2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

FILLING IN MISSING MATERIAL LAYER INFORMATION FOR GENERATING WALL AND SLAB OBJECT LIBRARIES FROM DRAWING

Sungkyu Lee[1], Miyoung Uhm[1], and Ghang Lee[1,2]
[1]Yonsei University, Korea, Republic of South Korea
[2]Technical University of Munich, Germany

Abstract

Automated generation of building information modeling objects from computer-aided design drawings require comprehensive information about material layers, including their functional roles—information typically absent from drawings. This study proposes a hybrid two-step methodology that combines a large language model and a machine learning algorithm to address it, focusing specifically on wall and slab objects. First, the method identifies the function of each material layer as a multi-class classification task, utilizing few-shot prompting with GPT-4o, achieving 99.8% macro F1-score. Second, it classifies whether material layer information pertains to walls or slabs using random forest model with FastText embedding, achieving 89.5% macro F1-score.

Introduction

As the effectiveness of building information modeling (BIM) becomes increasingly evident, the adoption of BIM in the construction industry is accelerating rapidly. However, the industry still predominantly follows a BIM conversion design approach, wherein detailed design is first carried out using computer-aided design (CAD), and then a BIM model is authored for delivery based on the CAD drawings (Lee & Lee, 2023). This modeling process is time and cost intensive, often leading to errors such as missing information and model inaccuracies (Rho et al., 2020). To address time and cost issues, academia has conducted research on the automatic conversion of geometric objects, such as walls, columns, and slabs, from 2D floor plans into 3D models (Zeng et al., 2019; Liao et al., 2021). However, existing studies have limitations in that they overlook the rich semantic information embedded in section views, such as material layers. Material layer information in drawings is critical as it determines the intended design, structural integrity, and environmental performance of a building, directly influencing its overall quality. Figure 1 illustrates an example of wall types represented with material layers in drawings. As material layer information is one of the fundamental components of a BIM model, its information should be considered in the CAD-to-BIM domain.

However, material layer information extracted from annotations in drawings is difficult to directly apply to BIM models. This problem can be addressed by generating BIM objects that contain material layers. Because CAD drawings typically include only material names and thickness data. In contrast, major BIM authoring tools such as Revit, ArchiCAD, and Bentley's OpenBuildings Designer require material function information for each material (e.g., Revit: structure, substrate, membrane, thermal, finish), necessitating the inference of additional material properties (Bentley Systems, 2023; Graphisoft, 2024; Revit API Docs, 2024). Furthermore, the extracted material layer information does not explicitly indicate whether it pertains to a wall or a slab, posing another challenge. To address these research issues, This study aims to leverage natural language processing (NLP) techniques to fill in the missing information, enabling the application of material layer information from drawings to BIM authoring tools.

This paper is structured into seven sections, including the Introduction. The second section reviews studies that have applied NLP in the architectural domain to address the research challenges outlined earlier. The third section introduces the research methodology derived from the literature review. The fourth section explains the experimental design used to validate the proposed methodology, while the fifth section presents the experimental results obtained from detailed design CAD drawings. The Discussion section interprets the results, followed by the Conclusion section, which summarizes the contributions and limitations of this study.

Literature Review

CAD-to-BIM Conversion

This section reviews existing studies related to the research topic and approach, with a particular focus on construction related research that addresses both geometric and semantic information in the conversion of CAD drawings into BIM models. The following studies illustrate cases where vector-based methods have been employed to extract the geometric and semantic data necessary for BIM generation from CAD drawings.

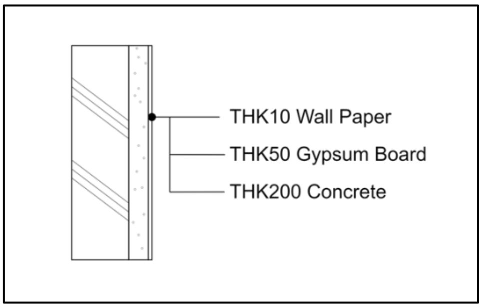

Figure 1: Example of material layer information on a drawing

Byun & Sohn (2020) identified the shape and location of structural objects in floor plans based on nearby text annotations, classified their types, and extracted size and reinforcement information from member lists using code symbols. Similarly, Yang et al. (2020) utilized CAD layers to recognize the shape and position of structural objects while extracting codes and slab thicknesses from text annotations. These extracted codes were then used to identify object dimensions and material information from member lists. Unlike other studies, Yin et al. (2024) leveraged elevation drawings by identifying CAD layers corresponding to walls, openings, heights, and grid bubbles and matching elevation views with architectural floor plans. Subsequently, they recognized the shape and location of walls and openings in floor plans and extracted opening height information from elevation views, enabling the accurate reconstruction of BIM models. The details of these studies are summarized in Table 1.

