Training-Free Element Classification Aligned with BIM-driven Automated Construction Schedule Generation

Steven Heung[1], **Christoph Sydora**[1], **and Eleni Stroulia**[1]

[1]Department of Computing Science, University of Alberta, Canada

syheung@ualberta.ca, csydora@ualberta.ca, stroulia@ualberta.ca

Abstract -

Construction projects are often challenged by time and cost overruns due to the poor quality of their manually constructed schedules. This is why recent research has been exploring how to leverage the Information Model of the to be constructed Building (BIM) to automatically produce a high-quality schedule. The major impediment to this effort is the often poor quality of the building models: typically represented as Industry Foundation Classes (IFC) specifications, exported from building design tools, they often include untyped and mistyped elements, whose construction activities are impossible to infer. To mitigate these issues, this paper proposes an automated relabelling method for assigning a construction-relevant type to each IFC element. In contrast to competitor relabelling methods, our approach is practical and data-efficient, since it does not rely on large labelled datasets for constructing a labelling model. We have validated our method on a dataset of twelve building models with over 20,000 building elements, and our results show that our training-free approach achieves an accuracy of 88.1%, comparable to the state-of-the-art data-reliant methods.

Keywords -

Element Classification, Building Information Modelling, Semantic Enrichment, Automatic Schedule Generation

1 Introduction

Essential to construction management is the project schedule. High-quality schedules are critical for construction projects as project managers depend on them to monitor the project progress and effectively manage timelines, budgets, and resources. An accurate construction schedule should include all the tasks necessary to build the to-be-constructed building, their logical and temporal dependencies, and good estimates of the resources and time required to complete these tasks. A poorly developed schedule can result in significant disruptions, delays, cost overruns, resource conflicts, budget escalation, and labour inefficiencies.

Recent research has attempted to reduce the complexity of construction project scheduling through automation, typically adopting one of two main approaches [1]. Data-driven scheduling utilizes historical project data to identify common task patterns, durations, and dependencies through statistical or machine learning methods [1, 2]. Model-based scheduling derives construction tasks, material and labour estimates, and sequencing dependencies directly from a Building Information Modelling (BIM) [3], typically represented in the Industry Foundation Classes (IFC) format [4].

Model-based scheduling leverages the model of the to-be-constructed building, typically available through the project design and planning phase, to infer the tasks necessary to produce the elements of this envisioned deliverable. This process can be quite effective, provided a complete and semantically accurate BIM model of the to-be constructed building is available. Unfortunately, this input requirement is a fundamentally limiting assumption of model-based scheduling, as real-world models frequently include mistyped elements, i.e., elements with overly generic or even incorrect types [1], resulting in missing or incorrect tasks. For instance, an improper setting in the Revit BIM software can result in the default type `IfcBuildingElementProxy` being used, which has no semantic value [4], and cannot be associated with any construction activity. Even when the IFC types are correctly assigned, their granularity may be too fine for scheduling purposes. For example, distinguishing between `IfcWall` and `IfcWallStandardCase` adds no value to, and may even confuse, construction activity planning, since both would simply correspond to a general wall construction activity.

This paper addresses the shortcomings of poorly typed IFC elements by developing a training-free method for relabelling poorly typed IFC elements. To date, state-of-the-art automatic IFC labelling techniques have been relying on constructing models for inferring the correct IFC object type based on the object's geometry. These methods require data sets of correctly labelled element shapes[5] or element images[6, 7] or the geometric relationships of elements with their adjacent elements[8]. The reliance of these methods on high-quality labelled data for model training is a fundamental shortcoming, given the scarcity of high-quality IFC models.

Instead, the approach we describe in this paper leverages the semantics captured in the object name and descriptions.

Our approach is grounded in the observation that building designers often assign to IFC elements meaningful names and other descriptive properties that can be compared against the documentation of IFC types to identify the type most similar to the object description, and therefore most appropriate for the object. Based on this key observation, three important problems need to be addressed: (i) Which resources capture IFC type documentation? (ii) Which IFC types should be considered? and (iii) How can the most appropriate type for a given object be chosen? Our method adopts the authoritative textual type definitions available in the official IFC schema documentation. To eliminate the noise incurred by the numerous and overlapping IF types, our method puts forward a subset of IFC types, henceforth called *BIM Core* types, that naturally correspond to construction activities. Finally, our method considers two alternative similarity metrics. The first is the Term Frequency-Inverse Document Frequency (TF-IDF) similarity metric, using only the texts of the objects and the type definitions. The second leverages general pretrained language models to map these texts into semantic embeddings (SemEmb) that can be compared against each other.

