leading

to inconsistent mappings across buildings. Foundational models, by contrast, offer the potential for broader generalization since they are pre-trained on diverse, large-scale text corpora and can leverage contextual reasoning beyond what is explicitly present in the training data.

In this work, we address the limitations of prior approaches to semantic tagging of building metadata, which often struggle with unseen abbreviations, overlapping point types, and vendor-specific conventions. We systematically evaluate methods for mapping BACnet point names to Brick classes, comparing the performance of machine learning classifiers, embedding-based semantic matching, and foundational language models. Specifically, we:

- Develop an automated pipeline that integrates classifier-based normalization, embedding-based semantic matching, and foundational language models.
- Evaluate the pipeline across eight anonymized PNNL-operated buildings, comprising a total of 7,429 BACnet point names, using SME-curated mappings as ground truth.
- Introduce and test retrieval-augmented prompting (RAP) to provide building-domain exemplars as context, improving the accuracy of foundational model inference.

Our evaluation reveals that classifiers trained on Mortar data achieved an overall accuracy of 28.3%, performing well in buildings where naming conventions aligned closely with the training corpus. Foundational models (Llama3) showed stronger generalization to rare or novel abbreviations but achieved a slightly lower overall accuracy of 26.6%, reflecting their lack of building-specific grounding. Introducing retrieval-augmented prompting (RAP) provided a significant boost, raising accuracy to 35.6%, a gain of 9.0 percentage points over the baseline foundational model and 7.3 points over the best classifier. Error analysis highlighted three recurring challenges: coverage gaps for rare setpoints, confusion between semantically overlapping classes, and misclassification of highly specific vendor conventions. RAP effectively addressed the first category by contextualizing inference with relevant exemplars, though additional strategies are needed for the latter two.

The remainder of this paper is organized as follows. Section 2 reviews related work on semantic standards and automated tagging approaches in building metadata. Section 3 introduces the case study buildings, describes the data collection process, and outlines the experimental setup. Section 4 details the methodology, including classifier design, semantic matching, and foundational model integration with retrieval-augmented prompting. Section 5 presents the evaluation results, highlighting comparative performance across approaches and error analysis. Section 6 discusses key findings, limitations, and implications for deployment. Finally, Section 7 concludes with future directions, including opportunities for contextual enrichment, transfer learning, and advancing foundational models for fully automated semantic tagging.

2.0 Background

The effective use of building automation system (BAS) data depends not only on reliable sensing and control but also on the ability to represent metadata consistently. Raw BACnet point names are notoriously inconsistent across vendors and sites, creating a major barrier for scalable deployment of applications such as fault detection, diagnostics, and energy management (Bergmann et al. 2020). Over the past decade, researchers have approached this challenge from several directions: (i) automated metadata extraction and normalization techniques, (ii) the development of semantic standards for structuring building metadata, and (iii) methods to convert raw BACnet metadata into these standards. This section reviews prior work relevant in this context and positions our contributions within this evolving landscape. For a detailed discussion on the broader landscape, readers are referred to surveys by (Butzin et al. 2017) and (Aniakor et al. 2024), which outline existing approaches and open challenges.

2.1 Automated Metadata Extraction and Normalization

Early efforts to automate metadata interpretation relied on pattern matching and heuristics. (Reinisch et al. 2008) developed ontology-based rules (with regular expressions) to map sensor labels to standardized concepts. (Schumann et al. 2014) advanced this by combining linguistic similarity, curated dictionaries, and ontology markers to align BMS point labels with application inputs. (Bhattacharya et al. 2014) introduced program-synthesis to learn transformation rules from SME-provided examples, reducing manual curation but still requiring domain-specific input. These approaches demonstrated feasibility but were brittle across buildings.

The field advanced with clustering and active learning. (Bhattacharya et al. 2015) used tokenized point names as features in a supervised maximum entropy classifier to enable portable metadata construction. Zodiac (Balaji et al. 2015) combined hierarchical clustering with random forest classifiers to reduce labeling effort, while Building Adapter (Hong, Wang, Ortiz, et al. 2015; Hong, Ortiz, Bhattacharya, et al. 2015) leveraged clustering of point names together with classifiers trained on time-series data to support cross-building transfer. (Gao et al. 2015) explored unsupervised clustering to group semantically similar point names over large deployments. (Lin et al. 2019) extended this direction by casting metadata tagging as a sequence-labeling problem and employing active partial labeling to minimize expert burden. (Shi et al. 2019) evaluated clustering and time-series features for point-type inference in retrofit contexts. (Mishra et al. 2020) proposed treating point names as sequences and computing k-mer frequency features (analogous to n-gram / bag-of-substring) to vectorize names and feed into supervised classifiers. Wu et al. (2021) introduced Sensei, a self-supervised segmentation framework that decomposes point names into meaningful tokens without labels, which can then be used as improved inputs to clustering or classification models. (Chamari et al. 2023) revisited these challenges in the context of schema generation, presenting a taxonomy of name-based, signal-based, and active-learning methods for smart building metadata.

Recent studies have turned toward probabilistic, neural, and transfer-learning strategies to improve automation and generalization. (Jiao et al. 2020) proposed SeNsER, a cross-building sensor metadata tagger that uses neural sequence models (Char-LSTM + CRF, co-trained language models, and k-mer features) to transfer learned tagging behavior from annotated to new buildings. (Waterworth et al. 2021) explored applying pretrained NLP models (e.g. transformer

architectures) and a custom tokenizer to building sensor metadata, showing that transfer learning from large language models can boost tagging performance. More recently, (Almashor et al. 2023) investigated autoencoder-based encodings of point name strings to embed metadata features for downstream classification. Some approaches integrated metadata with semantic reasoning rather than pure clustering. Together, these studies marked a shift from rigid rule-based approaches toward adaptive, data-driven methods capable of reducing labeling effort and supporting transfer across diverse building datasets.

Despite these advances, challenges persist. Many approaches remain tightly coupled to the training corpus or heuristic rules they rely on, limiting robustness across vendors and building types. Rare abbreviations and site-specific naming conventions are still difficult to capture. Even when clustering or measurement-driven techniques succeed, semantically overlapping point types—for example, distinguishing *Zone_Temperature* from *Zone_Temperature_Setpoint*—require richer contextual reasoning that goes beyond string similarity or basic statistical learning. These limitations have motivated the development of semantic standards, which aim to formalize building metadata in a structured, interoperable form.

2.2 Semantic Standards for Building Metadata

Several standards have emerged to formalize the representation of building entities, their properties, and relationships. Project Haystack (“Project Haystack” 2025) introduced a tag-based framework to flexibly annotate metadata, with adoption in HVAC and metering applications. Industry Foundation Classes (IFC) grew out of the building information modeling (BIM) community, standardizing the exchange of 3D architectural and construction data, though offering limited coverage of BMS metadata. ASHRAE 223P (“ASHRAE Standard 223 User Documentation” 2025) and SAREF (Daniele et al. 2015) have pursued more general ontologies for smart building and IoT applications. Among these, the Brick Schema (Balaji et al. 2016) has gained momentum in both research and practice due to its ontology-based design, graph representation in RDF, and ability to capture spatial and functional hierarchies.

Despite their promise, all of these standards face a practical bottleneck: they still require raw BACnet point metadata to be extracted and mapped into the schema. Without an automated translation layer, adoption remains resource-intensive and error-prone.

2.3 Automated Conversion to Standards

Recognizing this bottleneck, researchers have developed methods to directly map BACnet point metadata into semantic standards. BuildingMOTIF (National Renewable Energy Laboratory [2022] 2025) proposed a library-driven approach to map BACnet points into Brick graphs. Complementary frameworks such as Plaster (Koh, Hong, et al. 2018) unified different normalization methods under a common interface, enabling evaluation and integration with Brick conversion pipelines. Scrabble (Koh, Balaji, et al. 2018) advanced this idea by using an intermediate representation layer and active learning to normalize heterogeneous point names before mapping them into semantic schemas.

Beyond normalization pipelines, several toolkits and interfaces have been developed to streamline conversion. Brickify (“Brickify Tool” 2025) parses point labels and equipment

metadata, applies rule-based and template-driven mappings, and then generates RDF triples that instantiate the corresponding Brick entities and relationships. VizBrick (Lee et al. 2022; 2023) created a GUI-based workflow for authoring and validating Brick-compliant models, supporting interactive schema generation. (Fierro, Koh, et al. 2019; Fierro et al. 2020) formalized Brick’s metadata tags in “Beyond a House of Sticks,” providing the theoretical foundation for expressing portable and machine-readable tags, while Mortar (Fierro, Pritoni, et al. 2019; Fierro, Guduguntla, et al. 2019) offered a large-scale testbed of Brick-modeled buildings to benchmark and evaluate portable building analytics.