Despite these efforts, significant research gaps remain. Existing studies have primarily focused on extracting information from 2D floor plans and providing detailed object attributes by relying on member lists. Although Yin et al. (2024) incorporated elevation views, their focus was predominantly on geometric aspects such as height. Furthermore, all studies have concentrated on handling geometric information, such as dimensions and heights, by utilizing text annotations. While Byun & Sohn (2020) addressed detailed attributes such as reinforcement, there has been a lack of research on material layer information, which is an essential attribute for defining semantically rich architectural objects. Therefore, our approach proposes a methodology that utilizes NLP techniques to integrate material layer information from sectional views into BIM models. The objective is to fill in the missing information of material layer data represented in sectional views to generate compound objects consisting of multiple materials.

Multi-class Classification of Material Layer Function

Multi-class classification, used to categorize material layers according to the various functional classes required by BIM authoring tools, corresponds to named entity recognition (NER) in NLP. Traditional NER techniques perform tagging based on predefined datasets. However, creating a dataset that encompasses all architectural materials is a labor-intensive process, and even once established, it requires continuous updates. Moreover, building materials exhibit context dependent functional variations. For example, epoxy coating is generally used as a membrane material but can also function as a finish. Therefore, rather than constructing a large-scale dataset for building materials, it is necessary to perform NER based on the contextual information of material layers.

The following studies have performed NER in the construction industry by leveraging contextual information rather than relying on large-scale datasets. Moon et al. (2020) employed a bi-LSTM-based NER model trained on 1,650 sentence samples to extract bridge components, damage types, and causes from bridge inspection reports. Jeon et al. (2022) developed a thesaurus of 1,097 terms from 69,750 defect complaint records and applied transfer learning to a pre-trained language model to filter defect information from noisy complaint data.

Table 1: Studies for CAD-to-BIM

Paper	Byun and Sohn. (2020)	Yang et al. (2020)	Yin et al. (2020)
Floor plan	●	●	●
Member list	●	●	
Elevation			●
Section view			
Real image			
Code	●	●	
Size	●	●	
Height			●
Grid bubble			●
Rebar	●		
Material		●	

Moon et al. (2022) trained a bi-LSTM-based NER model on 56 construction specifications to extract potential conflict factors in construction project documentation. Baek et al. (2023) combined a RoBERTa model with FastText to perform NER on a decade's worth of Middle Eastern economic weekly reports to identify companies operating in the region's construction market. These studies demonstrated that by leveraging contextual information from training datasets, NER could be effectively performed even with limited data. However, material layer information has inherently sparse contextual information and lacks sufficient relational data for effective training.

As a result, the methods used in previous studies cannot be directly applied. To address this issue, our approach proposes using pre-trained large language models (LLMs) for NER in material layer classification.

Nevertheless, LLMs often exhibit suboptimal performance in domain specific NLP tasks. To overcome this limitation, techniques such as fine-tuning and prompt engineering can be employed. Baek et al. (2023) fine-tuned GPT-4 to extract social conflict factors in Korea's civil engineering projects via NER. Baek et al. (2023) used web scraped Korean news articles, annotated key phrases, and incorporated them into the fine-tuning dataset, achieving an F1-score of 84.7% for NER performance. However, while Baek et al. (2023) demonstrated the application of LLMs in the construction domain, it did not specifically target construction material information. Therefore, our approach proposes a prompt engineering-based multi-class classification approach like NER using LLMs to achieve high performance in extracting material information from CAD drawings within the construction domain.

Binary Classification of Building Object Type

Even if the function of materials has been inferred through NER algorithm, the material layer information is merely a textual aggregation of material names, functions, and thicknesses extracted from CAD. Therefore, a binary classification task is required to determine whether this information corresponds to a wall or a slab.