We have comparatively evaluated these two similarity metrics over the IFC specifications of 12 buildings collected from various sources, demonstrating a wider variety of building element names and descriptive information. Next, we compared our findings against the self-reported accuracies of state-of-the-art methods requiring training, to gauge how well the proposed training-free methods compare with data-reliant state-of-the-art methods.

The contribution of this paper is a training-free method for labelling IFC elements with their correct types, using two metrics for assessing the similarity of an object with a type, evaluated on twelve different building IFC models.

The rest of this paper is organized as follows: Section 2 reviews methods on semantic enrichment with a focus on element type classification and labelling. Section 3 describes the overall relabelling workflow and the two semantic comparison methods, namely TF-IDF and SemEmb. The experimental setup is described in Section 4 followed by the results and discussion in Section 5. Finally, Section 6 summarizes the findings and contributions and puts forth future directions for this work.

2 Related Work

Correcting unlabelled, mislabelled or overly generic IFC element types is a significant and ongoing area of research within the broader field of model quality improvement. Research in this area can be grouped into three main directions: The first includes rule-based and heuristic approaches, where classifications are defined through explicit geometric, topological, and relational rules [9, 10, 11], or statistical anomaly detections [12]. The second direction involves neural network models trained on element shapes or images[6, 13, 14, 15]. The third, the most recent direction, focuses on training models with multi-modal data utilizing non-geometric evidence, such as spatial and topological relations, property-set semantics, and metadata. Recent studies train graph neural networks (GNNs) to combine geometry with spatial and topological relationships[8, 16], or ensembles of multiple classification pipelines[5], achieving better performance on complex IFC models.

The earliest rule-based, non-NN approaches, such as those done by [9, 12], offered high transparency and interoperability. They relied on predefined logic, whether via geometric outlier detection or explicit mapping rules, making it clear in their classification reasoning. However, these methods showed limited scalability in the real world and complex datasets. In practice, their reliance on fixed assumptions often failed to generalize across diverse IFC BIM datasets.

Geometry-only NN-based approaches, including [7, 13, 14, 15, 17, 18], tested by IFCNet[6], moved toward data-driven classification by learning directly from shapes, either via rendered images or raw 3D point clouds. These models improved adaptability over previous non-NN-based approaches and could capture subtle geometric distinctions that might be overlooked when defining manual rules. Nevertheless, they inherit a similar limitation, that is, semantically different elements sharing similar geometric properties, and vice versa. The model often cannot disambiguate without non-geometric context, making their application limited and often still requiring manual supervision in real-world applications.

Context-aware GNN approaches, as shown by [8], address this ambiguity by embedding elements in a graph that captures not only geometry but also relational context. These approaches demonstrated that incorporating connectivity significantly improves classification accuracy on real-world datasets.

Multi-modal and ensemble methods explicitly fuse heterogeneous evidence. [5] stack four channels (semantic search, rules, geometric ML, MVCNN) with a meta-learner that claims to outperform all other tools, reports an accuracy of the ensemble classifier of 91%. Meanwhile, MMDL [16] fuse graphical and non-graphical (Pset-level) features with feature selection for high-accuracy fine-grained labelling. These studies collectively indicate that robust type correction is best treated as multi-evidence fusion.

Excluding the earlier rule-based approaches, all related research requires high-quality training data in the form of correctly labelled IFC elements, to construct machine learning models for labelling unseen elements. This knowledge requirement is unrealistic given the noise

present in most IFC specifications, and this shortcoming is the fundamental motivation of our work.

3 Building Element Relabelling

IFC documents, while rich, often contain mistyped elements, overly generic default types, or inconsistent subtype usage across authoring tools, due to IFC's lack of strict schema enforcement [19]. Such issues can lead to incorrect task assignments, missing elements for tasks, or unreliable Quantity Takeoff (QTO) when generating schedules [20]. Furthermore, many IFC subtype distinctions are too fine-grained to correspond meaningfully to construction tasks.

To address these challenges, this paper frames the construction-oriented relabelling task as a *definition retrieval* problem. Given an IFC model, each one of its elements is treated as a query to be matched against a corpus of IFC type definitions, using two similarity metrics. The lexical-based *Term Frequency-Inverse Document Frequency (TF-IDF)* metric computes the frequency of words shared by the query and the definition, while under-weighing words frequently occurring across the definitions corpus. The *Embedding-based Semantic* metric maps the query and the definition on a semantic embeddings model and computes the similarity of the resulting vectors.

The overall relabelling process follows the workflow of Figure 1. First, all IFC elements are extracted from an IFC file. For each element, an element-text is extracted from the IFC element's name, description, and property sets. The original element IFC type assignments are removed from the input representation and do not influence the relabelling process. The element-text is compared against all IFC type definitions, using either the two metrics above, resulting in a similarity score for each IFC type, i.e. a type confidence score. This is then mapped to the IFC type's associated *BIM Core* label, resulting in both a new IFC element type and *BIM Core* label for each element.