Efforts have also expanded toward cross-domain integration. BIM-to-BRICK approaches (Vittori et al. 2025) demonstrated translation of building information models into Brick graphs, enriching spatial and semantic context. Field demonstrations (Roa et al. 2023) have shown the feasibility of mapping BACnet metadata into Brick at scale, validating the ontology in operational buildings. Linked building data frameworks (Ben Ahmed et al. 2024) proposed extending Brick for interoperability with building energy modeling (BEM) workflows. Commercial connectors are also emerging: the Mapped BACnet-to-Brick connector (“Mapped | Connectors | BACnet” 2025) automatically extracts raw BACnet point metadata and translates it into Brick-compliant models, providing evidence that automated conversion is beginning to be deployed in practice.

Collectively, these efforts demonstrate that automated conversion to standards is feasible and increasingly practical. Yet, existing approaches often rely on template-based mappings or narrowly trained models, which hinders their ability to generalize across heterogeneous sites and evolving schema versions. This is precisely where foundational models offer a transformative opportunity: trained on broad and diverse corpora, they capture rich semantic structure that can be adapted to new contexts with little or no supervision. Unlike task-specific models, they can perform zero-shot or few-shot mapping, allowing raw BAS point names to be aligned with standards like Brick and Haystack even in buildings never seen during training.

2.4 Use of Foundational Models

While foundational models hold significant promise for automating semantic tagging, only a handful of studies have begun to investigate their potential in the building domain. BrickLLM (Perini et al. 2025) proposed a Python library that uses LLMs to generate Brick-compliant RDF graphs from natural-language descriptions or structured prompts, focusing on ontology-conformant graph creation rather than direct point-name classification. (Mulayim et al. 2024) analyze how LLMs can assist with the creation and use of semantic ontologies in buildings and outline practical requirements and pitfalls (e.g., hallucination, domain adaptation). (Zheng et al. 2025) further showed that LLMs can tag building management system (BMS) data points with minimal supervision, revealing the promise of few-shot and zero-shot learning for metadata standardization.

Together, these studies indicate that foundational models can capture semantic relationships across heterogeneous metadata without handcrafted rules or training from scratch. However, these approaches remain in their infancy, with open challenges around grounding, context injection, and systematic evaluation across real-world building deployments. In this work, we extend these early efforts by conducting a comparative evaluation of classifiers, semantic matching, and foundational models for mapping BACnet metadata to Brick classes across

multiple operational buildings. To address the lack of building-specific context, we explore retrieval-augmented prompting (RAP) and benchmark performance across 7,429-point names from eight PNNL-operated sites. Our contributions include: (i) developing an automated tagging pipeline that integrates classifiers, embeddings, and foundational models, (ii) introducing RAP as a lightweight contextual enrichment strategy, and (iii) analyzing common error modes to inform future research directions.

3.0 Methodology

Our methodology integrates ontology-guided reasoning with data-driven inference to automate the process of mapping sensor point names to standardized classes. The approach is designed to handle the heterogeneity of building management system (BMS) data, which often relies on inconsistent abbreviations, vendor-specific conventions, and incomplete documentation.

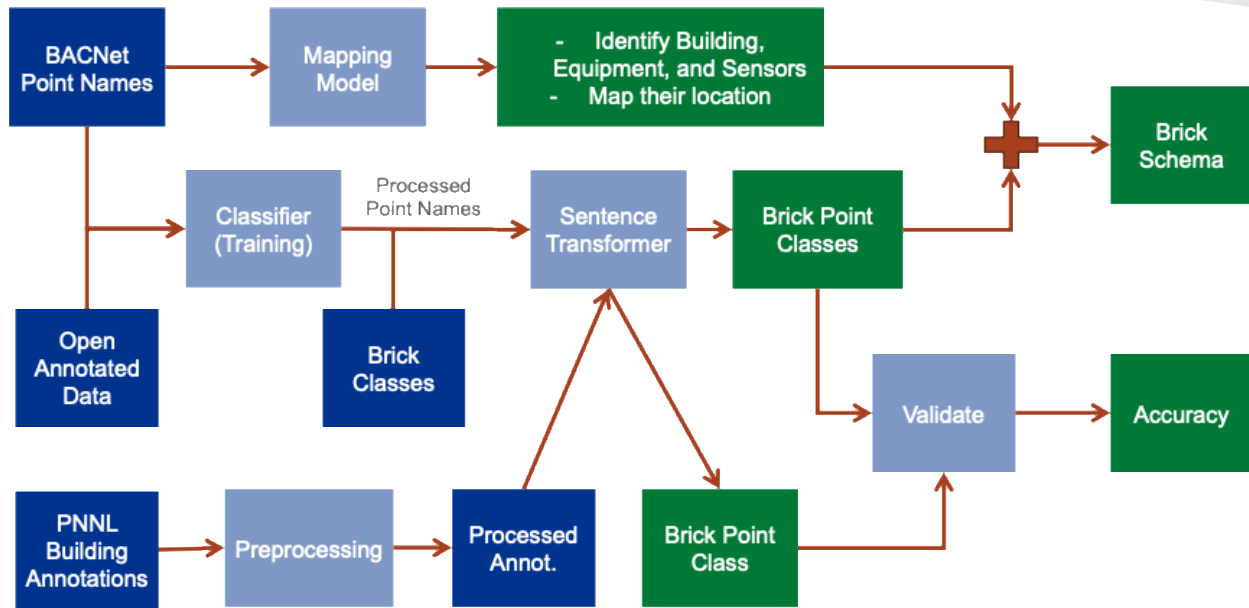


Figure 1. System Architecture

Figure 1 depicts our workflow used to convert raw BACnet point names into Brick-compliant semantic models through two main stages: **Brick Class Identification** and **Building Mapping**. The workflow integrates preprocessing, AI-based point-normalization, semantic similarity, and ontology-driven alignment to generate a structured schema.

3.1 Automated Mapping

To automate schema generation from BACnet point names, we implement a rule-based mapping layer that leverages regular expressions and ontology anchoring. We define a structured parsing rule using regular expressions of the form:

site.equipment.point_type

This pattern enables the automatic extraction of three key components from a point name:

- **Site** – high-level identifier (e.g., campus or facility).
- **Equipment** – subsystem or device hosting the point (e.g., AHU, VAV).
- **Point Type** – sensor or setpoint associated with the equipment (e.g., SATSP, ZN-T).

Extracted components are translated into **RDF triples** that conform to the Brick schema. Each site is instantiated as a `brick:Site`, linked to a `brick:Building`; equipment instances are created and associated with their parent building using `brick:locatedIn`; and individual points are instantiated with their inferred Brick class. Point names serve as human-readable labels, while ontology relationships such as `brick:hasPart` and `brick:hasPoint` ensure proper hierarchical nesting.

Brick class assignment leverages the point normalization and semantic matching pipeline described later. If a point type cannot be resolved to a known Brick class, it defaults to a generic `brick:Point`, while logging a warning for future refinement. This guarantees schema completeness even in the presence of unmapped or novel point types.

Example:

Consider the point name: BLDG405.AHU1.SATSP

1. *Regex Parsing:*

- a. Site = BLDG405
- b. Equipment = AHU1
- c. Point Type = SATSP (Supply Air Temperature Setpoint)

2. *RDF Graph Construction:*

<code>ex:Site_BLDG405</code>	<code>rdf:type</code>	<code>brick:Site</code>
<code>ex:Building_BLDG405</code>	<code>rdf:type</code>	<code>brick:Building</code>
<code>ex:Site_BLDG405</code>	<code>brick:hasPart</code>	<code>ex:Building_BLDG405</code>
<code>ex:Building_BLDG405</code>	<code>brick:isPartOf</code>	<code>ex:Site_BLDG405</code>
<code>ex:AHU1</code>	<code>rdf:type</code>	<code>brick:AHU</code>
<code>ex:AHU1</code>	<code>brick:locatedIn</code>	<code>ex:Building_BLDG405</code>
<code>ex:BLDG405_AHU1_SATSP</code>		<code>brick:Supply_Air_Temperature_Setpoint</code>
<code>ex:AHU1</code>	<code>brick:hasPoint</code>	<code>ex:BLDG405_AHU1_SATSP</code>
<code>ex:BLDG405_AHU1_SATSP</code>	<code>rdfs:label</code>	<code>"BLDG405.AHU1.SATSP"</code>

The result is a **Brick-compliant RDF graph** where the site, building, equipment, and point are explicitly linked through ontology relationships. This structured representation allows downstream applications to query, visualize, and analyze building systems consistently across diverse facilities.