The following are previous studies on classifying building objects in architectural drawings. Byun & Sohn (2020) determined object types by utilizing object shapes, topological relationships, and symbols to convert geometric information in the CAD-to-BIM domain. First, the smallest rectangle in the drawing was classified as a column, rectangles connecting columns were categorized as walls or beams, and rectangles enclosed by beams were identified as slabs. Subsequently, nearby symbols were further examined to accurately classify object types. Yang et al. (2020) classified object types by simplifying the geometric representations of line-based objects in CAD drawings to convert them into BIM. Yang et al. (2020), closed box-shaped forms were classified as columns, while two parallel lines were categorized as beams. Wu et al. (2022) reviewed errors in industrial foundation classes (IFC) files by extracting information such as object shape, absolute and relative positions, and materials from IFC, leveraging machine learning algorithms to classify object types. The study employed Decision Tree, Bayesian Network, Support Vector Machine, Random Forest, and Neural Network algorithms, demonstrating that the Random Forest algorithm achieved the highest performance. All three studies effectively classified building objects by utilizing relevant object-related information. However, Byun & Sohn (2020) and Yang et al. (2020) relied solely on geometric data, such as object shapes and topological relationships, making them unsuitable for text-based data such as material layer information. In contrast, Wu et al. (2022) utilized both geometric and semantic information to examine IFC object types, making it highly relevant to the present study, particularly in its use of material data. Nevertheless, a key distinction of our approach lies in the fact that it performs binary classification using only material names and functions, without incorporating various attributes such as object shape and position. Therefore, our approach aims to propose the most effective binary classification method for distinguishing between walls and slabs based solely on material names and functions.

Proposed Method

The research methodology consists of two stages, as illustrated in Figure 2: 'Multi-class Classification of Material Layer Function' and 'Binary Classification of Building Object Type.'

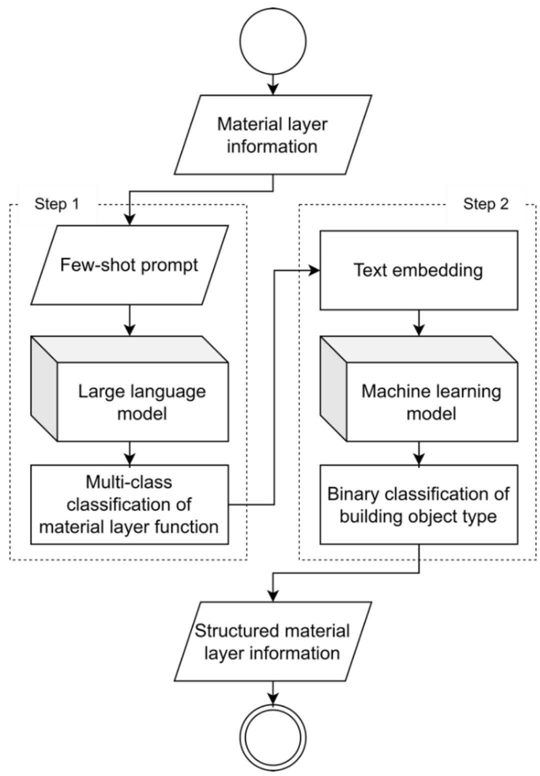

Figure 2: Flow chart of proposed method

Multi-class Classification of Material Layer Function

The first stage involves classifying the function of each material layer. BIM authoring tools require essential information, including material name, thickness, and function, to create compound walls and slabs with multiple material layers.

For instance, Graphisoft's ArchiCAD mandates the classification of material function into three categories: core, finish, and others. Similarly, Bentley's OpenBuildings Designer requires specifying whether each material constitutes a core layer (Bentley Systems, 2023; Graphisoft, 2024). Meanwhile, Autodesk's Revit categorizes material function into five classes: structure, substrate, thermal, membrane, and finish (Revit API Docs, 2024). Since Revit demands a more intricate classification system than other authoring tools, it inherently encompasses the classification schemes used by other software. Accordingly, this study considers Revit as the representative BIM authoring tool for material function classification and adopts its classification system.

The process of classifying material functions based on material layer names proceeds as follows. Initially, the material names are input into a few-shot prompt, as illustrated in Table 2, and then provided to an LLM. Then, the outputs of LLM are mapped back to the original material layer information and stored in CSV format.

Binary Classification of Building Object Type

The second stage involves classifying the object type indicated by the material layer information. Walls and slabs have distinct material compositions, and this study leverages these characteristics to perform binary classification using machine learning models. The classification model, Random Forest, is pre-trained on a dataset and the material layer information is text-embedded using FastText for material names and functions. The selection of these models was derived from the combination that achieved the best performance in the experiments. Then, the classification results were integrated into the CSV file generated in Step 1, structuring the dataset for further analysis.