Table 1. Example of the IFC-type definition database used as the search target for TF-IDF and embedding-based similarity matching.

IFC Type	Definition
`IfcWall`	The wall represents a vertical construction that may bound or subdivide spaces. Walls are usually vertical, or nearly vertical, planar elements, often designed to bear structural loads.
`IfcDoor`	The door is a built element that is predominantly used to provide controlled access for people, goods, animals and vehicles. It includes constructions with hinged, pivoted, sliding, and additionally revolving and folding operations.
...	

Table 2. Selected IFC Type to *BIM Core* label mapping, grouped by label category.

BIM Core Label	IFC Types Mapped
Structural Elements	
Slab	`IfcSlab`
Wall	`IfcWall`, `IfcWallStandardCase`, `IfcCurtainWall`
StructuralElement	`IfcBeam`, `IfcColumn`, `IfcMember`
Footing	`IfcFooting`
Roof	`IfcRoof`
Plate	`IfcPlate`
Envelope and Access	
Opening	`IfcOpeningElement`
Window	`IfcWindow`, `IfcWindowStyle`
Door	`IfcDoor`, `IfcDoorStyle`
Railing	`IfcRailing`
Stair	`IfcStair`, `IfcStairFlight`
Ramp	`IfcRamp`, `IfcRampFlight`
Covering	`IfcCovering`
Building Services and Equipment	
Pipe	`IfcFlowTerminal`, `IfcFlowFitting`, `IfcFlowSegment`, `IfcPipeSegment`, `IfcFlowTreatmentDevice`, `IfcFlowStorageDevice`, `IfcFlowController`, `IfcFlowMovingDevice`, `IfcDistributionFlowElement`
EnergyConversionDevice	`IfcSolarDevice`, `IfcEnergyConversionDevice`
TransportElement	`IfcTransportElement`
WashroomAccessory	`IfcSanitaryTerminal`, `IfcWasteTerminal`
Other	
Other	*<everything else>*

3.1 IFC Types Dictionary and *BIM Core* Labels

The proposed relabelling workflow relies on a dictionary of IFC element definitions as the target for each element in the input IFC model. This dictionary can be constructed from the official IFC schema documentation [1], which provides official textual definitions for IFC types. Examples of two definitions can be seen in Table 1.

This documentation organizes the IFC type definitions in a hierarchy of building concepts and physical elements. Using this hierarchy as the basis, this paper defines *BIM Core* labels as a subset of element-level IFC types, organized in a set of construction-oriented categories. The *BIM Core* types are at the level of granularity so as to correspond to construction activities, thus supporting automated schedule construction. The definition dictionary stores the official definition for each IFC type that participates in the *BIM Core* mapping, while the mapping in Table 2 records which IFC types belong to each *BIM Core* label.

3.2 Element-Text Extraction

In order to formulate an IFC element into a query for retrieving the most appropriate IFC definition for this element, the following element aspects are extracted: (i) the *name* of the IFC element as a string; (ii) its *description* as a string; and (iii) its `IfcPropertySet`, as a container of property-value pairs of strings.

The extracted strings are concatenated to form the query, after being pre-processed as follows. First, all

[1] `https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/`. Accessed: 2025-11-17

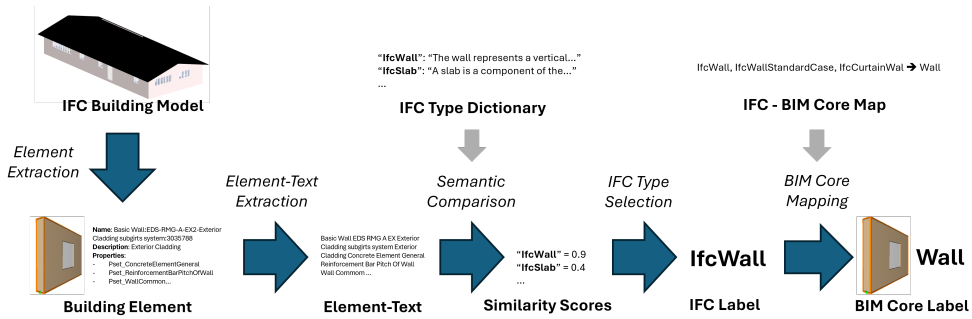

Figure 1. Relabelling Overview

non-alphabetic characters are converted into whitespace to eliminate punctuation, numeric codes, and formatting characters. To ensure consistent behaviour when tokenizing, all leading and trailing white spaces are removed, and repeated white spaces are consolidated into a single white space. Next, any string in camelCase or PascalCase is split into separate words. For example, the words firstSecond and first_Second would both be split into "first" and "Second".