3.2 Automated Tagging

While automated mapping establishes the hierarchical structure of sites, buildings, equipment, and sensors, it does not by itself determine the semantic identity of each point. Point names in

building management systems are often inconsistent, relying on abbreviations and vendor-specific codes. The goal of automated tagging is to infer standardized Brick classes for sensor points by combining AI-based point normalization with ontology-driven semantic matching. This step resolves the ambiguity in vendor-specific abbreviations and inconsistent naming practices, ensuring that each point is tagged with an appropriate and interoperable Brick class.

3.2.1 AI-Based Point Normalization

Point normalization serves as the first step in automated tagging, where raw point names are transformed into consistent representations that capture their intended sensor type. To standardize them, we first normalize point names into canonical forms before mapping them to Brick classes. This stage addresses the challenge of diverse abbreviations and naming conventions used across vendors and sites.

3.2.1.1 NLP-Based Text Classification for Point Normalization

In this approach, we framed point-name normalization as a short-text classification problem, where each raw point name is transformed into a lexical feature vector and mapped to a canonical sensor-type label. To achieve this, we represent point names using the Term Frequency–Inverse Document Frequency (TF–IDF) vectorizer, a standard technique in natural language processing for converting text into numerical features.

The TF–IDF vectorizer assigns weights to tokens based on both their frequency within a point name and their rarity across the dataset:

- **Term Frequency (TF):** counts how often a token (character or word n-gram) appears in the given point name.
- **Inverse Document Frequency (IDF):** downweights tokens that appear frequently across many point names, as they are less informative.
- **TF–IDF weight:** the product of TF and IDF, emphasizing tokens that are frequent in one name but rare overall.

This weighting is critical for building management data, where many point names share generic tokens (e.g., TEMP, SNSR), but meaningful distinctions lie in rarer substrings (e.g., SATSP, ZN-T, RAT). By emphasizing distinctive abbreviations and downweighting generic ones, TF–IDF ensures that classifiers focus on the parts of a point name that best predict its sensor type.

Example:

Consider three-point names:

- SATSP (Supply Air Temperature Setpoint)
- RATSP (Return Air Temperature Setpoint)
- ZN-T (Zone Temperature Sensor)

The trigram SAT appears only in SATSP. Its $TF = 1$, and across 3 names it occurs once, so $IDF = \log(3/1) \approx 1.10 \rightarrow TF-IDF \approx 1.10$ (highly distinctive). The token TEMP appears in all three. Its $IDF = \log(3/3) = 0$, so $TF-IDF = 0$ (uninformative). Thus, $TF-IDF$ highlights unique patterns like SAT and RAT, while ignoring common tokens like TEMP.

Concretely, we apply $TF-IDF$ over two complementary tokenization:

1. **Character n-grams (length 3–6)** to capture vendor abbreviations (e.g., SATSP, ZN-T, RAT) and
2. **Word n-grams (1–2)** to retain informative token co-occurrences and capture semantically meaningful combinations (e.g., air temp, zone setpoint).

Once point names are vectorized, we evaluate a suite of supervised classifiers to map feature vectors to canonical sensor-type labels. The $TF-IDF$ vectors feed into a configurable discriminative classifier, chosen at runtime from:

1. **Logistic Regression:** A linear classifier that models the probability of each class using a softmax function. It is well-suited for high-dimensional sparse inputs like $TF-IDF$ vectors and produces interpretable coefficients that indicate the contribution of specific n-grams to sensor type predictions.
2. **Linear Support Vector Machine (LinearSVC):** A maximum-margin classifier optimized for sparse feature spaces. By focusing on the most discriminative boundaries, SVMs often outperform probabilistic models on short-text classification tasks with many overlapping classes.
3. **Random Forest:** An ensemble of decision trees that captures non-linear interactions between tokens. Although less efficient on very high-dimensional vectors, it provides robustness to noisy features and a measure of feature importance.
4. **Gradient Boosting Models (XGBoost, LightGBM):** High-capacity ensembles that iteratively correct misclassifications. These models are particularly effective in handling class imbalance and subtle abbreviation differences by learning complex token–class relationships.

Classifiers output a sensor-type label, which is standardized using a **LabelEncoder** to assign stable integer indices and maintain consistency across training and inference. The entire workflow—combining preprocessing, feature extraction, and classification—is encapsulated in a scikit-learn Pipeline and persisted with joblib, allowing models to be deployed and reused without retraining.

To address the imbalance common in building datasets, classifiers are configured with **balanced class weights** where supported, ensuring that underrepresented sensor types influence the learning process proportionally. Additionally, sensor types with extremely low support (fewer than two samples) are pruned during training to avoid unstable decision boundaries. The outcome of this stage is a **normalized sensor-type tag**—for example, mapping SATSP to *supply air temperature setpoint*—which provides a consistent intermediate representation. These

normalized outputs then serve as the input to the next stage: **ontology-driven semantic matching**, where they are aligned with standardized Brick classes.

3.2.1.2 Leveraging Pre-Trained Language Models

A key limitation of supervised classification is its dependence on labeled training data. While annotated corpora exist, they cannot capture the diversity of naming conventions across all buildings, vendors, and regions. As a result, purely supervised models risk overfitting to the training distribution and failing when exposed to unseen abbreviations or token patterns.

To address the limitations of supervised approaches, we turn to pre-trained foundational language models. Unlike supervised classifiers, pretrained large language models/ foundational models are trained on massive, diverse text corpora and learn representations that are inherently more generalizable. As a result, they can support zero-shot and few-shot learning, where accurate predictions are possible even for abbreviations or naming conventions not seen during training. By transferring this broad linguistic knowledge into the building domain, pre-trained models provide a scalable way to handle the variability and incompleteness of real-world point names.

In our workflow, we used foundational models to **normalize raw point names** into **canonical sensor-type candidates**. Instead of directly classifying abbreviations, the model is prompted to interpret them in plain language. In this approach, the model is treated as a translator: it is given a raw point name and asked, through a carefully designed natural-language instruction, to provide its interpretation in plain text. For example, when given the input ZN-T, the model returns *zone temperature sensor* as its normalized form. Similarly, SATSP can be mapped to *supply air temperature setpoint*. This avoids the need for extensive labeled training data and leverages the broad knowledge embedded in foundational models.

This method is a form of **prompt engineering**, where the power of the model comes not from additional training but from carefully crafted instructions that steer its outputs. Alternatives to prompt engineering include **Retrieval-Augmented Generation (RAG)** and the **Model Context Protocol (MCP)**. RAG integrates an external knowledge base (e.g., the Brick ontology) into the model's context window, ensuring responses are directly grounded in curated data. MCP, on the other hand extends this idea further by allowing structured data and APIs to be connected to the model, providing stronger guarantees of accuracy and control. While both RAG and MCP offer robust grounding, they require more infrastructure and integration effort. In contrast, our lightweight prompt-based approach is simple, fast to deploy, and sufficient for the point-name normalization task, especially since downstream semantic matching with Brick still provides a strong constraint on the model's outputs.

Our prompt is designed to guide the model toward producing consistent, ontology-friendly outputs. For example, the system uses instructions such as:

""""

You are helping to expand compressed BACnet point names into readable phrases.

Rules:

- Use ONLY words from the provided vocabulary (case-insensitive) and digits that appear in the name (e.g., 2 in CO2 if present).
- Do NOT invent new words.
- Prefer concise noun phrases (e.g., "zone temperature setpoint", "supply air temperature sensor").
- Output ONE short phrase, no punctuation except spaces, no quotes, no extra lines.

Vocabulary:

{vocab_for_prompt}

BACnet name:

{bacnet_name}

""

These normalized descriptions then serve as intermediate representations that are passed into the semantic matching stage, where they are aligned with valid Brick classes. This use of LLMs effectively acts as a translator between vendor-specific abbreviations and ontology-compliant terms, allowing the system to handle unfamiliar or site-specific naming conventions without requiring them to appear in the training data.