Experiment Design

The experiment in this study is divided into three sub-steps, as illustrated in Figure 3: 'Data Collection and Preprocessing,' 'Prompt and Models Preparation,' and 'Implementation within the Proposed Method.'

The first sub-step involves the collection and preprocessing of datasets for few-shot prompting and classification models training. In this study, data are extracted from 20 BIM models that have undergone detailed design using the Revit API. The target data is the material layers of compound walls and slabs, which are system families within the models. The collected data are stored in CSV format, including material names, thicknesses, and function information. Subsequently, duplicate material layer information is removed by researchers, and material names that have the same meaning but are expressed differently are standardized to construct the dataset.

The second sub-step focuses on generating few-shot prompts for multi-class classification and training machine learning models for binary classification using the dataset. The detailed process for few-shot prompt generation is as follows: The material names from the dataset is input into a zero-shot prompt, which is then fed into an LLM. The LLM classifies the material function based on the prompt, and the output is manually reviewed and evaluated by researchers. The dataset's material function information serves as the ground truth. Ultimately, pairs of material names and functions that were misclassified are incorporated into the instruction of the prompt to generate the few-shot prompt.

Table 2: Few-shot prompt for multi-class classification of material layer function

Few-shot Prompt
- Material name: *user input* - Question: Which of the following five types does each layer of material fall into? The five types and their definitions are as follows. 1) Structures: Layer that supports the remainder of the wall, floor, or roof. 2) Substrate: Material, such as plywood or gypsum board, which acts as a foundation for another material. 3) Thermal: Provides insulation and prevents air penetration. 4) Membrane: A membrane that commonly prevents water vapor penetration. The membrane layer should have zero thickness. 5) Finish: Finish is typically the exterior or interior layer. ※ Below are the examples. Example 1) Input = glass wool, output = Thermal, Example 2) Input = wallpaper, output = finish - Instruction: You should list only the type names in the order of the input. No explanation is necessary. ※ And please refer to the list below to answer. Metal Deck: Structure, Bracket: Structure, Cement Mortar: Substrate, Protective Mortar: Substrate, Unreinforced Concrete: Substrate, Lightweight Foamed Concrete: Substrate, Floor Drain: Substrate, Cushioning: Substrate, PE Protectant: Substrate, OA Floor: Substrate, Gypsum board: Substrate, Bead Method Insulation: Thermal, Laminate Flooring: Finish, Brick: Finish