The resulting text is tokenized using the spaCy[2] model `en_core_web_sm`, which is a lightweight English tokenizer that handles common lexical patterns and boundary cases encountered in technical text. Finally, tokens shorter than three characters are removed to filter out fragments and meaningless abbreviations. Table 3 shows an example of an IFC element's textual metadata before and after the tokenization process.

3.3 Term-based Similarity (TF-IDF)

The TF-IDF similarity metric measures the similarity between the input query and each IFC type definition by comparing their weighted word-frequency vectors, using cosine similarity.

First, each document is transformed into a vector of numbers representing the TF-IDF score for every unique term in the corpus by multiplying the frequency of the term within the document (TF) by its rarity (inverse frequency) across the corpus (IDF). A high score indicates a rare, relevant word, while low scores suggest common, less informative words.

The method follows a standard vector-space retrieval pipeline: both element-text and IFC type definitions are converted into TF-IDF vectors, and their similarity is computed using cosine similarity.

TF-IDF vectorization is applied consistently to two text sources: (i) the element-text and (ii) the IFC type definition texts stored in the definition dictionary. Each IFC type definition is treated as a document and transformed into a sparse TF-IDF vector using the same fitted vectorizer. The element-text is transformed using the same TF-IDF model so that element-text and IFC type definition vectors lie in the same vocabulary space.

For comparability with the embedding-based scores, the TF-IDF similarities are min–max normalized at the IFC type level for each element-text:

$$\hat{s}_{\text{tfidf}}(t) = \frac{s_{\text{tfidf}}(t) - s_{\min}}{s_{\max} - s_{\min}}, \tag{1}$$

where $s_{\min}$ and $s_{\max}$ are the minimum and maximum TF-IDF similarities observed across all IFC types for the element-text.

Since each *BIM Core* label aggregates multiple IFC types, a label score is obtained by max-pooling the normalized type scores over the IFC types mapped to that label. Specifically, for a label ℓ with mapped IFC types $T(\ell)$:

$$\hat{s}_{\text{tfidf}}(\ell) = \max_{t \in T(\ell)} \hat{s}_{\text{tfidf}}(t). \tag{2}$$

The predicted *BIM Core* label is the label with the highest $\hat{s}_{\text{tfidf}}(\ell)$.

3.4 Semantic Similarity (SemEmb)

The second similarity metric explored in the relabelling pipeline relies on a pre-trained Sentence-Transformers model [21] based on MiniLM [22], i.e., `all-MiniLM-L6-v2`[3].

Semantic mapping encodes the same two text sources as the TF-IDF module, namely the element-text and IFC type definition text from the definition dictionary. Both are encoded into two 384-dimensional embedding vectors.

To reduce ambiguity in element-text construction, the element metadata fields are formatted using lightweight prefixes (e.g., `Name:`, `Properties:`, `Description:`) before encoding. The name field is always included, while optional fields (description, property set names) are included only when present and enabled.

Let $\mathbf{e}_q$ denote the embedding of an element-text and let $\mathbf{e}_t$ denote the embedding of the definition text associated with IFC type t. All embeddings are L2-normalized, so cosine similarity reduces to the dot product:

$$s_{\text{sem}}(t) = \mathbf{e}_q \cdot \mathbf{e}_t. \tag{3}$$

[2] `https://spacy.io/api/tokenizer`. Accessed: 2025-11-17

[3] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`. Accessed: 2025-11-17

Table 3. Example IFC element fields before and after tokenization.

Field	Before tokenization	After tokenization
Name	`Compound Ceiling:EDS-RMG-ZZ-A-GWB Ceiling:4257884`	`Compound Ceiling EDS RMG GWB Ceiling`
Description	`Ceiling 4257884`	`Ceiling`
Property Set	`Pset_CoveringCommon=>Reference, Pset_CoveringCommon=>Finish`	`Covering Common Reference, Covering Common Finish`

Table 4. Summary of the IFC buildings used in the dataset.