Ontology-Constrained Expansion

When relying only on foundational models to interpret abbreviations, the system can produce false positives because many abbreviations in building data are ambiguous. For example, the token SP may represent setpoint in one context and static pressure in another; RAT might mean return air temperature or return air terminal. Left unconstrained, the model may generate a valid sounding but incorrect expansion, leading to errors downstream.

To address this, we constrain the model's outputs using the **Brick ontology**. Specifically, the {vocab_for_prompt} placeholder in the prompt is populated with terms drawn directly from Brick classes and properties. This forces the model to choose only from schema-compliant words when expanding abbreviations. For example, when expanding SP, the prompt restricts the output to Brick-defined terms such as *setpoint* or *static pressure*, rather than allowing the model to invent unrelated interpretations. By grounding the model in ontology vocabulary, we substantially reduce the chance of spurious matches while still preserving the model's ability to generalize to unseen abbreviations.

Hinting with Building Data

Even with ontology constraints, foundational models can still fail when encountering building-specific abbreviations that are not widely recognized. For example, a site may use RHT to denote a *reheat coil temperature sensor*, but this abbreviation is unlikely to appear in generic corpora or the Brick vocabulary. Without additional context, the model may misinterpret or default to a generic class.

To overcome this limitation, we incorporate hinting, where **building-specific knowledge** is embedded directly into the model prompt. In this setup, the {vocab_for_prompt} placeholder

is extended beyond Brick classes to include common building automation terms (e.g., *zone*, *temperature*, *sensor*, *supply*, *setpoint*) as well as curated site-specific expansions. At the same time, the raw abbreviation is passed through the `{bacnet_name}` placeholder, ensuring that the model interprets the actual point name in context. By providing both controlled vocabulary and examples of local conventions, the model learns to correctly normalize site-specific abbreviations. For instance, if RHT is introduced as *reheat coil temperature sensor* in the prompt, subsequent occurrences of RHT are normalized consistently. Once validated, these mappings are fed back into the vocabulary, improving the system’s robustness across different buildings and vendors.

Together, these strategies harness the generalization power of foundational models while embedding them in domain-specific constraints, enabling scalable, accurate, and ontology-compliant normalization of point names across diverse building systems. The normalized outputs produced through this combination of foundational models, ontology constraints, and building-specific hinting then serve as inputs to the semantic matching stage, where they are formally aligned with Brick classes to complete the tagging process.

3.2.2 Semantic Matching

Once point names have been normalized into canonical sensor-type candidates, the next step is to align them with standardized Brick classes through ontology-driven semantic matching. This stage ensures that every point is not only linguistically normalized but also formally integrated into the Brick schema. Candidate phrases produced by classifiers or foundational models are compared against the Brick vocabulary using a combination of lexical similarity (e.g., token overlap, edit distance) and semantic similarity (embedding-based cosine measures). By constraining the final assignments to valid Brick entities, the system guarantees that normalized outputs are mapped into a structured, interoperable representation suitable for downstream applications such as building analytics, fault detection, and control optimization.

To enable matching, we first prepare the Brick search space and then evaluate each normalized phrase against it with one of several matchers. The Brick schema (in TTL format) is parsed into a graph, from which we collect all subclasses of `brick:Point` and generate human-readable labels (for example, `Supply_Air_Temperature_Setpoint` becomes “*supply air temperature setpoint*”). These labels are tokenized and stored together with their URIs and ancestor chains so that later steps can both identify the correct class and, if needed, “roll up” to a higher-level category.

3.2.2.1 Generate Semantic Embeddings

The actual comparison is carried out using semantic embeddings generated by a pre-trained Sentence-BERT (SBERT) model, a widely used variant of BERT fine-tuned for sentence similarity tasks. Each Brick label and each normalized phrase is encoded into a dense vector representation in a high-dimensional space (typically 384–768 dimensions, depending on the model variant). In this space, phrases with similar meaning lie close together even if their wording differs. For instance, “zone temp sensor” and “zone temperature sensor” produce embeddings with a cosine similarity close to 1.0, while unrelated terms like “static pressure setpoint” are much farther apart. To improve efficiency, embeddings for all Brick labels are

precomputed and cached at initialization, so that only the incoming point phrases need to be encoded during inference.

Once embeddings are generated, the next task is to measure how close each normalized point name is to the available Brick classes. This step transforms high-dimensional vectors into actionable similarity scores, allowing us to identify the most appropriate Brick class for every input.

3.2.2.2 Similarity Matching

Similarity matching is the process of comparing the embedding of a normalized point name against precomputed embeddings of all Brick point classes, using several distance and similarity metrics to measure alignment. Cosine similarity and dot product capture directional alignment in the embedding space, while Euclidean and Manhattan distances measure absolute proximity. Building on these measures, we explore two families of matchers: a single-stage approach, which directly applies one of the metrics to perform a nearest-neighbor search across all Brick classes, and a two-stage ontology matcher, which first retrieves a small candidate set using lexical search (BM25) and then reranks them with semantic embeddings. We next discuss both these approaches in detail.

Single-Stage Embedding Matchers (Cosine / Dot / Euclidean / Manhattan)

In the single-stage family, every Brick class label is embedded once using a Sentence-BERT model, and these embeddings are cached for efficient reuse. At inference time, the normalized input phrase is encoded into the same vector space and compared against all Brick label embeddings using one of several distance or similarity metrics. This produces a score for each class, and the class with the highest score is selected as the match. Because the Brick label embeddings are precomputed, this approach reduces to a nearest-neighbor search in vector space, making it both fast and effective when normalized phrases are close in form to canonical Brick labels.

We evaluate four metrics for computing similarity:

- **Cosine similarity** measures the angle between two vectors:

$$\text{cosine}(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$$

This metric is scale-invariant and emphasizes directional alignment, which is useful when embeddings capture meaning regardless of magnitude. For example, “zone temp sensor” and “zone temperature sensor” produce embeddings with high cosine similarity because they point in nearly the same direction in the vector space.

- **Dot product** directly measures the unnormalized inner product between two vectors:

$$\text{dot}(x, y) = x \cdot y$$

Unlike cosine, this incorporates both direction and magnitude. It can reward stronger signals when vectors are not only aligned but also have higher norms.

- **Euclidean distance** measures straight-line (L2) distance in the embedding space:

$$d_E(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

Here, smaller distances indicate stronger matches. This metric emphasizes absolute proximity in the vector space, making it sensitive to both alignment and vector magnitude.

- **Manhattan distance** (L1 distance) measures the sum of absolute differences across dimensions:

$$d_M(x, y) = \sum_i |x_i - y_i|$$

This metric is less sensitive to outliers in individual dimensions and can provide robustness in high-dimensional embeddings.

In practice, cosine similarity and dot product are angle-aware and tend to work best for short text similarity, while Euclidean and Manhattan distances emphasize geometric closeness. The matcher returns the top-ranked Brick class together with its URI, score, processing time, and full ontology hierarchy. An optional **cap-depth** can be applied to roll the selection up to a higher-level ancestor class for more general mappings, while still recording the original match.

Two-Stage Ontology Matcher (BM25 Candidate Generation → SBERT Rerank + Tag Consistency)

While single-stage embedding matchers work well when normalized point names are already close to canonical Brick labels, they can struggle with shorter, noisier, or highly abbreviated inputs. In such cases, the embedding alone may not provide enough context for a reliable match. To address this, we employ a two-stage ontology matcher that combines lexical retrieval with semantic reranking. This hybrid design narrows the candidate space to the most relevant terms before applying deeper semantic comparison, improving robustness when dealing with ambiguous or abbreviation-heavy point names.

In **Stage 1 (candidate generation)**, we construct a tokenized corpus of Brick labels and apply BM25 (Okapi) retrieval to identify the top-k most relevant candidates (default ~20). BM25 is a bag-of-words retrieval function that scores each document D with respect to a query Q as:

$$BM25(Q, D) = \sum_{t \in Q} IDF(t) \cdot \frac{f(t, D) \cdot (k_1 + 1)}{f(t, D) + k_1 \cdot (1 - b - b \cdot \frac{|D|}{avgdl})}$$

where:

- $f(t, D)$ is the frequency of term t in document D ,
- $|D|$ is the length of document D ,
- $avgdl$ is the average document length across the corpus,
- k_1 and b are hyperparameters (commonly $k_1 \approx 1.2, b \approx 0.75$), and
- $IDF(t)$ is the inverse document frequency of term t .