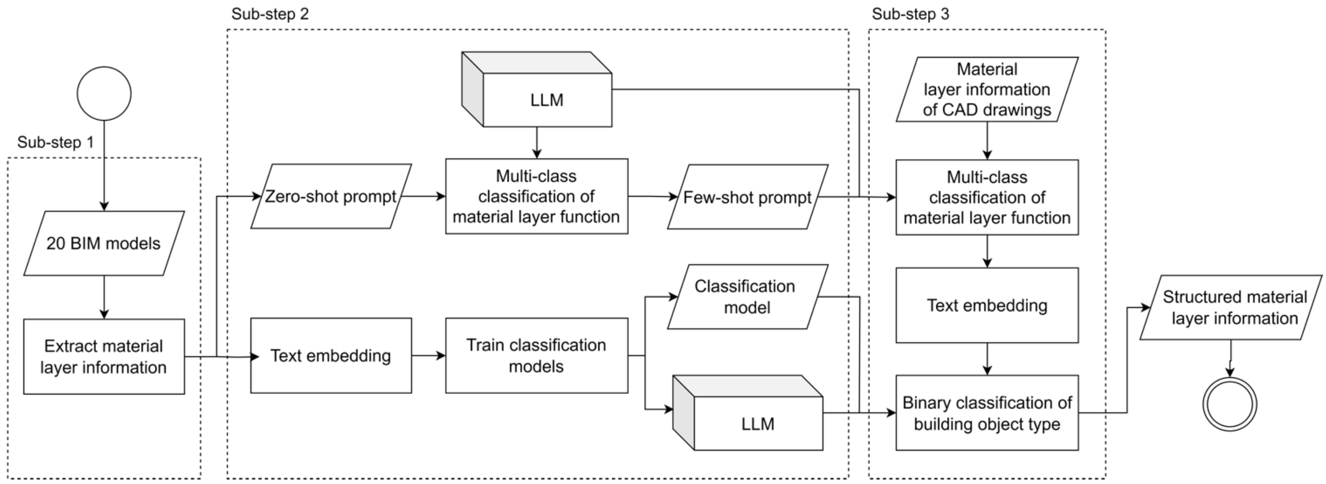

Figure 3: Process of research experiment

The detailed steps for binary classification model training are as follows: First, the dataset is analyzed to determine whether walls and slabs exhibit characteristic material combinations. Subsequently, the material names and functions in the dataset undergo text embedding using both FastText and text-embedding-ada-002 which is LLM embedding model. The resulting vectors are then split into training and testing datasets in an 8:2 ratio for model training. The classification models used include K-Nearest Neighbor, Random Forest, Support Vector Machine, AdaBoost, XGBoost, Decision Tree, and Logistic Regression.

In the third sub-step, the few-shot prompt and classification models developed in the previous steps are utilized to validate the proposed method. Additionally, for binary classification performance comparison, an alternative classification approach is employed, in which material layer information is directly input into the LLM without undergoing a training process. The validation dataset consists of material layer information extracted from 7 detailed design CAD drawings. The evaluation metric used is macro F1-score, which is adopted to provide a balanced evaluation across all classes, regardless of their frequency. This validation assesses how effectively the proposed method fills in the missing material layer information.

Experiment Results

Data Collection and Preprocessing

In this study, material layer information was collected from compound wall and slab objects modeled in 20 BIM models. The BIM models comprise 14 apartment buildings, 1 modular house, 1 neighborhood commercial facility, 2 educational facilities, and 2 office buildings. After eliminating duplicates, a total of 121 compound walls and 141 compound slabs were collected, resulting in a total of 262 objects. The total number of materials composing these objects was found to be 864.

Among the collected materials, words with identical meanings but different lexical expressions were standardized. Because it makes the structured generation of few-shot prompt and model training challenging. As a result, the dataset was refined to 50 unique terms, with 10 terms shared between walls and slabs. These common materials include 'air,' 'ceramic tile,' 'extruded polystyrene foam,' 'fiber-reinforced cement board,' 'cement mortar,' 'cement waterproofing,' 'porcelain tile,' 'concrete,' 'paint,' and 'granite.' These materials were identified as widely used in both walls and slabs.

Prompt and Models Preparation

Without providing any additional information, only the task and descriptions of function classes were given. After extracting material name information from the dataset, the data was fed into the GPT-4 Turbo and Omni models. The dataset also included material functions, which were used as the ground truth. As a result, it was found that function inference failed for a total of 10 materials: ‘cement mortar’, ‘floor cushioning material’, ‘protective mortar’, ‘PE protector’, ‘metal deck’, ‘OA floor’, ‘drainage board’, ‘brick’, ‘bracket’, and ‘gypsum board’. Most of the failed cases involved materials classified as substrates, while metal deck and bracket, which belong to the structure category, were also inferred incorrectly. These failed materials were paired with their correct classifications and used for generating few-shot prompts.

Figure 4 presents the results of training and testing 7 different machine learning models using the material layer dataset. The dataset was first processed through text embedding and encoding, then split into an 8:2 ratio for training and testing. Although it was initially expected that LLM-based embedding would outperform FastText due to its ability to comprehend material semantics and material layer contexts, both embedding methods achieved high performance, with AdaBoost, XGBoost, Random Forest, and Decision Tree models all achieving accuracy of 0.90 or higher.