Building	Source	Elems	Building Description
A	Open IFC Model Repository	779	Three-level gym / wellness facility
B	Open IFC Model Repository	574	Low-rise mixed-use building with four storeys and one basement level
C	Nordic Sustainable Construction	12,744	As-built five-storey concrete office building with one basement level
D	Kaggle IFC samples	1,775	Six-storey concrete multi-family residential apartment building with one basement level.
E	Kaggle IFC samples	126	Single-storey concrete residential house
F	University of Alberta	3,294	University of Alberta's First Peoples' House
G	Open IFC Model Repository	299	Two-storey detached single-family house with a pitched roof
H	Kaggle IFC samples	122	Small low-rise building complex with two blocks, one two-storey and one single-storey
I	Open IFC Model Repository	432	Two-storey spa/hotel building with an outdoor pool area
J	Kaggle IFC samples	58	Four-storey low-rise basic concrete building
K	Open IFC Model Repository	1,453	Three-storey apartment building with one basement level
L	Open IFC Model Repository	548	Villa-style residential site with outdoor amenity areas

Finally, to enable hybrid ranking with TF-IDF, semantic similarities are min–max normalized at the IFC type level for each element-text, and the semantic label score is obtained by max-pooling the normalized type scores, with formulas similar to the ones discussed in the previous section.

4 Experimental Setup

We have evaluated the effectiveness of the two proposed IFC relabelling methods using a collection of real-world IFC building models, representing a broad range of real-world buildings and element specification data. The elements of these models were relabelled using both similarity metrics and three element-text extraction settings: Name only (N), Name+Description (ND), or Name+Description+PropertySet (NDP).

4.1 Building and Element Dataset

The dataset used in this study serves as the ground truth reference for testing and evaluating IFC type classification methods. The dataset has been carefully curated to ensure diversity in building model sources, building typologies, details, and element distributions. In total, 12 IFC building models were selected.

The building models included in the dataset were selected to support the construction of a reliable ground-truth reference, and to reduce cases where a large fraction of elements are exported under generic placeholders such as `IfcBuildingElementProxy`.

Each building element in the model contained an initial IFC type. To establish ground truth labels, each element was manually inspected to ensure that the IFC type accurately represented the type of the element and modified if needed. Using the IFC type to *BIM Core* mapping, a ground truth *BIM Core* label was assigned to each element. Therefore, the evaluation of the relabelling is the accuracy of re-learning each element's verified initial ground-truth IFC type and *BIM Core* label.

The complete element dataset consists of 20,097 elements. Table 4 details each building model's source, type, and element count. The largest category is `StructuralElement`, with 12,193 instances, reflecting the structural focus of the building models. `Walls` form the second largest group (2,631), followed by `Other`, which contains furniture and appliances not directly relevant to construction scheduling. Finishing-related categories such as `Covering`, `Plate`, `Window`, and `Door` are also well represented, ensuring coverage of both structural and architectural domains. Service-related categories such as `WashroomAccessory`, `EnergyConversionDevice`, and `Pipe` provide representation of mechanical and sanitary systems.

Overall, the dataset emphasizes structural and envelope tasks, a progression into interior and service installations, and a smaller presence of specialized elements.

4.2 Evaluation Metrics

To quantitatively assess the performance of the two relabelling metrics, we computed (i) the overall classification accuracy, i.e., how well the two similarity metrics assign the correct IFC type to each element and (ii) recall for the `Other` category, i.e., how well they recognize elements that do not belong to any construction-relevant type, correspondingly.

The **Overall Accuracy** metric measures the proportion of IFC elements whose predicted *BIM Core* label matches the ground-truth label. It is defined as:

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \tag{4}$$

where N_{correct} is the number of correctly classified elements and N_{total} is the total number of evaluated elements.

The **Recall** for the `Other` category is defined as:

$$\text{Recall}_{\text{Other}} = \frac{TP_{\text{Other}}}{TP_{\text{Other}} + FN_{\text{Other}}} \tag{5}$$

where TP_{Other} is the number of `Other` elements correctly predicted as `Other`, and FN_{Other} is the number of ground-

Table 5. Overall accuracy pooled across all IFC elements from the 12 evaluated buildings.

Text used	TF-IDF	SemEmb
Name	29.5%	30.1%
Name+Description	29.5%	30.2%
Name+Description+PropertySet	**88.1%**	32.5%

Table 6. Per-category classification accuracy for each *BIM Core* label for Name+Description+PropertySet. The best value(s) in each row are in bold.

BIM Core label	Elems	TF-IDF	SemEmb
StructuralElement	12193	**95.9%**	3.3%
Wall	2631	**90.1%**	88.7%
Other	2231	73.7%	**80.6%**
Covering	980	**26.9%**	19.7%
Plate	860	**91.1%**	12.7%
Window	788	99.5%	**99.6%**
Door	732	**99.2%**	61.8%
Slab	665	**94.4%**	84.8%
WashroomAccessory	405	**59.8%**	54.8%
EnergyConversionDevice	203	0.0%	0.0%
Pipe	183	**81.4%**	57.4%
Railing	141	**100.0%**	**100.0%**
Stair	93	**100.0%**	**100.0%**
Footing	60	0.0%	0.0%
Ramp	12	**100.0%**	66.7%
TransportElement	4	**75.0%**	0.0%

truth `Other` elements incorrectly assigned to a non-`Other` category.