Intuitively, BM25 rewards terms that are distinctive in the corpus (via IDF) and frequent in the candidate label (via $f(t, D)$), while normalizing for document length. This ensures that short but informative labels (like “supply air temp setpoint”) are not overshadowed by longer ones. In our setup, BM25 typically retrieves the top-k candidates (default $k=20$) for reranking. When BM25 is unavailable, we fall back to a lightweight IDF-weighted scheme that approximates this scoring.

In **Stage 2 (semantic reranking)**, only the k retrieved candidates are passed to a bi-encoder model (Sentence-BERT). Both the query phrase and the candidate labels are encoded into semantic embeddings, normalized to unit length, and compared using cosine similarity. This reranking step ensures that the most semantically appropriate class among the retrieved candidates is selected. To enrich the query, we also infer lightweight tags from the normalized point name (e.g., “supply,” “air,” “temperature,” “setpoint”) and append them to the query string (e.g., ... [TAGS] supply air temperature setpoint) before encoding, providing additional context for disambiguation.

After reranking, we apply a light ontology-consistency check. If the highest-scoring label is inconsistent with the inferred tag set, we scan the next few candidates within a small similarity margin (ϵ) and select the first tag-consistent option. This safeguard helps avoid errors when two candidates are semantically close but only one matches the expected tag profile. Finally, as with the single-stage matcher, the selected class can be rolled up to a higher-level ancestor in the ontology using an optional cap-depth, allowing greater flexibility for reporting while still preserving the original match in the results.

Example

For the normalized phrase “supply air temperature setpoint,” BM25 typically retrieves candidates such as “supply air temperature setpoint”, “zone air temperature setpoint”, and “return air temperature setpoint”. The SBERT reranker assigns the highest similarity to “supply air temperature setpoint”, and the tag gate approves this choice because all required concepts are present. The assigned Brick class is `brick:Supply_Air_Temperature_Setpoint`. If a cap-depth is configured, this can be rolled up to a higher ancestor while the system still records the leaf match, URI, score, hierarchy, and processing time in the result.

In conclusion, our methodology combines ontology-guided reasoning with AI-driven inference to transform raw BACnet point names into Brick-compliant semantic models. The workflow progresses through automated mapping to establish site, building, equipment, and point

hierarchies; AI-based point normalization to standardize heterogeneous naming conventions; and ontology-constrained semantic matching to assign each point to the most appropriate Brick class. By integrating machine learning classifiers, foundational language models, and hybrid similarity matchers, the approach balances efficiency with robustness across a wide range of naming patterns. Figure 1 also shows a third pipeline at the bottom, which is the validation on ground truth data collected from PNNL. Specifically, we compare (1) the full automated pipeline—from raw point name → AI-based point normalization → semantic matching—against (2) a path that uses SME-provided normalized names directly in semantic matching. The resulting Brick classes from these two paths are then compared to quantify how closely the automated pipeline replicates expert tagging. The next section will discuss this in detail.

4.0 Case Study

We deployed the proposed semantic tagging framework across eight PNNL-operated buildings (names anonymized). The sites represented diverse sizes, system complexities, and BACnet vendor implementations, with point counts ranging from fewer than 100 to nearly 3,000. Such heterogeneity reflects the broader landscape of commercial building operations, where inconsistent naming conventions and proprietary design choices make semantic interoperability especially difficult.

Table 1. Deployment Summary

Building	# of BACNet Point Names	Vendor
B1	320	JCI
B2	71	JCI
B3	1,696	Alerton
B4	308	Alerton
B5	2,997	JCI
B6	549	JCI
B7	586	JCI
B8	902	JCI

The goal of this case study is to evaluate whether our ontology-guided, AI-enabled framework can reliably transform raw BACnet metadata into Brick-compliant models under these real-world conditions. In particular, we aim to demonstrate (i) scalability to thousands of point names across multiple facilities, (ii) robustness to vendor-specific naming practices, and (iii) integration-readiness for downstream analytics platforms.

BACnet metadata was collected through VOLTTRON, an open-source, agent-based platform for distributed sensing and control in building and grid applications. VOLTTRON's BACnet driver provided a uniform interface to extract point metadata despite vendor-specific differences. The extracted data from each building included the following columns:

- **Point Name:** Native BACnet label as defined in the building automation system.
- **Volttron Point Name:** Canonicalized identifier used within the VOLTTRON framework.
- **Units / Unit Details:** Engineering units (e.g., °F, CFM) and expanded descriptions.
- **BACnet Object Type:** The object category (e.g., analogInput, binaryValue).

- **Property:** The property within the BACnet object (e.g., presentValue).
- **Writable:** Boolean flag indicating if the point is commandable.
- **Index:** Numeric index assigned for efficient lookup.
- **Write Priority:** Priority level for writeable points (per BACnet standard).
- **Notes:** Additional vendor-specific or site-specific annotations.

In our workflow, the **Point Name** column served as the raw input for inference, while the **Volttron Point Name** was used as the canonical reference for ground truth validation.

With building metadata collected and standardized, point names from each building were then processed through the automated pipeline:

1. **Ontology-guided parsing** to establish site, building, equipment, and point hierarchies.
2. **AI-based normalization** to resolve diverse abbreviations (e.g., ZN-T, SATSP, CO2-SNSR).
3. **Ontology-constrained semantic matching** to assign the most appropriate Brick class.

To assess the accuracy of our approach, we benchmarked the automated pipeline against SME-curated ground truth mappings. The goal of this validation was to determine how closely the end-to-end automated process could replicate expert tagging, and to isolate the role of the normalization step in enabling robust semantic matching. We therefore evaluated performance under two configurations:

- **Fully automated path** – raw point name → normalization → semantic matching.
- **SME-assisted path** – SME-normalized names → semantic matching.

For model training and to support retrieval-augmented prompting of the foundational language model, we used the Mortar dataset. Mortar is a large-scale repository of real-world building metadata, designed specifically to advance semantic modeling research in the building domain. It consists of hundreds of building metadata instances encoded as Brick graphs, where BACnet-style point names are mapped to standardized Brick classes. The dataset is distributed in TTL (Terse RDF Triple Language) format, each file containing a graph with equipment, point hierarchies, and semantic annotations.

This dataset was particularly valuable for two reasons. First, it offers diverse and heterogeneous point naming conventions collected from different sites, making it well-suited for training classifiers to generalize beyond a single building or vendor. Second, because Mortar graphs already encode validated Brick mappings, they provide a reliable source of ground truth labels that can be directly used for supervised learning. We parsed the TTL files to construct a corpus of (point name, Brick class) pairs, which served as both ground truth for model training and context for the foundational model during inference.

5.0 Results and Discussion

All experiments were conducted on a MacBook Pro with an Apple M3 Max chip, featuring 16 CPU cores and a 40-core integrated GPU. Both classifier training and pipeline execution were performed on the CPU. The pipeline was implemented in Python 3.13, with classifiers developed using scikit-learn (v1.7) and XGBoost (v3.0.3). For semantic matching, we employed the all-mpnet-base-v2 embedding model from sentence-transformers (v5.0) in combination with a CosineSimilarityMatcher, which measured similarity between normalized point names and candidate Brick classes. The Brick ontology (v1.4.4) served as the reference schema, providing a standardized vocabulary for equipment, sensors, setpoints, and control points to ensure consistent alignment across heterogeneous BACnet point names. For foundational model inference, we used ollama (v0.5.1) with llama3 as the base language model, supporting retrieval-augmented prompting.

5.1 Trained Classifier with Semantic Matching

To evaluate the performance of trained classifiers in combination with semantic matching, we applied four models—XGBoost, Logistic Regression, Random Forest, and Linear SVC—to point name data from eight anonymized PNNL-operated buildings. Importantly, the classifiers were trained exclusively on the Mortar dataset and tested only on the PNNL buildings, ensuring that evaluation reflects generalization to previously unseen naming conventions.

For each building, the raw point names (shown in brackets in

Table 2) were first processed by the classifier to produce normalized labels, which were then passed to the all-mpnet-base-v2 embedding model with a CosineSimilarityMatcher for final Brick class assignment. The predicted Brick classes were then compared against the SME-curated mappings referenced through the Volttron Point Names, which served as the ground truth for evaluation.