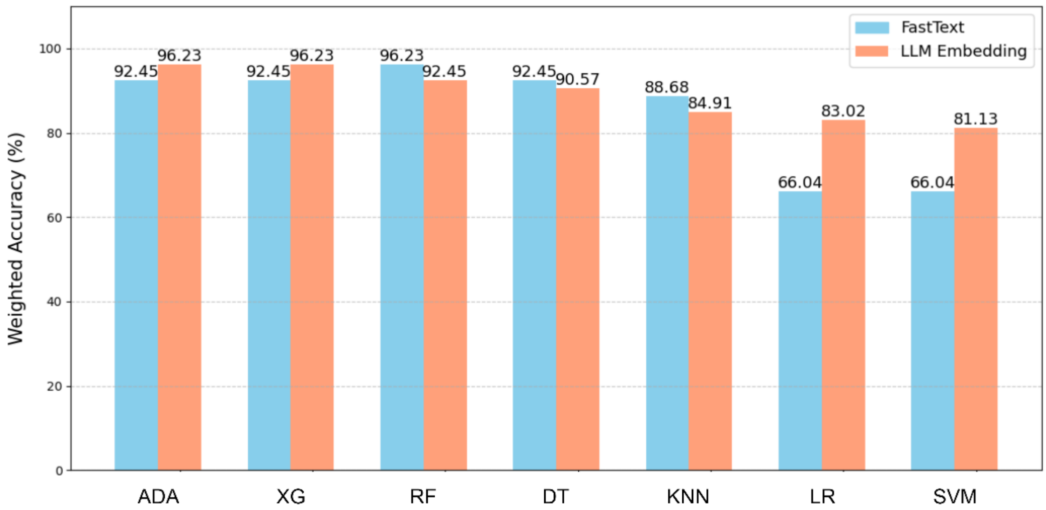

Figure 4: Binary classification of building object type performance on datasets

Implementation within the Proposed Method

In this phase, 7 CAD drawings from architectural design offices that had completed detailed design were used. The collected drawings consisted of 3 apartment buildings, 3 neighborhood commercial facilities in Korea, and 1 small-scale wooden house in Canada. Within these 7 drawings, a total of 59 compound walls and compound slabs were identified, comprising 172 distinct materials.

First, multi-class classification was conducted on the material layer information in each CAD drawing to determine the material functions using the previously constructed few-shot prompts. The results are presented in Figure 5. The GPT-4 Omni model outperformed the GPT-4 Turbo model, and the use of few-shot prompts resulted in better performance compared to zero-shot prompts. Ultimately, the GPT-4 Omni model with few-shot prompting achieved a macro F1-score of 99.8%.

Second, Figure 6 illustrates the binary classification results of material layers into object types using the material names from the CAD drawings and the inferred material function information obtained through few-shot prompting. The 4 machine learning models with superior performance and the 2 text embedding models were applied, generating a total of 8 classification results. Additionally, binary classification was performed using only the GPT-4 Omni model. The best performance was achieved by the combination of the FastText model and the Random Forest model, which attained a macro F1-score of 89.5%. Following this, the combination of the FastText model and the K-Nearest Neighbor demonstrated strong performance with a macro F1-score of 86.5%, while the standalone GPT-4 Omni model achieved a macro F1-score of 86.2%.

Discussion

The filling in the missing information process required by BIM authoring tools was successfully conducted using the proposed methods. However, each method exhibited certain distinctive characteristics.

Initially, in the ‘Multi-class Classification of Material Layer Function’ step, the GPT-4 Omni model with few-shot prompting achieved 99.8% macro F1-score. The only misclassified case was where the model failed to respond to the material ‘decorative tile finish’. Additionally, significant performance differences were observed between different LLMs and depending on whether a prompt was used. For instance, GPT-4 Turbo with zero-shot prompting misclassified ‘wood flooring’, ‘wood stud’, ‘drain board’, ‘protective mortar’, ‘cement mortar’, and ‘cushioning material’ as substrate, thermal, membrane, finish, finish, and structure, respectively. In contrast, GPT-4 Omni, even with zero-shot prompting, correctly classified ‘wood flooring’, ‘wood stud’, and ‘drain board’ as finish, structure, and substrate, demonstrating superior performance. While the misclassification rate for ‘protective mortar’ and ‘cement mortar’ has decreased, errors still occur in some cases.

With few-shot prompting, GPT-4 Turbo was able to resolve most of the previously mentioned issues through prompt engineering. However, inconsistencies observed in cases where materials had multiple functions, such as ‘lightweight aerated concrete’ and ‘transparent epoxy coating,’. These materials were alternatively classified as thermal or structure, and as finish or membrane, depending on the context. In contrast, GPT-4 Omni with few-shot prompt provided consistently accurate responses.

For the ‘Binary Classification of Building Object Type’ step, the combination of the FastText model and the Random Forest model yielded the highest macro F1-score at 89.5%.

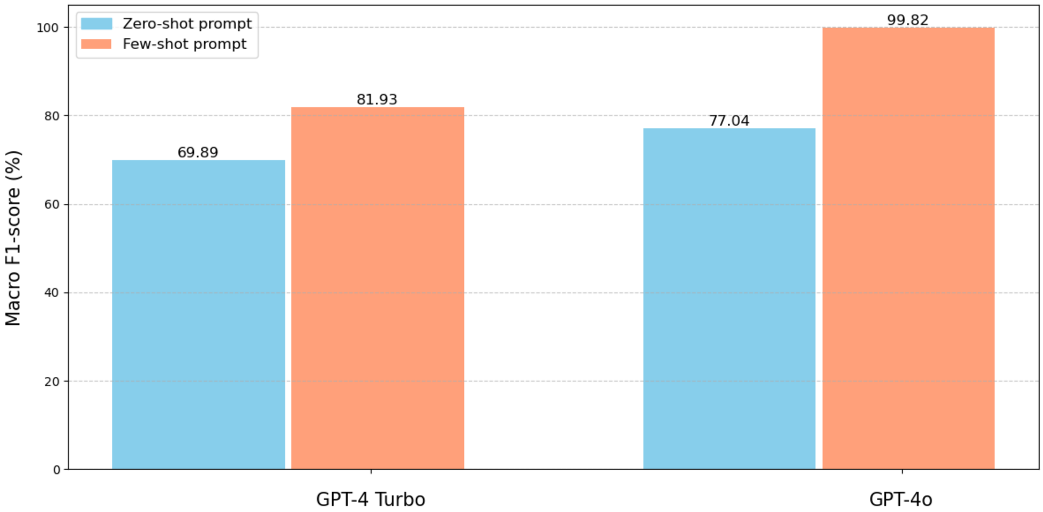

Figure 5: Multi-class classification of material layer function performance on CAD drawings

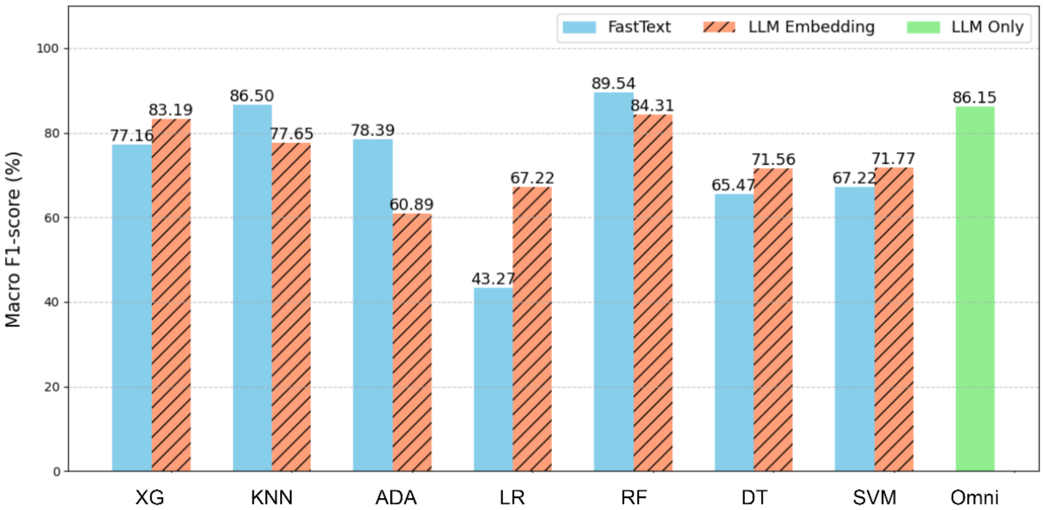

Figure 6: Binary classification of building object type performance on CAD drawings

However, 3 compound wall material layers — 'granite finish, extruded polystyrene,' 'granite honed finish, extruded polystyrene,' and 'penetrative liquid waterproofing, plain concrete' — were misclassified as slabs. This misclassification occurred because, in the training dataset, the materials 'granite,' 'extruded polystyrene,' and 'plain concrete' appeared more frequently in slabs than in walls. However, since these material layer information consisted of only two materials and lacked contextual information, and since these materials are commonly found in both walls and slabs, these misclassifications cannot be considered complete failures.

Additionally, the GPT-4 Omni model, despite not receiving any additional prompts beyond the task, demonstrated the third highest performance. This result indicates that pre-trained LLMs are highly effective in classifying object types based on material layer information. It is expected that, with further fine-tuning or the application of prompt engineering, LLMs could achieve even higher classification macro F1-score than traditional machine learning models.

Conclusion

This study aims to fill in the missing material layer information in CAD drawings to meet the object library creation requirements of BIM authoring tools. To achieve this, material function classification and object type classification were conducted using machine learning models and LLMs. This hybrid approach achieved a macro F1-score of 99.8% for classifying material layers functions and 89.5% for categorizing object type of material layer information, successfully filling in the missing information.

This research contributes to the CAD-to-BIM automation field, which often overlooks semantic information such as material layers, by filling in the missing material layer information in CAD drawings. It presents an approach for classifying the material functions based solely on their names and effectively categorizing objects using both material names and inferred material function.

Nevertheless, this study has two major limitations. First, the standardization of the dataset for few-shot prompt generation and machine learning model training resulted in a loss of flexibility. The existing dataset consists of 262 composite objects made up of 864 materials. However, despite having the same meaning, these 864 materials are expressed differently, making the structured generation of few-shot prompt and model training challenging. To address this issue, the vocabulary was standardized, reducing the number of distinct material expressions to 50. However, this process fails to encompass the diverse material terminology arbitrarily used by designers in CAD drawings. Therefore, future research should focus on either expanding the dataset or emphasizing dynamic inference methods. Second, this study is limited to internal walls and slabs, excluding various object types such as ceilings, exterior walls, and foundations. Future research should extend the proposed algorithm to encompass all object types. Advancing such studies would further enhance the semantic knowledge within the CAD-to-BIM domain.

Acknowledgement

This work was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (No. RS-2024-00407028) and the Hans Fischer Senior Fellowship program at the Technical University of Munich - Institute for Advanced Studies (TUM-IAS) in Germany.

References

Baek, S., Han, S.H. & Jung, W. (2023) Automated Identification of Active Players for International Construction Market Entry Using Natural Language Processing. Journal of Management in Engineering, 39(5), 04023025.

Baek, S., Namgoong, D., Won, J. & Han, S.H. (2023) Automated Detection of Social Conflict Drivers in Civil Infrastructure Projects Using Natural Language Processing. Applied Sciences, 13(20), 11171.

Bentley Systems (2023) OpenBuildings Designer 2023.

Byun, Y. & Sohn, B.-S. (2020) ABGS: A System for the Automatic Generation of Building Information Models from Two-Dimensional CAD Drawings. Sustainability, 12(17), 6713.

Graphisoft (2024) Archicad 28 API Archicad 28 C++ API.

Jeon, K., Lee, G., Yang, S. & Jeong, H.D. (2022) Named entity recognition of building construction defect information from text with linguistic noise. Automation in Construction, 143(104543).

Lee, K. & Lee, G. (2023) 2022/2023 Survey on the Status of BIM Adoption in Korea. Seoul, Korea: Building Informatics Group, Yonsei University.

Liao, W., Lu, X., Huang, Y., Zheng, Z. & Lin, Y. (2021) Automated structural design of shear wall residential buildings using generative adversarial networks. Automation in Construction, 132, 103931.

Moon, S., Chung, S. & Chi, S. (2020) Bridge Damage Recognition from Inspection Reports Using NER Based on Recurrent Neural Network with Active Learning. Journal of Performance of Constructed Facilities, 34(6), 04020119.

Moon, S., Lee, G. & Chi, S. (2022) Automated system for construction specification review using natural language processing. Advanced Engineering Informatics, 51, 101495.

Rho, J., Lee, H.-S. & Park, M. (2020) Automated BIM Generation Using Drawing Recognition and Line-Text Extraction. Journal of Asian Architecture and Building Engineering, 20.

Wu, J., Akanbi, T. & Zhang, J. (2022) Constructing Invariant Signatures for AEC Objects to Support BIM-Based Analysis Automation through Object Classification. Journal of Computing in Civil Engineering, 36(4), 04022008.

Yang, B., Liu, B., Zhu, D., Zhang, B., Wang, Z. & Lei, K. (2020) Semiautomatic Structural BIM-Model Generation Methodology Using CAD Construction Drawings. Journal of Computing in Civil Engineering, 34, 04020006.

Yin, D., Deng, X. & Lu, K. (2024) Automatic generation of pipelines from 2D CAD drawings to 3D IFC models using a parametric method. International Journal of Construction Management, 0(0), 1–19.

Zeng, Z., Li, X., Yu, Y.K. & Fu, C.-W. (2019) Deep Floor Plan Recognition Using a Multi-Task Network With Room-Boundary-Guided Attention. In. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, 9095–9103.