5 Results and Discussion

Table 5 reports the classification accuracy pooled over all the IFC elements across all 12 buildings. Approaches that rely only on element names achieve low accuracy (29.5% for TF-IDF; 30.1% for semantic search), and adding descriptions yields negligible change (29.5% and 30.2%, respectively). Incorporating property-set text produces a substantial improvement as TF-IDF with Name+Description+PropertySet sets reaches 88.1%. Semantic search with Name+Description+PropertySet remains markedly lower at 32.5%. The significantly poor performance of the SemEmb metric could potentially be improved with a domain-specific language model, which could be obtained by retraining the all-MiniLM-L6-v2 with IFC documents.

5.1 Per-category Effectiveness

To examine label-level behaviour beyond the pooled scores, Table 6 reports per-category accuracy for each *BIM Core* type. Comparing the two similarity methods, it is evident that TF-IDF is either equal to or outperforms SemEmb in nearly all categories except for `Other` and `Window`. For both methods, some labels remain challenging: `Covering` has a low accuracy even under the best setting (26.9%), `WashroomAccessory` achieves 59.8%, and `EnergyConversionDevice` and `Footing` remain at 0% accuracy across all settings. Results for rare categories such as `Ramp` and `TransportElement` should be interpreted cautiously due to small sample sizes.

Several category-specific failures can be traced to similarly weak or ambiguous textual cues. For `EnergyConversionDevice`, element names are frequently generic (often containing terms such as "panel"), while distinctive property sets are not consistently present; this leads to systematic confusion with `Plate`. For `Footing`, names are often non-descriptive identifiers (e.g., coded tags), and the available text does not provide stable cues for differentiating this class from other structural elements, resulting in persistent misclassification.

Finally, `Covering` remains difficult because it is broad and heterogeneous, spanning elements such as insulation, paint, and finishing components. This semantic breadth reduces the likelihood that a small set of tokens or property-set terms will uniquely characterize the category, and misclassifications commonly drift toward nearby or related labels (e.g., `Wall` or `Slab`). Overall, these error patterns indicate that the dominant failure mode is not the matching mechanism itself, but the sparsity, inconsistency, and low specificity of the textual metadata available for many elements and buildings. Fortunately, these types do not have major implications for construction schedules, and therefore, the impact of their misclassification would not be too costly.

Table 7. Description availability statistics and Accuracy for the 12 evaluated IFC buildings (A–L). $\#D$ denotes the number of elements that contain a description and $\mu_t(Desc.)$ denotes the mean token count of the IFC *Description* field after basic whitespace splitting.

Building	Elems	$\#D$	$\mu_t(D)$	TF-IDF	SemEmb
A	779	0	0	**97.2%**	27.2%
B	574	307	1.70	**98.6%**	**98.6%**
C	12744	0	0	**97.8%**	13.7%
D	1775	0	0	85.4%	**88.0%**
E	126	0	0	68.3%	**90.5%**
F	3294	0	0	**76.9%**	43.4%
G	299	299	1.00	**11.0%**	0.0%
H	122	0	0	**98.4%**	93.4%
I	432	160	0.86	**98.8%**	**98.8%**
J	58	0	0	**98.3%**	93.1%
K	1453	106	1.17	30.3%	**31.4%**
L	548	61	0.32	**97.1%**	96.2%

5.2 Per-building effectiveness

Table 7 shows the labelling accuracies for each of the 12 buildings in our dataset. Overall, for the TF-IDF with property sets, per-building accuracy is equal to or better than SemEmb for nine of the twelve buildings. In particular, TF-IDF's accuracy is 33.5% or greater than SemEmb for buildings A, C, and F, while SemEmb sees only a significant accuracy improvement (22.2%) for building E. All other buildings show similar accuracies (difference <10%) between the two methods.

Building G (11.0%) and Building K (31.4%) are the two that show very poor accuracy for both methods. Upon further inspection, we found that these buildings had element names dominated by coded identifiers, abbreviations, or project-specific shorthand. Such tokens are difficult to match reliably using TF-IDF, and also provide limited semantic grounding for embedding-based similarity, which reduces the effectiveness of both scoring strategies.

5.3 Comparison with Previous Work

This work examines whether a lightweight, keyword-driven relabelling pipeline can achieve competitive performance relative to recent learning-based approaches such as [5, 7, 8, 16], while requiring no training data and limited computational resources. When interpreting such comparisons, it is essential to distinguish between what is numerically comparable and what is comparable only at the level of problem setting. Prior studies differ in (i) the target taxonomy (fine-grained IFC entities vs. construction-oriented categories), (ii) the input information they require (e.g., geometry, neighbourhood context, property values, or element metadata), and (iii) evaluation datasets and curation assumptions. Consequently, reported accuracies are not directly comparable with our results unless the same dataset, label space, and input constraints are used.