Table 2. Classifier Accuracy in Mapping BACnet Point Names to Brick Classes Across Eight PNNL Buildings

Classifier	B1 (320)	B2 (71)	B3 (1696)	B4 (308)	B5 (2997)	B6 (549)	B7 (586)	B8 (902)
XGBoost	53	11	103	8	734	68	72	222
Logistic	76	18	303	0	656	77	81	271
Random Forest	84	13	248	24	945	104	110	349
Linear SVC	75	17	301	20	1021	116	126	425

Across all eight buildings, the best-performing model (Linear SVC) achieved an overall accuracy of 28.3% when evaluated against this ground truth. Performance varied by building size and complexity: the hardest building to infer was B4 (6.5%), where naming conventions were particularly ambiguous, while the easiest building was B8 (47.1%), where point names were relatively consistent and yielded the highest proportion of correct mappings.

These results highlight three key observations:

- **Generalization:** Despite being trained only on Mortar, the classifiers transferred effectively to the PNNL dataset, underscoring the value of Mortar’s heterogeneous point names for robust training.
- **Scale sensitivity:** Models with greater capacity to capture non-linear relationships (RF, Linear SVC) generalized more effectively in larger, more complex buildings.
- **Model robustness:** Linear SVC delivered stable performance across buildings of different sizes, while XGBoost underperformed—likely due to the limited signal available in short, highly abbreviated point names.

5.2 Key Findings on Model Limitations

Error analysis revealed several consistent failure modes when applying classifier-based normalization with semantic matching. Representative examples are discussed below.

- **Inference Errors from Training Data Coverage:** In several cases, the inferred point names were constrained by the abbreviations and class labels available in the training data. For example, *Cooling Stand By Temperature Set Point* and *Unoccupied Cooling Temperature Set Point* (ground truth: *Cooling_Temperature_Setpoint*) were both misclassified as exhaust air static pressure setpoint and mapped to the *Static_Pressure_Setpoint* class. These errors suggest that the model struggles when training data lacks coverage of less common setpoint variations, forcing it to fall back on more frequently seen but semantically incorrect classes.
- **Close Calls in Semantically Similar Cases:** The model often produced outputs that were close to the intended meaning but not exact matches. For instance, *Occupancy Mode* (*Occupied_Mode_Status*) was mapped to *occupancy status* (*Occupancy_Status*), and *Occupancy Override* (*Occupancy_Command*) was mapped to *temporary occupancy* (*Temporary_Occupancy_Status*). These “near misses” highlight the difficulty of distinguishing between closely related operational states, especially when abbreviations (e.g., OCC) are overloaded across multiple contexts.
- **Failures with Highly Specific Naming:** Points with highly specific or idiosyncratic names led to incorrect mappings. For example, *Enable* (*Run_Enable_Command*) was inferred as *vav box mode* (*Box_Mode_Command*), and *Electric Heat Status* (*Heating_Mode_Status*) was misclassified as *chilled water pump vfd command* (*System_Enable_Command*) by the Subject Matter Expertise. These errors illustrate the

limitations of the model when encountering rare, site-specific naming conventions that deviate significantly from the broader patterns represented in Mortar.

These findings indicate that errors are most common in three situations: limited coverage of rare setpoints, difficulty separating closely related modes, and misalignment on unique site-specific terms. We next tried fixing errors related to the first category by introducing foundational models. Classifiers trained only on Mortar are constrained to the abbreviations and classes seen during training, which leads to systematic misclassifications when novel point types appear (e.g., mapping all unfamiliar temperature setpoints to `Static_Pressure_Setpoint`). Foundational models offer a way to address this limitation by leveraging their broader language understanding. By decomposing unseen abbreviations into meaningful tokens and reasoning over their semantic relationships (e.g., “cooling,” “standby,” and “setpoint”), foundational models can infer the intended meaning even when the exact term was absent from the training corpus, thereby reducing errors caused by coverage gaps.

5.3 Foundational Model with Semantic Matching

Table 3 compares performance between the best classifier (Linear SVC) and the foundational model across the eight anonymized buildings. Values in parentheses indicate the total number of point names per building, while the reported values are the number of correctly mapped point names.

Table 3. Comparison of classifier (Linear SVC) and foundational model (Llama3) performance on PNNL buildings

Classifier	B1 (320)	B2 (71)	B3 (1696)	B4 (308)	B5 (2997)	B6 (549)	B7 (586)	B8 (902)
Linear SVC	75	17	301	20	1021	116	126	425
FM (LLAMA3)	74	26	737	46	587	152	162	190

The foundational model achieved an overall accuracy of 26.6%, compared to 28.3% for the best classifier (Linear SVC). While its aggregate performance was slightly lower, the distribution across buildings highlights complementary strengths.

- **Generalization on smaller and mid-sized buildings:** The foundational model outperformed Linear SVC in five of the eight buildings, most notably in B3 (43.5% vs. 17.7%), B2 (36.6% vs. 23.9%), and B4 (14.9% vs. 6.5%). These gains show the model’s ability to interpret abbreviations and unseen naming patterns.
- **Weaker on large, homogeneous datasets:** In B5 (2,997 points) and B8 (902 points), where naming conventions were more consistent, Linear SVC retained a clear advantage (34.1% vs. 19.6%, and 47.1% vs. 21.1%).

- **Balanced performance in medium buildings:** The foundational model achieved higher accuracies in B6 and B7, indicating robustness where naming conventions are varied but not as extensive as in B5 or B8.

Together, these results suggest that classifiers remain effective where naming patterns are well represented in training data, while foundational models excel when facing unseen abbreviations or sparse contexts. On further analysis, we found that the key reason behind the poor performance of the foundational model is that, while it benefits from broad language knowledge, it lacks building-specific context. Point names in BACnet systems are highly localized—often reflecting site-specific conventions, abbreviations, or legacy practices—and without that grounding, the model tends to overgeneralize based on patterns seen in general language. As a result, its predictions were sometimes semantically reasonable but misaligned with the intended building-specific Brick class.

5.4 Context-Aware Inference with Retrieval-Augmented Prompting

To address the lack of building-specific context observed in the foundational model, we introduced retrieval-augmented prompting (RAP). RAP enhances model inference by retrieving relevant examples from a domain-specific knowledge base—in this case, the Mortar dataset—and appending them to the model prompt. These examples act as contextual hints that guide the model toward correct mappings when faced with novel or ambiguous point names. By grounding the model’s predictions in building-domain exemplars, RAP helps bridge the gap between the model’s broad language knowledge and the highly specific conventions used in BACnet point naming.

Table 4. Results of semantic tagging using Linear SVC, Llama3, and Llama3 with retrieval-augmented prompting (RAP) across eight anonymized buildings.

Classifier	B1 (320)	B2 (71)	B3 (1696)	B4 (308)	B5 (2997)	B6 (549)	B7 (586)	B8 (902)
Linear SVC	75	17	301	20	1021	116	126	425
FM (LLAMA3)	74	26	737	46	587	152	162	190
FM with RAP (LLAMA3)	85	30	693	34	1085	210	229	277

Table 4 reports the number of correctly mapped point names across the eight anonymized buildings for three approaches: Linear SVC, the foundational model without hints, and the foundational model with hints (implemented through RAP).

Overall, retrieval-augmented prompting (RAP) achieved an accuracy of 35.6%, compared to 26.6% for the foundational model without hints and 28.3% for Linear SVC. This corresponds to a 9.0 percentage point improvement over the baseline foundational model and a 7.3 percentage point improvement over the best classifier. Key observations include:

- **Substantial improvements in large buildings:** RAP boosted performance in B5 (36.2% vs. 19.6% without hints) and B8 (30.7% vs. 21.1%), outperforming both the classifier and the baseline foundational model.
- **Consistent gains across small and mid-sized buildings:** Even in smaller datasets such as B2 and B4, accuracy improved (42.3% vs. 36.6% and 11.0% vs. 14.9%).
- **Overall performance lift:** Averaged across all buildings, RAP achieved higher accuracy than either Linear SVC (28.3%) or the foundational model without hints (26.6%), demonstrating the importance of context in inference.

These results confirm that lack of context was the main driver of foundational model errors, and that RAP provides an effective mechanism to inject domain-specific knowledge, leading to significant improvements across both small and large building datasets.

6.0 Discussion and Future Work

This study explored automated semantic tagging of BACnet point names into Brick classes using three approaches: classifiers trained on Mortar, a foundational model (Llama3), and the foundational model augmented with retrieval (RAP). The results highlight both the promise and the challenges of applying machine learning and foundation models to this domain.

Classifiers performed reliably in cases where naming conventions were well represented in the Mortar dataset, particularly in large buildings with consistent vendor practices. Their limitations became clear in settings with unseen or rare abbreviations, where coverage gaps in the training corpus led to systematic misclassifications. Foundational models, by contrast, showed strong generalization to novel abbreviations and mid-sized datasets but underperformed in large, homogeneous buildings. This outcome reflects their strength in semantic reasoning but also their weakness when deprived of building-specific context.

RAP directly addressed this shortcoming. By injecting domain-specific exemplars at inference time, it provided the missing contextual grounding that allowed the foundational model to reduce systematic errors. RAP achieved the highest overall accuracy, outperforming both classifiers and the baseline foundational model, and proved particularly effective in large buildings where the baseline model struggled. These findings confirm that context, rather than capacity alone, is the limiting factor in robust semantic tagging.

Looking ahead, several directions could extend this work. A key direction for future work lies in contextual enrichment. Inference in this study was based only on point names, but building metadata offers additional context that could reduce errors, particularly in distinguishing semantically similar states. Incorporating information such as equipment type, spatial hierarchy, and time-series measurements could provide grounding for ambiguous abbreviations. Contextual enrichment can also be realized through retrieval-augmented prompting (RAP) by explicitly including examples of highly specific naming conventions. This strategy directly targets the third type of error observed—misclassifications of unique, site-dependent terms—and could help foundational models learn to generalize across heterogeneous vendor practices. Expanding the training corpus remains important as well, ensuring coverage of underrepresented abbreviations and point types.

Beyond data enrichment, modeling improvements form another promising avenue. Future work could investigate whether retrieval-augmented generation (RAG), which dynamically retrieves domain examples to ground predictions, or approaches such as model contextual prompting (MCP), which enriches prompts with structured metadata like equipment type, hierarchy, or units, can further improve performance. Both methods aim to better couple retrieval or context with inference, reducing overgeneralization and strengthening alignment between point names and Brick classes. Another important question is whether foundational models alone can subsume semantic matching, directly mapping point names to Brick classes without relying on an intermediate embedding similarity layer. Such an approach could simplify the pipeline and reduce potential error propagation between stages.

Transfer learning techniques also offer opportunities: by fine-tuning a foundational model on curated building metadata, the model may better adapt to domain-specific abbreviations and

improve zero-shot generalization. In addition, it will be valuable to explore the use of different types of foundational models. While Llama3 provided a useful baseline, domain-tuned or instruction-optimized models may offer stronger performance on structured metadata tasks.

Finally, we recognize that inferring a single exact class may be too restrictive. A practical alternative would be to generate a ranked set of top-k recommendations (e.g., top 3 classes) for each point name. This not only accommodates uncertainty but also enables subject matter experts to validate results more efficiently. Together, these directions—contextual enrichment, modeling improvements, and flexible output strategies—offer a pathway toward more robust, accurate, and scalable semantic tagging systems for building metadata.

7.0 Conclusion

This work presented and evaluated an automated semantic tagging framework for mapping BACnet point names to Brick classes across eight PNNL-operated buildings. The framework combined classifier-based normalization, embedding-based semantic matching, and foundational models to address the challenges of heterogeneous and vendor-specific naming conventions. Classifiers trained on the Mortar dataset, combined with semantic matching, achieved an overall accuracy of 28.3%, with Linear SVC emerging as the best-performing model. Foundational models, tested using Llama3, showed strengths in generalizing to unseen abbreviations and achieved 26.6% accuracy overall, outperforming classifiers in five of the eight buildings but underperforming in large, homogeneous datasets where classifiers retained an advantage. The introduction of retrieval-augmented prompting (RAP) addressed this limitation by supplying building-domain exemplars as context, raising accuracy to 35.6%. This corresponds to a 9.0 percentage point improvement over the baseline foundational model and a 7.3 percentage point improvement over the best classifier.

These findings confirm that context, rather than model capacity alone, is the main driver of robust semantic tagging. Classifiers remain effective where naming patterns align closely with the training data, while foundational models excel in scenarios with rare or ambiguous point names. RAP bridges these approaches by injecting domain-specific context, significantly improving performance across both small and large building datasets.

While significant challenges remain—such as handling highly specific site-dependent naming conventions and scaling the pipeline to large portfolios—the combination of foundational models with contextual enrichment represents a promising direction. Future research should continue to explore modeling advances, context-aware retrieval strategies, and domain adaptation techniques, ultimately working toward scalable, accurate, and automated tagging systems that enable richer analytics and more interoperable building data infrastructures.

8.0 References

- Almashor, Mahathir, Mashud Rana, John McCulloch, Ashfaqur Rahman, and Subbu Sethuvenkatraman. 2023. “What’s The Point: AutoEncoding Building Point Names.” *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (New York, NY, USA), BuildSys ’23, November 15, 256–60. <https://doi.org/10.1145/3600100.3623748>.
- Aniakor, Miracle, Vinicius V. Cogo, and Pedro M. Ferreira. 2024. “A Survey on Semantic Modeling for Building Energy Management.” arXiv:2404.11716. Preprint, arXiv, April 17. <https://doi.org/10.48550/arXiv.2404.11716>.
- “ASHRAE Standard 223 User Documentation.” 2025. <https://docs.open223.info/intro.html>.
- Balaji, Bharathan, Arka Bhattacharya, Gabriel Fierro, et al. 2016. “Brick: Towards a Unified Metadata Schema For Buildings.” *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments* (New York, NY, USA), BuildSys ’16, November 16, 41–50. <https://doi.org/10.1145/2993422.2993577>.
- Balaji, Bharathan, Chetan Verma, Balakrishnan Narayanaswamy, and Yuvraj Agarwal. 2015. “Zodiac: Organizing Large Deployment of Sensors to Create Reusable Applications for Buildings.” *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments* (New York, NY, USA), BuildSys ’15, November 4, 13–22. <https://doi.org/10.1145/2821650.2821674>.
- Ben Ahmed, Mohamed, Anouar Abdelhakim Boudhir, Rani El Meouche, and İsmail Rakıp Karaş, eds. 2024. *Innovations in Smart Cities Applications Volume 7: The Proceedings of the 8th International Conference on Smart City Applications, Volume 2*. Vol. 938. Lecture Notes in Networks and Systems. Springer Nature Switzerland. <https://doi.org/10.1007/978-3-031-54376-0>.
- Bergmann, Harry, Cory Mosiman, Avijit Saha, et al. 2020. *Semantic Interoperability to Enable Smart, Grid-Interactive Efficient Buildings*. November 1. <https://escholarship.org/uc/item/1325d5j3>.
- Bhattacharya, Arka A., Dezhi Hong, David Culler, Jorge Ortiz, Kamin Whitehouse, and Eugene Wu. 2015. “Automated Metadata Construction to Support Portable Building Applications.” *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, November 4, 3–12. <https://doi.org/10.1145/2821650.2821667>.
- Bhattacharya, Arka, David E Culler, Jorge Ortiz, Dezhi Hong, and Kamin Whitehouse. 2014. *Enabling Portable Building Applications through Automated Metadata Transformation*.
- “Brickify Tool.” 2025. <https://brickschema.readthedocs.io/en/latest/brickify/index.html>.
- Butzin, Björn, Frank Golasowski, and Dirk Timmermann. 2017. “A Survey on Information Modeling and Ontologies in Building Automation.” *IECON 2017 - 43rd Annual*

Conference of the IEEE Industrial Electronics Society, October, 8615–21.
<https://doi.org/10.1109/IECON.2017.8217514>.