The ideal comparison is therefore under the same evaluation procedure. To this end, we implemented the Graph Neural Network (GNN) classifier of [8] and evaluated it on our 12-building dataset using the same *BIM Core* ground truth and the same constraint that original IFC type assignments are removed from the input representation. Under this setting, the reproduced implementation achieved an overall accuracy of 50.3%. An enhanced variant that augments each IFC element with a semantic embedding of its property-set names (encoded using the same `all-MiniLM-L6-v2` model used in SemEmb increased overall accuracy to 74.5%. These results indicate that dataset composition and available metadata strongly affect attainable performance for learning-based baselines, and that property-set text provides substantial discriminative signal in our setting. Under the same dataset and label space, our best similarity-based configuration (Table 5) attains 88.1% overall accuracy without any model training.

For a broader context, several recent studies report high accuracies under different task definitions and input assumptions. IFCNet [6] benchmarks geometry-centric baselines (including MVCNN [7]) for fine-grained IFC entity classification and reports 85.5% overall accuracy on its benchmark. [8] report 87.6% accuracy for a context-aware GNN on their dataset and label space. While both headline numbers are slightly higher than the best result reported in this work (88.1%), they are not directly comparable because they target raw IFC schema entities and rely on geometry and/or neighbourhood context rather than a text-driven relabelling into construction-oriented *BIM Core* categories. [16] report 98.4% accuracy for a multi-modal deep learning model, but this performance is achieved under a different label space and with learned representations and additional modalities beyond the text-only constraints considered here. [5] report 91.0% accuracy for an ensemble classifier in model-performance workflows; importantly, their element text representation includes the element's IFC type as an explicit attribute, whereas this work intentionally excludes original IFC type assignments from the input due to their unreliability in practice.

Finally, an interesting comparison would be to consider the motivation of this study: training and computational requirements. In contrast to deep learning pipelines that require training and dataset-specific supervision, the proposed relabelling methods are training-free and operate via retrieval-style similarity against curated IFC definition text. The reproduction study above further suggests that, under our dataset and input constraints, this lightweight design can remain competitive with a recent learning-based baseline even without access to original IFC type assignments. This work did not re-implement the full MMDL and MVCNN pipelines or [5]; these methods are therefore discussed here based on their reported problem settings and input assumptions rather than through like-for-like numerical replication.

5.4 Downstream Impact to Scheduling

This work is motivated by the bigger problem of automated construction schedule generation. We have developed a tool for automatically generating the work-breakdown structure of a construction project based on the IFC model of the to-be-constructed building. The tool identifies the necessary tasks to be completed (based on the *BIM Core* types of the building's IFC elements), sequencing these tasks according to their inter-dependencies (based on a set of construction templates and the relationships of the building's IFC elements), and estimating their durations (based on general quantity-take-off rules applied to the properties of the IFC elements). We are currently conducting a study to evaluate this tool, in collaboration with the project management team of a building undergoing renovations on the UofAlberta campus. Our initial findings show that our method can create a schedule with similar tasks to the actual schedule and relatively good estimates of the durations of these tasks.

6 Conclusion and Future Directions

This paper describes a construction-oriented IFC relabelling approach to correct mislabelled IFC element types, and to assign each element with a label from a compact set of *BIM Core* labels that better align with the needs of construction planning. The relabeller employs two similarity methods, namely TF-IDF and Embedding-based Semantic (SembEmb) similarity, which were evaluated on 12 building models with over 20,000 elements. The TF-IDF relabelling method achieved a 88.1% total accuracy, demonstrating that semantic signals alone can provide competitive performance while avoiding the cost of training-heavy models.

Unlike machine-learning-based relabellers that assume the availability of geometry and topology information and require model training, the approach developed in the work

uses only semantic metadata typically available in IFC documents, including element names, descriptions, and property sets, and traditional similarity metrics. However, semantic-only methods remain sensitive to label quality: when element names and property sets are sparse or noisy, keyword-driven classifiers degrade. A robust extension would be to add an automatic fallback to geometry, topology, or rules-based classification when semantic confidence is low, enabling the pipeline to remain usable on low-quality IFC exports.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council (NSERC) Canada "Federated Platform for Construction Simulation" Alliance Grant. The authors also wish to thank Maria Al-Hussein, Stephen Hague, Dr. Yasser Mohamed and Dr. Simaan AbouRizk for many insightful conversations that guided this work.

References

[1] A. K. Singh, A. Pal, P. Kumar, J. J. Lin, and S.-H. Hsieh. Prospects of integrating bim and nlp for automatic construction schedule management. In *Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC)*, volume 40, pages 238–245, 2023. doi:10.22260/ISARC2023/0153.

[2] X. Zhao, K.-W. Yeoh, and D. K. H. Chua. Extracting construction knowledge from project schedules using natural language processing. In *The 10th international conference on engineering, project, and production management*, pages 197–211, 2020. doi:10.1007/978-981-15-1910-9_17.

[3] R. Sacks, C. Eastman, G. Lee, and P. Teicholz. *BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers*. 3rd edition, 2018. doi:10.1002/9781119287568.

[4] International Organization for Standardization. *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema*. 2024.

[5] D. Utkucu, H. Ying, Z. Wang, and R. Sacks. Classification of architectural and mep bim objects for building performance evaluation. *Advanced Engineering Informatics*, 61: 102503, 2024. doi:10.1016/j.aei.2024.102503.

[6] C. Emunds, N. Pauen, V. Richter, J. Frisch, and C. Van Treeck. Ifcnet: A benchmark dataset for ifc entity classification. In *EG-ICE 2021 Workshop on Intelligent Computing in Engineering*, volume 166, 2021. doi:10.48550/arXiv.2106.09712.

[7] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. doi:10.48550/arXiv.1505.00880.

[8] G. Austern, T. Bloch, and Y. Abulafia. Incorporating context into bim-derived data—leveraging graph neural networks for building element classification. *Buildings*, 14 (2):527, 2024. doi:10.3390/buildings14020527.

[9] T. Krijnen and M. Tamke. Assessing implicit knowledge in bim models with machine learning. In *Modelling Behaviour: Design Modelling Symposium 2015*, pages 397–406, 2015. doi:10.1007/978-3-319-24208-8_33.

[10] R. Sacks, L. Ma, R. Yosef, A. Borrmann, S. Daum, and U. Kattel. Semantic enrichment for building information modeling: Procedure for compiling inference rules and operators for complex geometry. *Journal of Computing in Civil Engineering*, 31(6):04017062 1–12, 2017. doi:10.1061/(ASCE)CP.1943-5487.0000705.

[11] L. Ma, R. Sacks, U. Kattel, and T. Bloch. 3d object classification using geometric features and pairwise relationships. *Computer-Aided Civil and Infrastructure Engineering*, 33 (2):152–164, 2018. doi:10.1111/mice.12336.

[12] B. Koo and B. Shin. Applying novelty detection to identify model element to ifc class misclassifications on architectural and infrastructure building information models. *Journal of Computational Design and Engineering*, 5(4): 391–400, 2018. doi:10.1016/j.jcde.2018.03.002.

[13] J. Kim, J. Song, and J.-K. Lee. Recognizing and classifying unknown object in bim using 2d cnn. In *International Conference on Computer-Aided Architectural Design Futures*, pages 47–57, 2019. doi:10.1007/978-981-13-8410-3_4.

[14] M. Leonhardt, N. Pauen, L. Kirnats, J.-N. Joost, J. Frisch, et al. Implementierung von ki-basierten referenzprozessen für die computergestützte objekterkennung im gebäude. In *BauSim Conference 2020*, volume 8, pages 599–606, 2020. doi:10.3217/978-3-85125-786-1-72.

[15] B. Koo, R. Jung, Y. Yu, and I. Kim. A geometric deep learning approach for checking element-to-entity mappings in infrastructure building information models. *Journal of Computational Design and Engineering*, 8(1):239–250, 2021. doi:10.1093/jcde/qwaa075.

[16] H. Liu, V. J. L. Gan, J. C. P. Cheng, and S. A. Zhou. Automatic fine-grained bim element classification using multimodal deep learning (mmdl). *Advanced Engineering Informatics*, 61:102458, 2024. doi:10.1016/j.aei.2024.102458.

[17] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5): 1–12, 2019. doi:10.1145/3326362.

[18] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao. Meshnet: Mesh neural network for 3d shape representation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8279–8286, 2019. doi:10.1609/aaai.v33i01.33018279.

[19] Y. Yu, S. Kim, H. Jeon, and B. Koo. A systematic review of the trends and advances in ifc schema extensions for bim interoperability. *Applied Sciences*, 13(23):12560, 2023. doi:10.3390/app132312560.

[20] S. Alathamneh, W. Collins, and S. Azhar. Bim-based quantity takeoff: Current state and future opportunities. *Automation in Construction*, 165:105549, 2024. doi:10.1016/j.autcon.2024.105549.

[21] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019. doi:10.48550/arXiv.1908.10084.

[22] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020. doi:10.48550/arXiv.2002.10957.