- Chamari, Lasitha, Joep van der Weijden, Lolke Boonstra, Stefan Hoekstra, Ekaterina Petrova, and Pieter Pauwels. 2023. “Metadata Schema Generation for Data-Driven Smart Buildings: 11th Linked Data in Architecture and Construction Workshop, LDAC 2023.” *Proceedings of the 11th Linked Data in Architecture and Construction Workshop*, CEUR Workshop Proceedings, March 30, 139–50.
- Daniele, Laura, Frank den Hartog, and Jasper Roes. 2015. “Created in Close Interaction with the Industry: The Smart Appliances REference (SAREF) Ontology.” In *Formal Ontologies Meet Industry*, edited by Roberta Cuel and Robert Young. Springer International Publishing. https://doi.org/10.1007/978-3-319-21545-7_9.
- Fierro, Gabe, Sriharsha Guduguntla, and David E. Culler. 2019. “Dataset: An Open Dataset and Collection Tool for BMS Point Labels.” *Proceedings of the 2nd Workshop on Data Acquisition To Analysis* (New York, NY, USA), DATA’19, November 10, 40–42.
<https://doi.org/10.1145/3359427.3361922>.
- Fierro, Gabe, Jason Koh, Yuvraj Agarwal, Rajesh K. Gupta, and David E. Culler. 2019. “Beyond a House of Sticks: Formalizing Metadata Tags with Brick.” *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (New York, NY, USA), BuildSys ’19, November 13, 125–34.
<https://doi.org/10.1145/3360322.3360862>.
- Fierro, Gabe, Jason Koh, Shreyas Nagare, et al. 2020. “Formalizing Tag-Based Metadata With the Brick Ontology.” *Frontiers in Built Environment* 6 (September).
<https://doi.org/10.3389/fbuil.2020.558034>.
- Fierro, Gabe, Marco Pritoni, Moustafa Abdelbaky, et al. 2019. “Mortar: An Open Testbed for Portable Building Analytics.” *ACM Trans. Sen. Netw.* 16 (1): 7:1-7:31.
<https://doi.org/10.1145/3366375>.
- Gao, Jingkun, Joern Ploennigs, and Mario Berges. 2015. “A Data-Driven Meta-Data Inference Framework for Building Automation Systems.” *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments* (New York, NY, USA), BuildSys ’15, November 4, 23–32.
<https://doi.org/10.1145/2821650.2821670>.
- Hong, Dezhi, Jorge Ortiz, Arka Bhattacharya, and Kamin Whitehouse. 2015. “Sensor-Type Classification in Buildings.” arXiv:1509.00498. Preprint, arXiv, September 1.
<https://doi.org/10.48550/arXiv.1509.00498>.
- Hong, Dezhi, Hongning Wang, Jorge Ortiz, and Kamin Whitehouse. 2015. “The Building Adapter: Towards Quickly Applying Building Analytics at Scale.” *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments* (New York, NY, USA), BuildSys ’15, November 4, 123–32.
<https://doi.org/10.1145/2821650.2821657>.

- Hong, Dezhi, Hongning Wang, and Kamin Whitehouse. 2015. “Clustering-Based Active Learning on Sensor Type Classification in Buildings.” *Proceedings of the 24th ACM International Conference on Information and Knowledge Management* (New York, NY, USA), CIKM ’15, October 17, 363–72. <https://doi.org/10.1145/2806416.2806574>.
- Jiao, Yang, Jiacheng Li, Jiaman Wu, Dezhi Hong, Rajesh Gupta, and Jingbo Shang. 2020. “SeNsER: Learning Cross-Building Sensor Metadata Tagger.” In *Findings of the Association for Computational Linguistics: EMNLP 2020*, edited by Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.findings-emnlp.85>.
- Koh, Jason, Bharathan Balaji, Dhiman Sengupta, Julian McAuley, Rajesh Gupta, and Yuvraj Agarwal. 2018. “Scrabble: Transferrable Semi-Automated Semantic Metadata Normalization Using Intermediate Representation.” *Proceedings of the 5th Conference on Systems for Built Environments* (New York, NY, USA), BuildSys ’18, November 7, 11–20. <https://doi.org/10.1145/3276774.3276795>.
- Koh, Jason, Dezhi Hong, Rajesh Gupta, Kamin Whitehouse, Hongning Wang, and Yuvraj Agarwal. 2018. “Plaster: An Integration, Benchmark, and Development Framework for Metadata Normalization Methods.” *Proceedings of the 5th Conference on Systems for Built Environments* (New York, NY, USA), BuildSys ’18, November 7, 1–10. <https://doi.org/10.1145/3276774.3276794>.
- Lee, Sangkeun, Borui Cui, Mahabir S. Bhandari, and Piljae Im. 2023. “Visual Brick Model Authoring Tool for Building Metadata Standardization.” *Automation in Construction* 156 (December): 105122. <https://doi.org/10.1016/j.autcon.2023.105122>.
- Lee, Sangkeun (Matt), Borui Cui, Mahabir Bhandari, Na Luo, and Piljae Im. 2022. *VizBrick: A GUI-Based Interactive Tool for Authoring Semantic Metadata for Building Datasets*. Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States). <https://www.osti.gov/biblio/1997805>.
- Lin, Lu, Zheng Luo, Dezhi Hong, and Hongning Wang. 2019. “Sequential Learning with Active Partial Labeling for Building Metadata.” *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, November 13, 189–92. <https://doi.org/10.1145/3360322.3360866>.
- “Mapped | Connectors | BACnet.” 2025. <https://www.mapped.com/connectors/bacnet>.
- Mishra, Sakshi, Andrew Glaws, Dylan Cutler, et al. 2020. “Unified Architecture for Data-Driven Metadata Tagging of Building Automation Systems.” *Automation in Construction* 120 (December): 103411. <https://doi.org/10.1016/j.autcon.2020.103411>.
- Mulayim, Ozan Baris, Lazlo Paul, Marco Pritoni, Anand Krishnan Prakash, Malavikha Sudarshan, and Gabe Fierro. 2024. “Large Language Models for the Creation and Use of Semantic Ontologies in Buildings: Requirements and Challenges.” *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, October 29, 312–17. <https://doi.org/10.1145/3671127.3698792>.

- National Renewable Energy Laboratory. (2022) 2025. *NREL/BuildingMOTIF*. Jupyter Notebook. February 16, Released September 24. <https://github.com/NREL/BuildingMOTIF>.
- Perini, Marco, Daniele Antonucci, Rocco Giudice, Marco Savino Piscitelli, and Alfonso Capozzoli. 2025. “BrickLLM: A Python Library for Generating Brick-Compliant RDF Graphs Using LLMs.” *SoftwareX* 30 (May): 102121. <https://doi.org/10.1016/j.softx.2025.102121>.
- “Project Haystack.” 2025. <https://www.project-haystack.org/>.
- Reinisch, Christian, Wolfgang Granzer, Fritz Praus, and Wolfgang Kastner. 2008. “Integration of Heterogeneous Building Automation Systems Using Ontologies.” *2008 34th Annual Conference of IEEE Industrial Electronics*, November, 2736–41. <https://doi.org/10.1109/IECON.2008.4758391>.
- Roa, Carlos Duarte, Paul Raftery, Anand Prakash, and Therese Pepper. 2023. *Field Demonstration of the Brick Ontology to Scale up the Deployment of ASHRAE Guideline 36 Control Sequences*.
- Schumann, Anika, Joern Ploennigs, and Bernard Gorman. 2014. “Towards Automating the Deployment of Energy Saving Approaches in Buildings.” *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings* (New York, NY, USA), BuildSys ’14, November 3, 164–67. <https://doi.org/10.1145/2674061.2674081>.
- Shi, Zixiao, Guy R. Newsham, Long Chen, and H. Burak Gunay. 2019. “Evaluation of Clustering and Time Series Features for Point Type Inference in Smart Building Retrofit.” *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, November 13, 111–20. <https://doi.org/10.1145/3360322.3360839>.
- Vittori, Filippo, Chuan Fu Tan, Anna Laura Pisello, Adrian Chong, and Clayton Miller. 2025. “BIM-to-BRICK: Using Graph Modeling for IoT/BMS and Spatial Semantic Data Interoperability within Digital Data Models of Buildings.” *Energy and Buildings* 348 (December): 116368. <https://doi.org/10.1016/j.enbuild.2025.116368>.
- Waterworth, David, Subbu Sethuvenkatraman, and Quan Z. Sheng. 2021. “Advancing Smart Building Readiness: Automated Metadata Extraction Using Neural Language Processing Methods.” *Advances in Applied Energy* 3 (August): 100041. <https://doi.org/10.1016/j.adapen.2021.100041>.
- Zheng, Zhiyu, Sylvain Marié, Elham Farazdaghi, et al. 2025. “Mastering Building Management Systems Data Points Tagging with Minimal Examples: Unveiling the Power of Large Language Models.” *Energy and Buildings* 328 (February): 115173. <https://doi.org/10.1016/j.enbuild.2024.115173>.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov