

Classifying Schedule Activities Using BERT & RoBERTa (A Case Study)

Mohamed ElSaadany

Senior Manager, PMC
Khatib & Alami



ABOUT THE SPEAKER



Mohamed ElSaadany

Senior Manager, PMC

Khatib & Alami

Founder of

SeedsWise.com

Academic & Professional Certificates

- PhD. Degree in Construction Management [AI and 4D BIM], Mansoura University, Egypt.
- MSc. Degree in Construction Management [PMO Processes in Program Management], Helwan University.
- Master of Business Administration [Finance] from Leicester University, UK.
- BSc. in Civil Engineering, Alexandria University.
- Level-7 Diploma in Strategic Management from the Chartered Management Institute (CMI), UK.
- Professional Diploma in Forensic Schedule Analysis from the College of Contract Management, UK.
- Professional certificates, including PMP, CCP, PMI_RMP, and P3O.

Summary of Work Experience

- 23 years of experience in construction projects in multiple sectors, including residential, mixed-use, hospitality, medical, harbors, and industrial.
- Consultancy experience in PMO setup and operation.
- PMC business development, including the development of technical and commercial proposals.
- leading planning, reporting, and performance management functions for complex projects.
- Developing and maintaining key artifacts, including PEP, PMP, EVM, forecasting, and KPIs.
- Instructor – Project Management and Cost Engineering.

TABLE OF CONTENT

- 1. Fine-tuning of Large Language Models (LLM)**
- 2. A Case Study on Automating the 4D BIM**
- 3. Reflection and Implications**
- 4. Questions**

HELP PERSONALIZE THIS DEMO

1. Your AI Experience?



Group A

I'm currently building AI, Machine Learning, or complex automation into my project controls systems.

Group B

I'm here to learn the fundamentals of AI's practical benefits and how to get started.

HELP PERSONALIZE THIS DEMO

2. Your Professional Focus?



Group A

I'm interested in exploring research gaps, developing new theories, and tailoring frameworks.

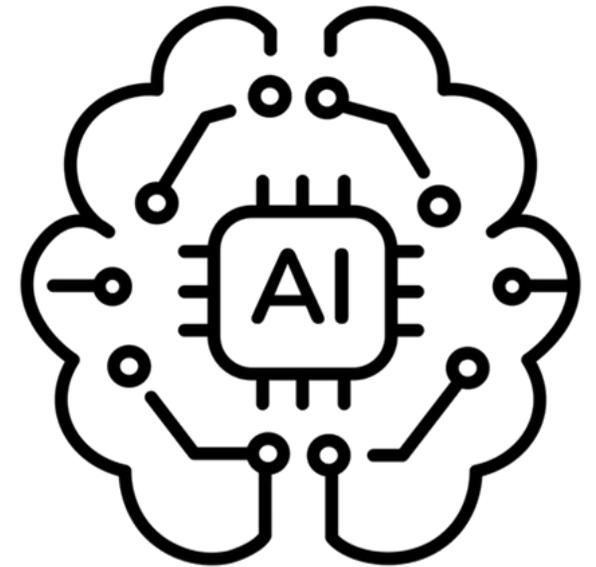
Group B

I'm interested in applying existing tools to deliver large-scale projects on time and on budget.

FINE-TUNING OF LARGE LANGUAGE MODELS (LLMs)

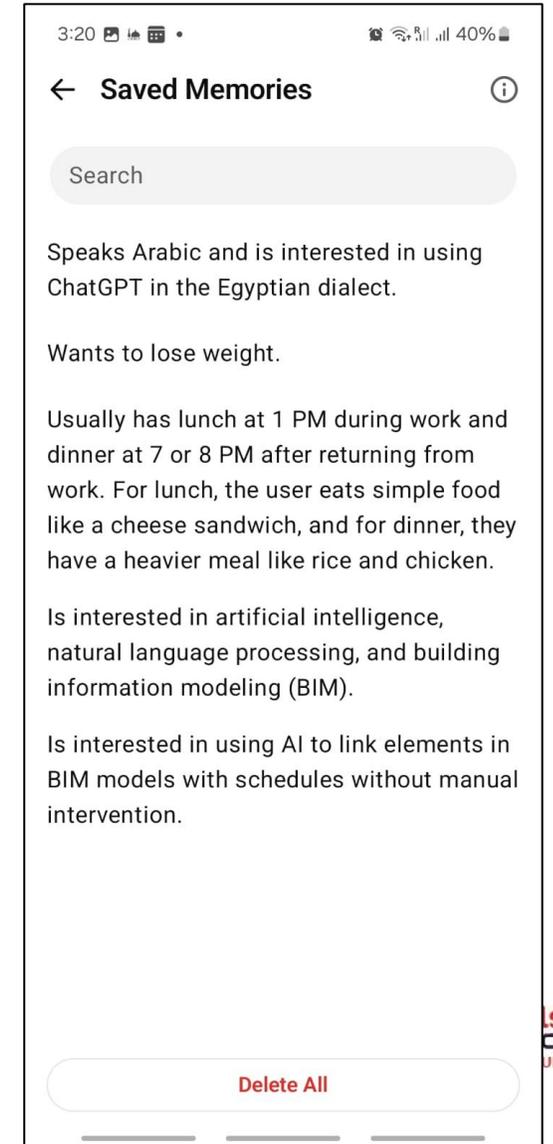
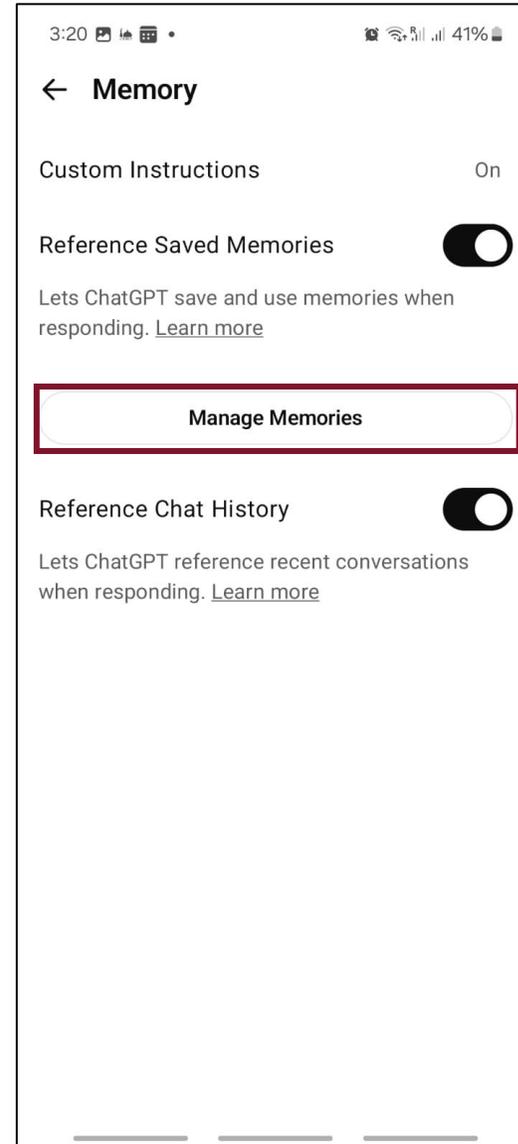
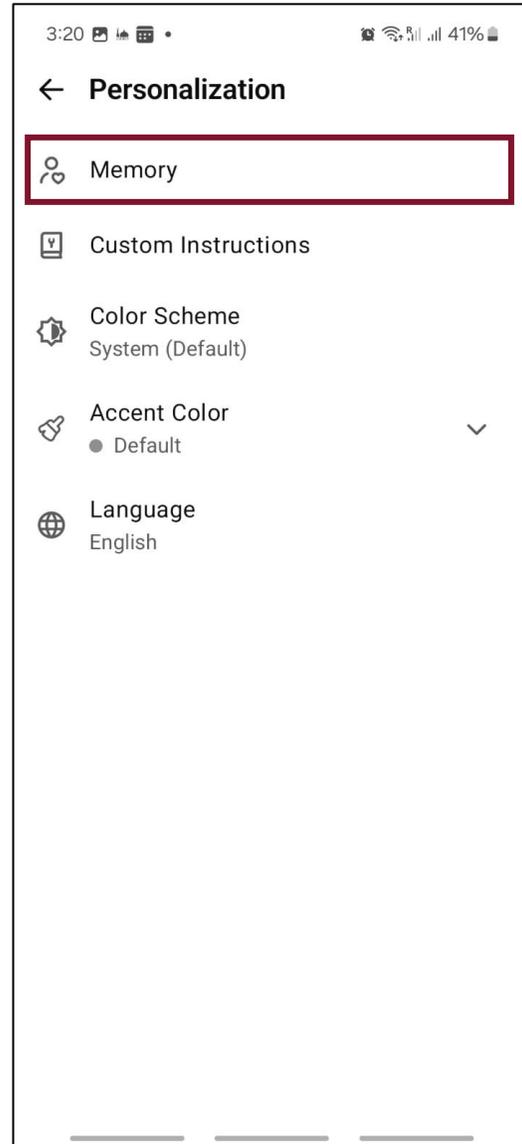
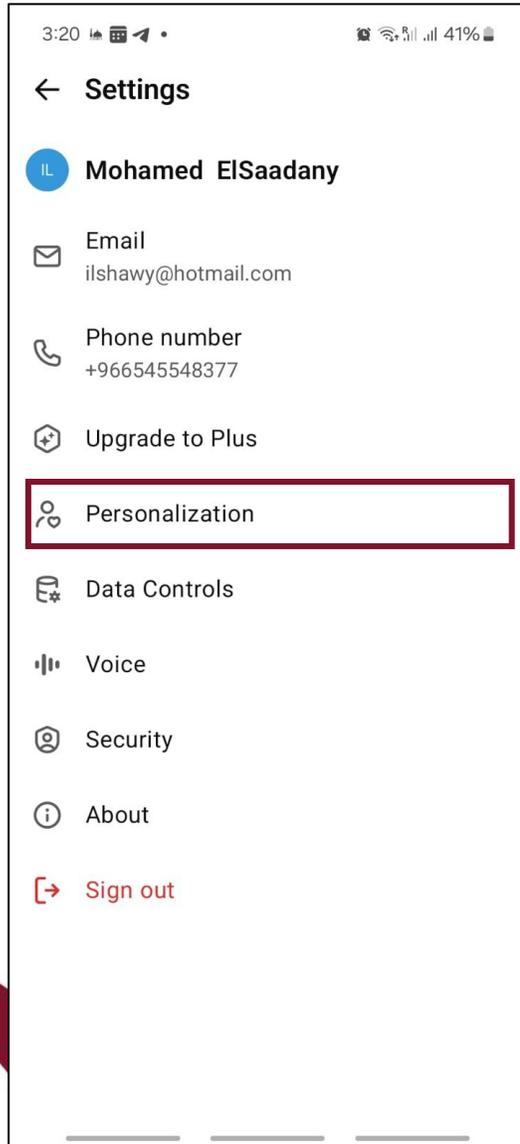
TO WHAT EXTENT DO YOU AGREE WITH THIS STATEMENT?

Every time you speak to the model, you are immediately customizing its core intelligence to fit your way of working.



We need to differentiate between the model memory and the training process

EXAMPLE: CHATGPT MEMORY

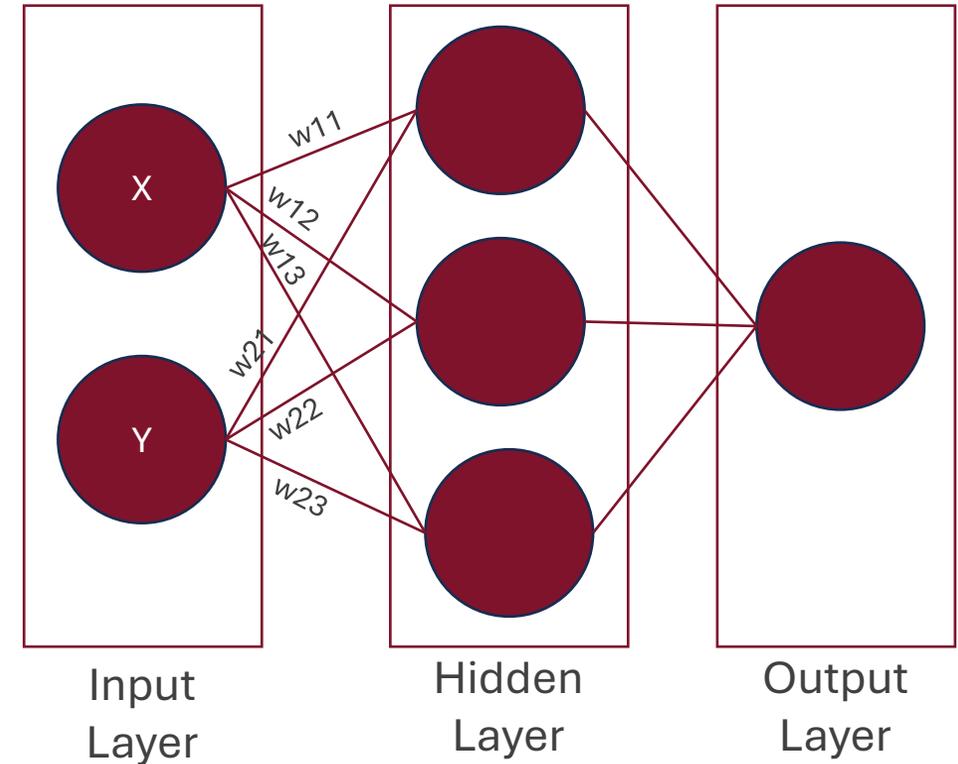


NEURAL NETWORK

Example:

Suppose you have a building of X m² and Y bedrooms.

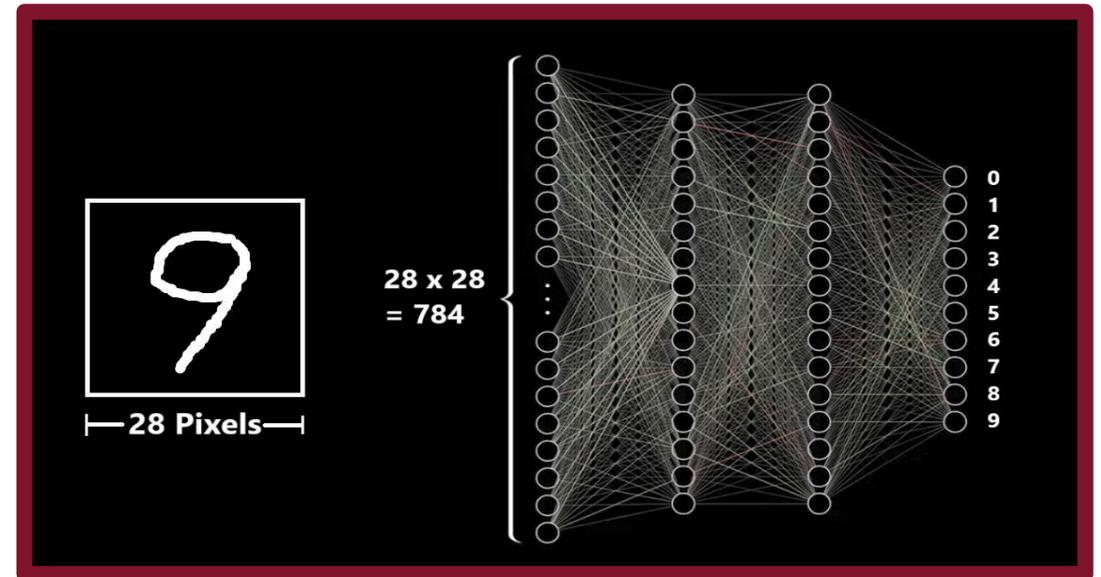
- Weights can be initially assumed.
- Weighted sum is calculated, then compared with the real result.
- During training, the network needs to learn the composition of weights and biases.



NEURAL NETWORK

Increasing the number of inputs and adding more hidden layers increases the depth of the neural network.

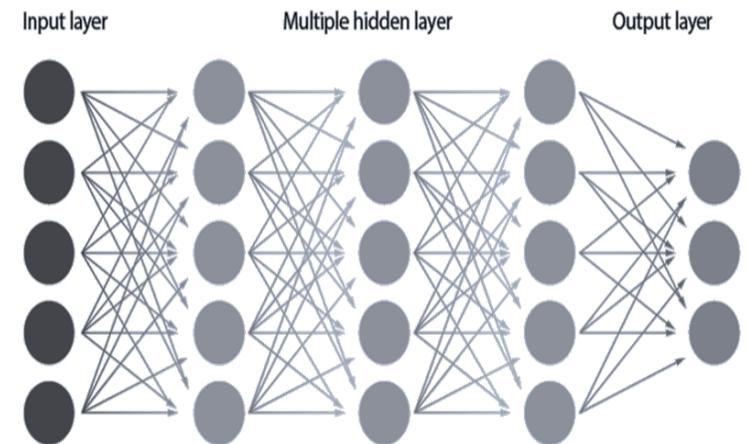
To represent an image of 28 x 28 pixels for recognizing a handwritten number, you need 784 neurons in the input layer.



LARGE LANGUAGE MODELS (LLMs) – The Generalists

What is an LLM?

- A deep learning model that is pre-trained on a massive corpus of text.
- They learn the statistical relationships between words and sentences, enabling them to understand, generate, and process human language.
- LLMs are powerful Generalists (unspecialized) with a broad understanding of semantics and common knowledge.



In highly complex models such as GPT3, 175 billion parameters with dozens of layers were used, taking a training duration of several weeks.

FINE-TUNING OF LLMs – The Path to Specialization

What is Fine-tuning?

- The process of taking a pre-trained LLM and continuing its training on a smaller, task-specific **dataset**.
- Think of an LLM as a brilliant, but general-purpose intern.
- Fine-tuning is the specialized, on-the-job training that turns that intern into an expert in a specific field, like scheduling or contract management.



Fine-tuning can be implemented to enhance the LLM capability in certain skills (e.g., **Text Classification, Text Processing, image analysis, etc.**)

FINE-TUNING PROCESS

1. Select a Pre-Trained Model

- Use a pre-trained LLM (e.g., BERT, RoBERTa, GPT-3, DeepSeek) as the starting point.
- These models have already learned general language representations.

2. Prepare a Dataset

- Collect a dataset relevant to the target task
- The dataset includes input-output pairs (e.g., text and corresponding labels).

3. Fine-Tune the Model

- Fine-tune the pre-trained model on the dataset
- During fine-tuning, the model's weights are **updated**.

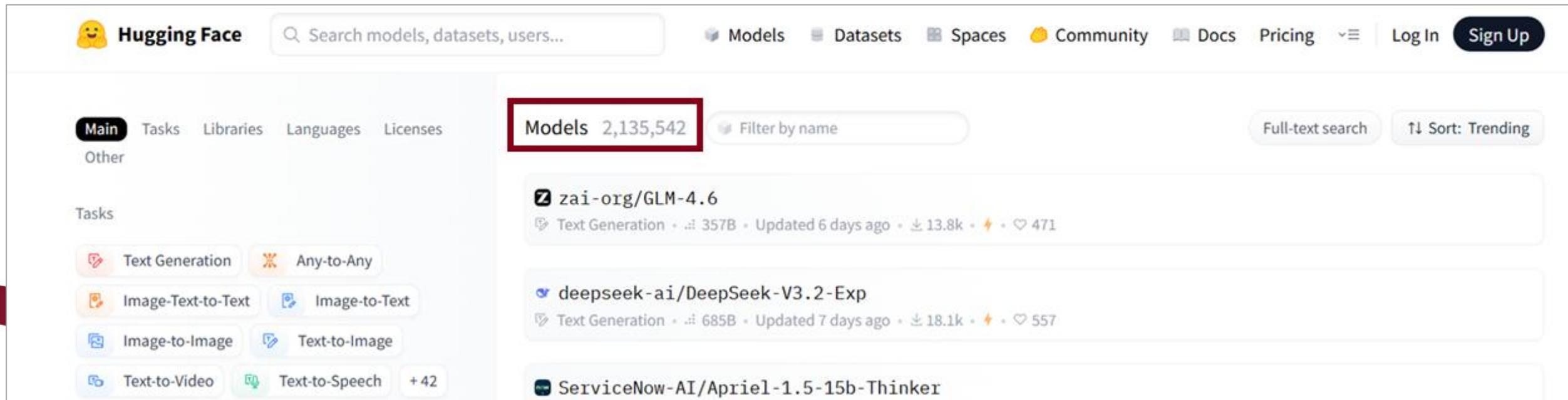
4. Evaluate and Iterate

- Test the fine-tuned model on a validation set.
- Calculate the precision of the fine-tuned model.

HUGGING FACE

- An open-source community (such as GitHub for Developers) that has become a central hub for Natural Language Processing (NLP) and Machine Learning (ML).
- It provides resources (models, datasets, spaces) to make it easier for researchers and developers to build, train, and deploy models.

<https://huggingface.co/models>



The screenshot displays the Hugging Face website interface. At the top, the Hugging Face logo is on the left, followed by a search bar with the placeholder text "Search models, datasets, users...". Navigation links for "Models", "Datasets", "Spaces", "Community", "Docs", "Pricing", "Log In", and "Sign Up" are visible. The "Models" section is highlighted with a red box, showing "Models 2,135,542" and a "Filter by name" input field. Below this, there are buttons for "Full-text search" and "Sort: Trending". The main content area lists several models, including "zai-org/GLM-4.6" (Text Generation, 357B, Updated 6 days ago, 13.8k downloads, 471 likes) and "deepseek-ai/DeepSeek-V3.2-Exp" (Text Generation, 685B, Updated 7 days ago, 18.1k downloads, 557 likes). A sidebar on the left contains navigation options like "Main", "Tasks", "Libraries", "Languages", "Licenses", and "Other", along with a "Tasks" section listing various model categories such as "Text Generation", "Any-to-Any", "Image-Text-to-Text", "Image-to-Text", "Image-to-Image", "Text-to-Image", "Text-to-Video", "Text-to-Speech", and "+42".

HUGGING FACE API

1. Simplify Access to Pre-trained Models like BERT, DeepSeek, GPT, etc.
2. Enable Integration of Hugging Face's tools into your own applications.
3. Facilitate collaboration by sharing models & datasets with the community.

To start using this API, you need the following:

- Install Python (<https://www.python.org/downloads>)
- Install Transformers library from CMD: (`pip install transformers`)
- ML framework: TensorFlow (`pip install tensorflow`) or PyTorch (`pip install torch`).
- CPU (for small tasks), GPU (for most deep learning tasks), or TPU (for very large models).
- Google Colab provides free access to CPUs, GPUs, and TPUs.

A CASE STUDY ON THE AUTOMATION OF THE 4D BIM MODEL

PROBLEM STATEMENT

The current practices for developing the 4D BIM model depend on either the manual classification and mapping or setting rule-based criteria.

Manual Classification and Mapping

The 4D BIM Modeler understands the scheduler's intention of each activity to manually link each activity to the relevant BIM element(s).

A time-consuming process

Rule-Based Criteria

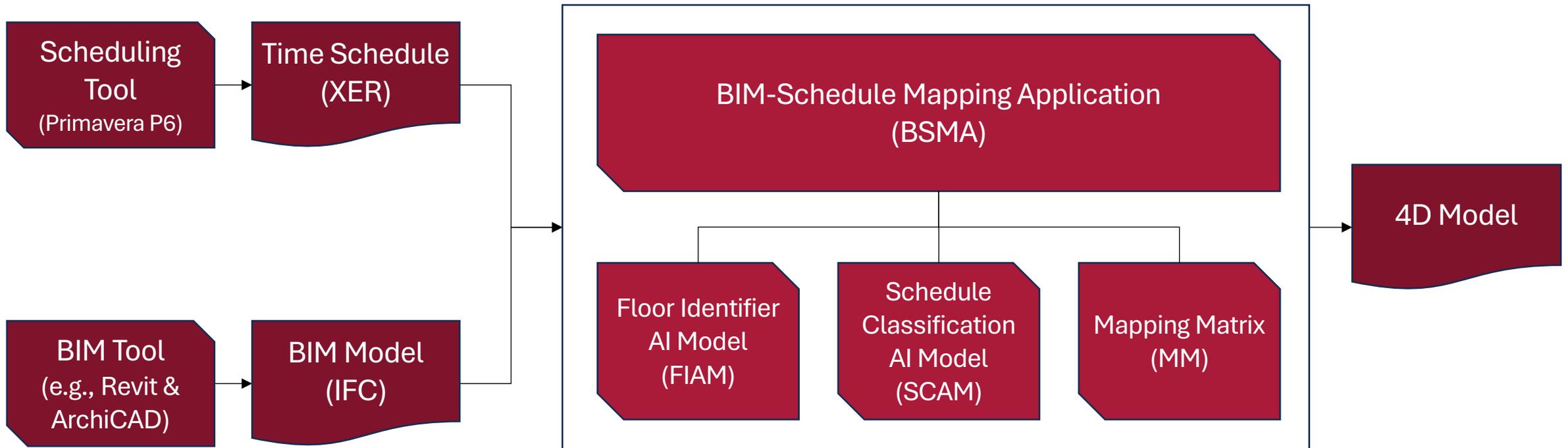
The BIM modelers have prior meetings with the schedulers to agree on a naming convention of the schedule activity codes.

It accelerates the process but still ineffective in case of existing schedules.

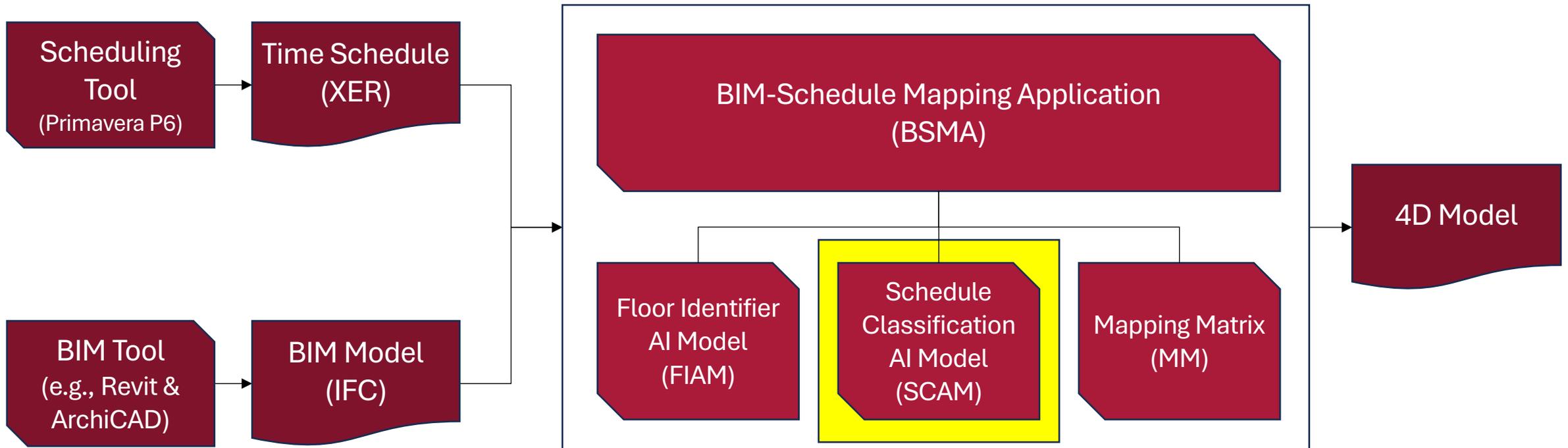
RESEARCH QUESTIONS

- How much structured is the data for both BIM models and project schedules?
- Do we need an intermediate/standard form of data structure (e.g., common names for building elements like OmniClass)?
- How to classify the schedule activities into building components?
- How to differentiate between the floors/stories of the building?
- How to map the BIM elements to the standard building elements?

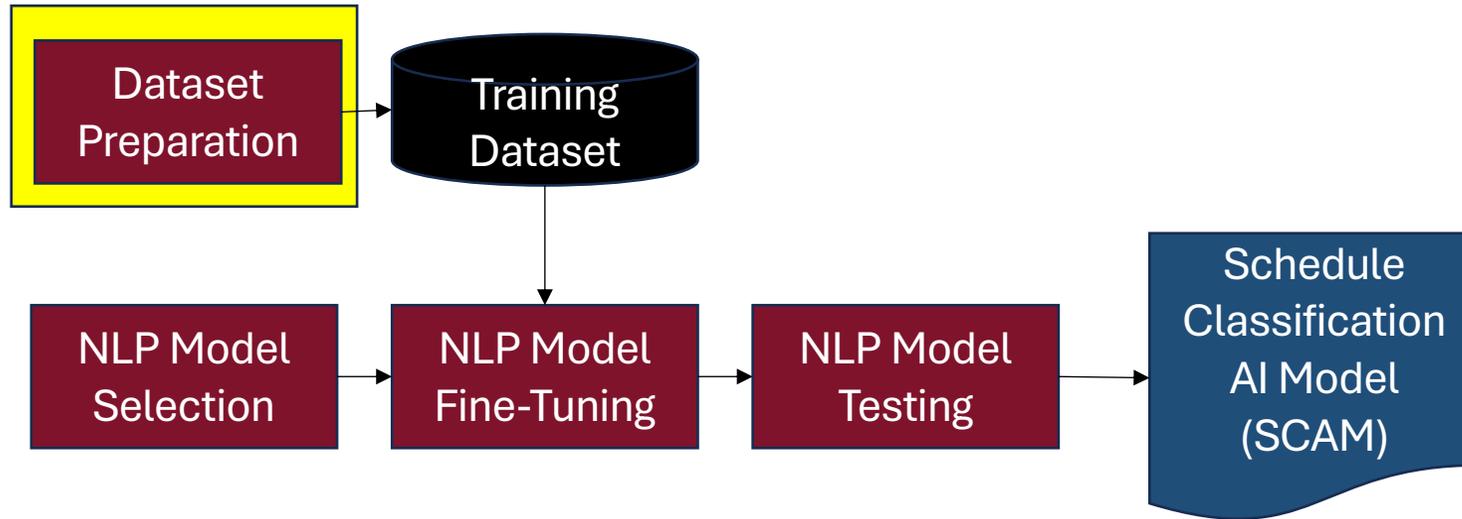
FRAMEWORK MODEL



FRAMEWORK MODEL

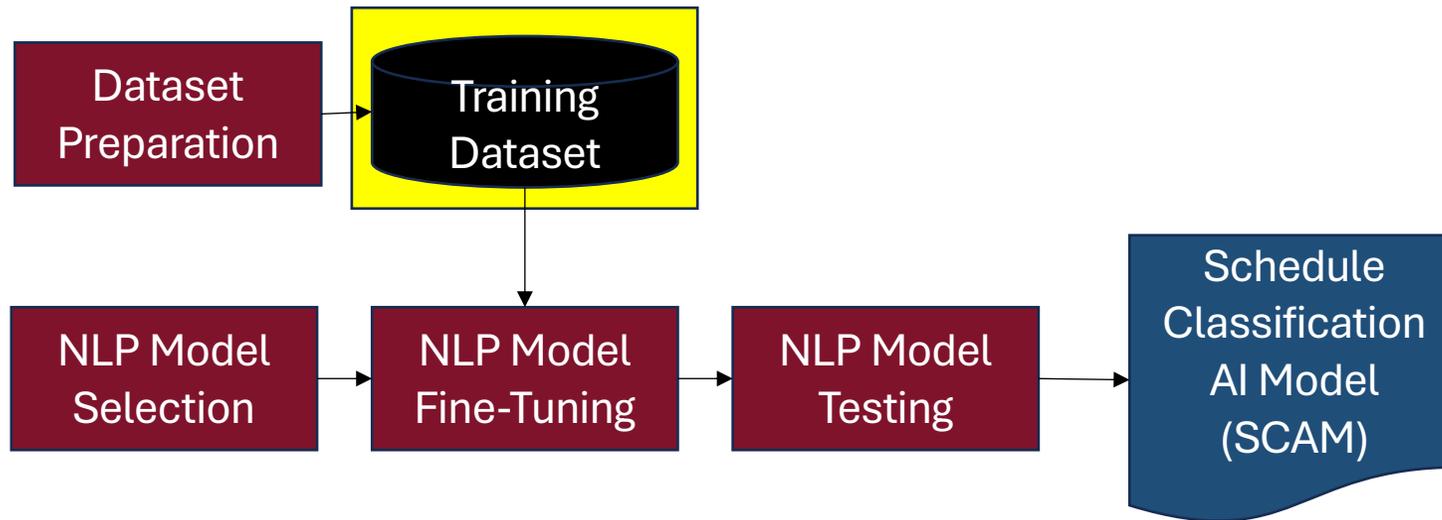


SCHEDULE CLASSIFICATION AI MODEL (SCAM)



- 76 schedules were collected from different building projects.
- After screening, 23 schedules were excluded.
- Activities were extracted from the remaining 53 schedules to a spreadsheet
- Repeated activities (e.g., same activity names for each floor) were not considered to minimize overfitting.

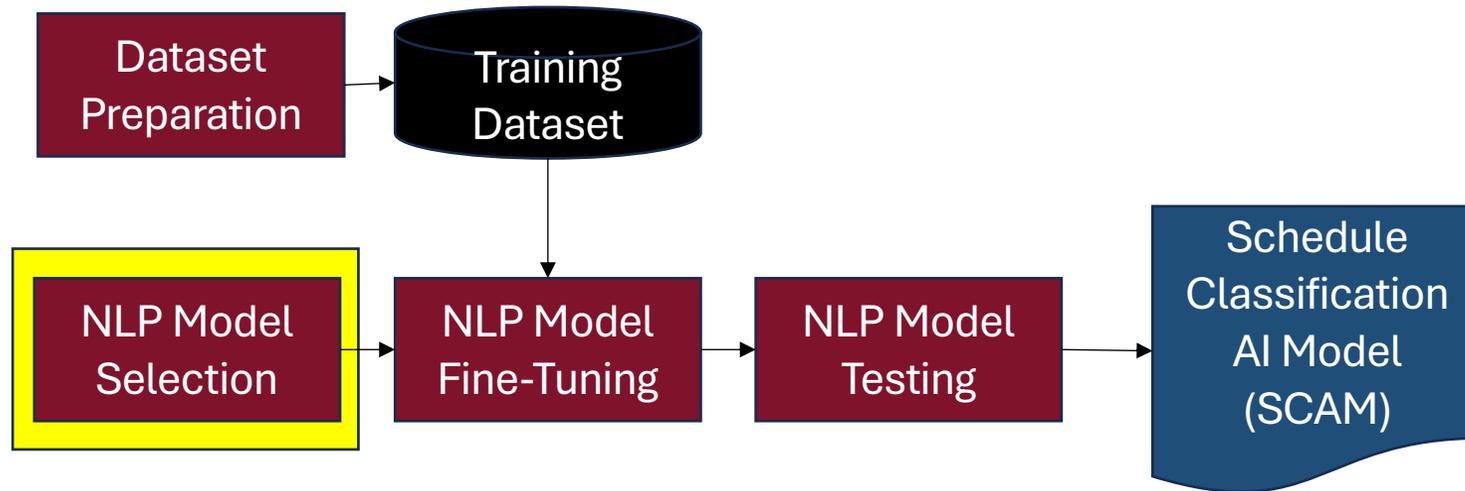
SCHEDULE CLASSIFICATION AI MODEL (SCAM)



- A total of 4231 activities were identified, sourced from 53 schedules.
- All the activities have been classified according to the OmniClass.
- The spreadsheet has been saved as a Comma-separated values file format (CSV).

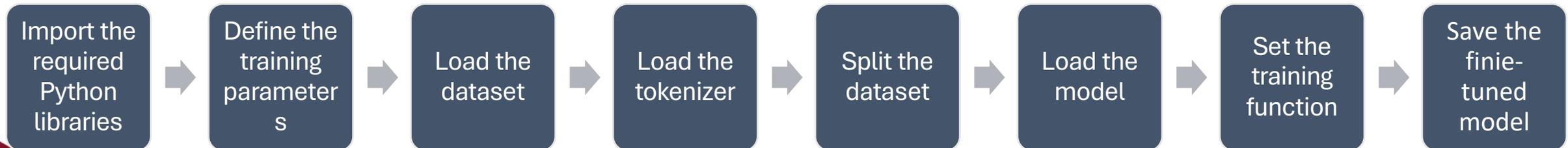
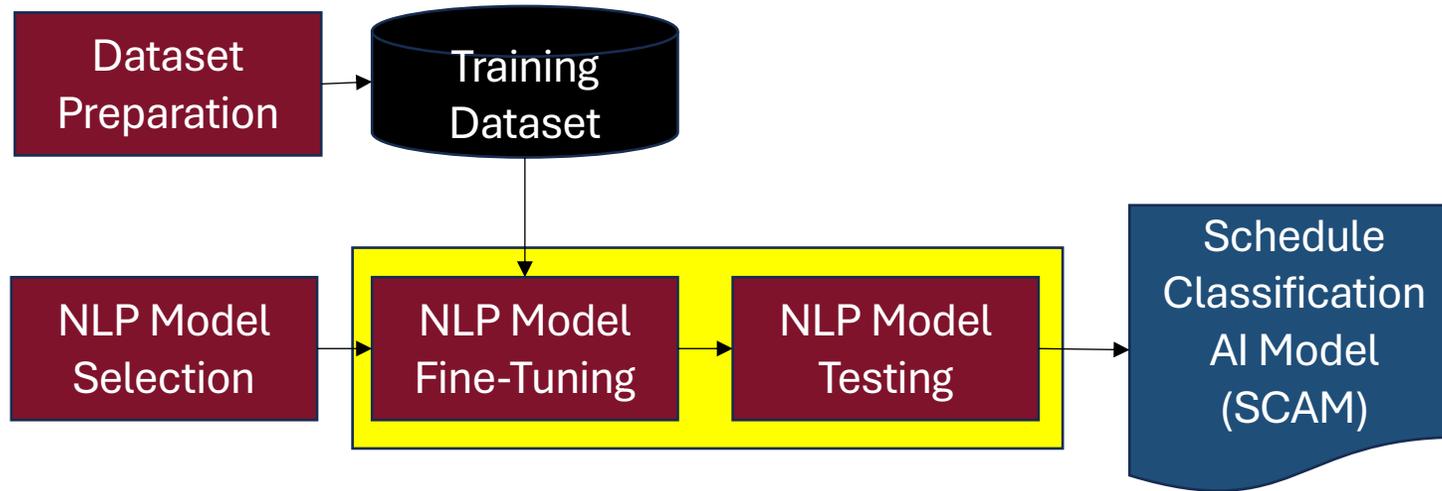
Schedule Activity	OmniClass
Columns upto Level 633	Columns
2nd Floor Slab	Floor_Decks_Slabs_and_Toppings
Roof Floor Slab	Floor_Decks_Slabs_and_Toppings
Block Work	Walls
Walls Partitions	Walls
Stairs for the building	Stairs
Windows installation	Windows
Wall Finishes	Wall_Painting_and_Coating
Ceiling Finishes	Ceiling_Painting_and_Coating
Doors Installation	Doors
Floor Finishes for the fourth floor	Tile_Flooring
Garbage chute Installation	Chutes
Plumbing Piping	Plumbing_Second_Fix
Accessories for toilets	Plumbing_Third_Fix
Emergency Chilled Water System	HVAC_Second_Fix
Insulation for HVAC piping	HVAC_Second_Fix
Fire Fighting System Piping Installation	Fire_Fighting_Second_Fix
Plumbing Fixturs	Plumbing_Third_Fix

SCHEDULE CLASSIFICATION AI MODEL (SCAM)



- There is no need for producing a text since the dataset in this research includes pairs of schedule activities and their classes. Therefore, generative pre-trained transformer (**GPT**) and text-to-text transfer transformer (**T5**) were excluded.
- On the other hand, Bidirectional Encoder Representations from Transformers (**BERT**) and Robustly Optimized BERT Pretraining Approach (**RoBERTa**) are designed to for text classification.

SCHEDULE CLASSIFICATION AI MODEL (SCAM)



SCHEDULE CLASSIFICATION AI MODEL (SCAM)

```
1  import torch
2  from torch.utils.data import Dataset, DataLoader
3  from transformers import AutoTokenizer, AutoModelForSequenceClassification, get_linear_schedule_with_warmup
4  import pandas as pd
5  from sklearn.model_selection import train_test_split
6  from sklearn.metrics import accuracy_score, classification_report
7  import numpy as np
8  from sklearn.preprocessing import LabelEncoder
9  from tqdm import tqdm # Import tqdm for progress bar
10
11 # 1. Define Training Parameters
12 print("#"*50)
13 FILE_PATH = '4dbim_ds_OC.csv'
14 TEXT_COLUMN = 'activity_name'
15 LABEL_COLUMN = 'var_name'
16 MAX_LEN = 128
17 BATCH_SIZE = 16
18 EPOCHS = 12
19 LEARNING_RATE = 2e-5
20 MODEL_NAME = 'bert-base-uncased'
21 # Check for GPU availability
22 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
23
```

SCHEDULE CLASSIFICATION AI MODEL (SCAM)

```
24 # 2. Load and Prepare the Dataset
25 def load_data(file_path, text_column, label_column):
26     """Loads CSV data and returns text and labels."""
27     df = pd.read_csv(file_path)
28     texts = df[text_column].tolist()
29     labels = df[label_column].tolist()
30     return texts, labels, df

```

```
66 # 3. Load Tokenizer and Model
67 tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
68 texts, labels, df = load_data(FILE_PATH, TEXT_COLUMN, LABEL_COLUMN)
69
70 label_encoder = LabelEncoder()
71 numerical_labels = label_encoder.fit_transform(labels)
72
73 # Split data into training (80%), temporary (20%)
74 train_texts, temp_texts, train_labels, temp_labels = train_test_split(
75     texts, numerical_labels, test_size=0.2, random_state=42, stratify=numerical_labels
76 )
77
78 # Split temporary data into validation (50% of temp = 10% of total) and testing (50% of temp = 10% of total)
79 val_texts, test_texts, val_labels, test_labels = train_test_split(
80     temp_texts, temp_labels, test_size=0.5, random_state=42, stratify=temp_labels
81 )
82
```

SCHEDULE CLASSIFICATION AI MODEL (SCAM)

```
85 # 4. Prepare Data Loaders
86 train_dataset = CustomDataset(train_texts, train_labels, tokenizer, MAX_LEN)
87 val_dataset = CustomDataset(val_texts, val_labels, tokenizer, MAX_LEN)
88 test_dataset = CustomDataset(test_texts, test_labels, tokenizer, MAX_LEN)
89
90 train_dataloader = DataLoader(train_dataset, batch_size=BATCH_SIZE, shuffle=True)
91 val_dataloader = DataLoader(val_dataset, batch_size=BATCH_SIZE)
92 test_dataloader = DataLoader(test_dataset, batch_size=BATCH_SIZE)
93
94 # 5. Define Optimizer and Scheduler
95 optimizer = torch.optim.AdamW(model.parameters(), lr=LEARNING_RATE)
96 total_steps = len(train_dataloader) * EPOCHS
97 scheduler = get_linear_schedule_with_warmup(
98     optimizer,
99     num_warmup_steps=0,
100     num_training_steps=total_steps
101 )
```

SCHEDULE CLASSIFICATION AI MODEL (SCAM)

```
140 # 7. Training and Evaluation with Monitoring
141 train_losses = []
142 val_accuracies = []
143
144 for epoch in range(EPOCHS):
145     train_loss = train_epoch(model, train_dataloader, optimizer, scheduler, device, epoch + 1, EPOCHS)
146     accuracy, report = eval_epoch(model, val_dataloader, device, label_encoder)
147     train_losses.append(train_loss)
148     val_accuracies.append(accuracy)
149     print(f'Epoch {epoch + 1}/{EPOCHS}, Train Loss: {train_loss:.4f}, Val Accuracy: {accuracy:.4f}\n{report}')
150
151 # 8. Evaluate on the Test Set
152 test_accuracy, test_report = eval_epoch(model, test_dataloader, device, label_encoder)
153 print(f'\n{"#"*30} Test Set Evaluation {"#"*30}')
154 print(f'Test Accuracy: {test_accuracy:.4f}\n{test_report}')
155
156 # 9. Save the Fine-Tuned Model
157 model.save_pretrained('./fine_tuned_bert')
158 tokenizer.save_pretrained('./fine_tuned_bert')
159 print("Fine-tuning complete. Model saved to ./fine_tuned_bert")
```


SCHEDULE CLASSIFICATION AI MODEL (SCAM)

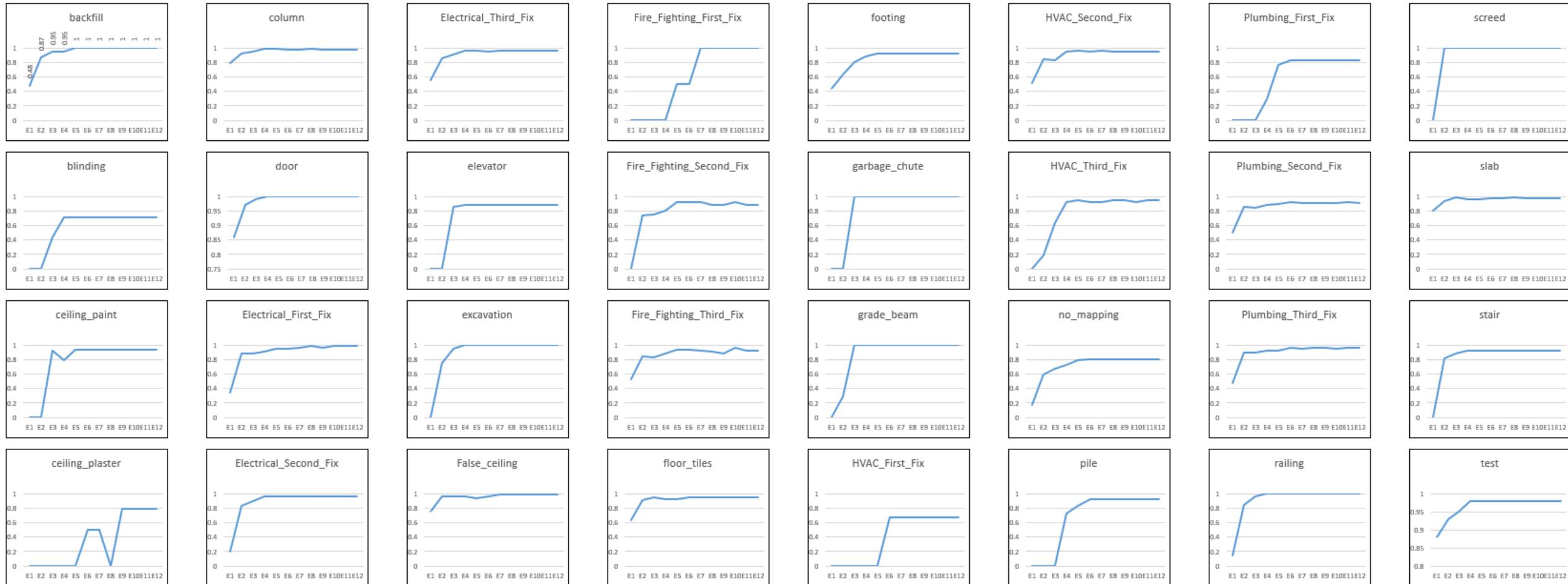
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Epoch 2/12: 100% [REDACTED] | 212/212 [07:52<00:00, 2.23s/batch, loss=0.814]

Epoch 2/12, Train Loss: 1.4372, Val Accuracy: 0.8428

	precision	recall	f1-score	support
Electrical_First_Fix	0.88	0.90	0.89	39
Electrical_Second_Fix	0.84	0.81	0.83	47
Electrical_Third_Fix	0.79	0.96	0.86	69
False_ceiling	0.92	1.00	0.96	34
Fire_Fighting_First_Fix	0.00	0.00	0.00	3
Fire_Fighting_Second_Fix	0.67	0.83	0.74	12
Fire_Fighting_Third_Fix	0.86	0.83	0.84	23
HVAC_First_Fix	0.00	0.00	0.00	2
HVAC_Second_Fix	0.74	0.97	0.84	29
HVAC_Third_Fix	1.00	0.11	0.19	19
Plumbing_First_Fix	0.00	0.00	0.00	6
Plumbing_Second_Fix	0.89	0.84	0.86	38
Plumbing_Third_Fix	0.83	0.97	0.90	40
backfill	0.77	1.00	0.87	10

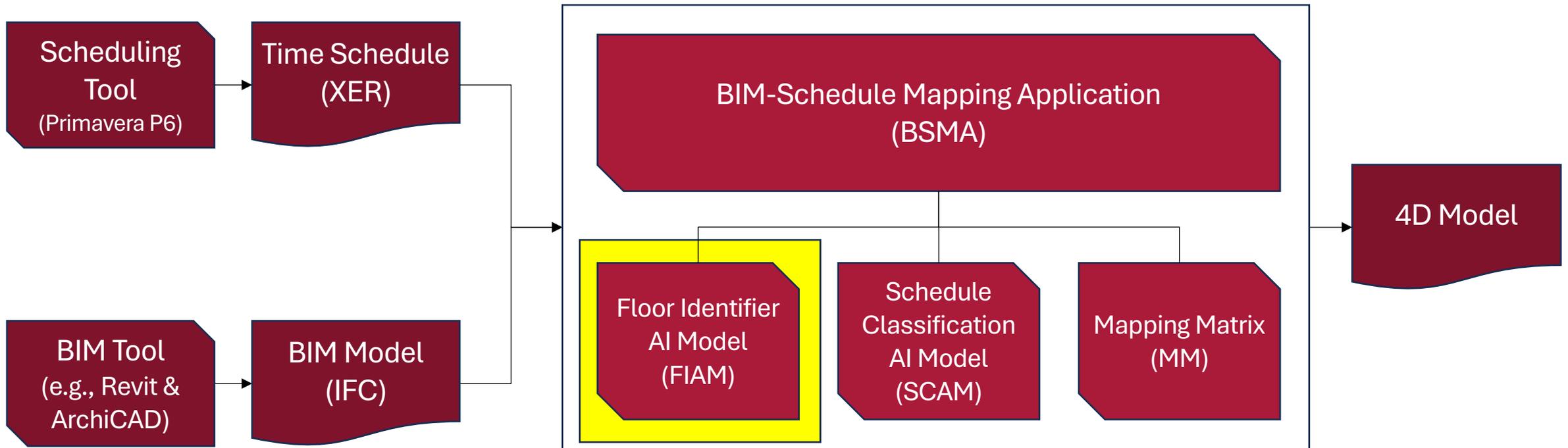
SCHEDULE CLASSIFICATION AI MODEL (SCAM)



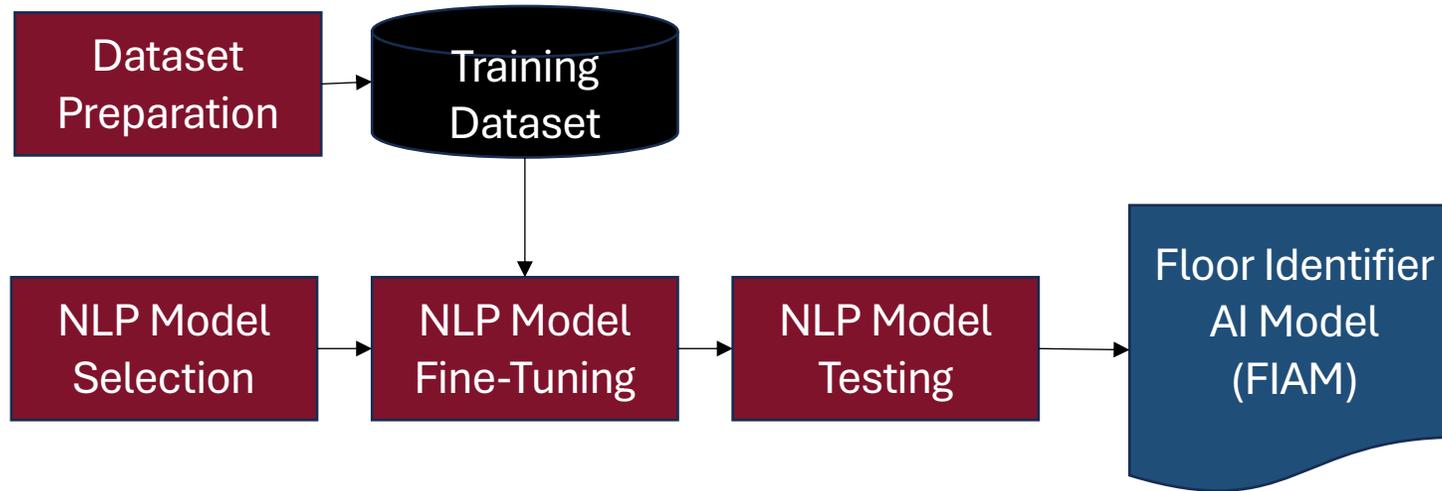
SCHEDULE CLASSIFICATION AI MODEL (SCAM)

<input type="checkbox"/> Name	Type	Size
<input type="checkbox"/>  config.json	JSON Source File	3 KB
<input type="checkbox"/>  model.safetensors	SAFETENSORS File	427,803 KB
<input type="checkbox"/>  special_tokens_map.json	JSON Source File	1 KB
<input type="checkbox"/>  tokenizer.json	JSON Source File	695 KB
<input type="checkbox"/>  tokenizer_config.json	JSON Source File	2 KB
<input type="checkbox"/>  vocab.txt	Text Document	227 KB

FRAMEWORK MODEL



FLOOR IDENTIFIER AI MODEL (FIAM)



- The dataset was created as a JSON Lines (JSONL) file.
- Each line is a JSON object representing a single training example.

FLOOR IDENTIFIER AI MODEL (FIAM)

wbs_dataset.jsonl

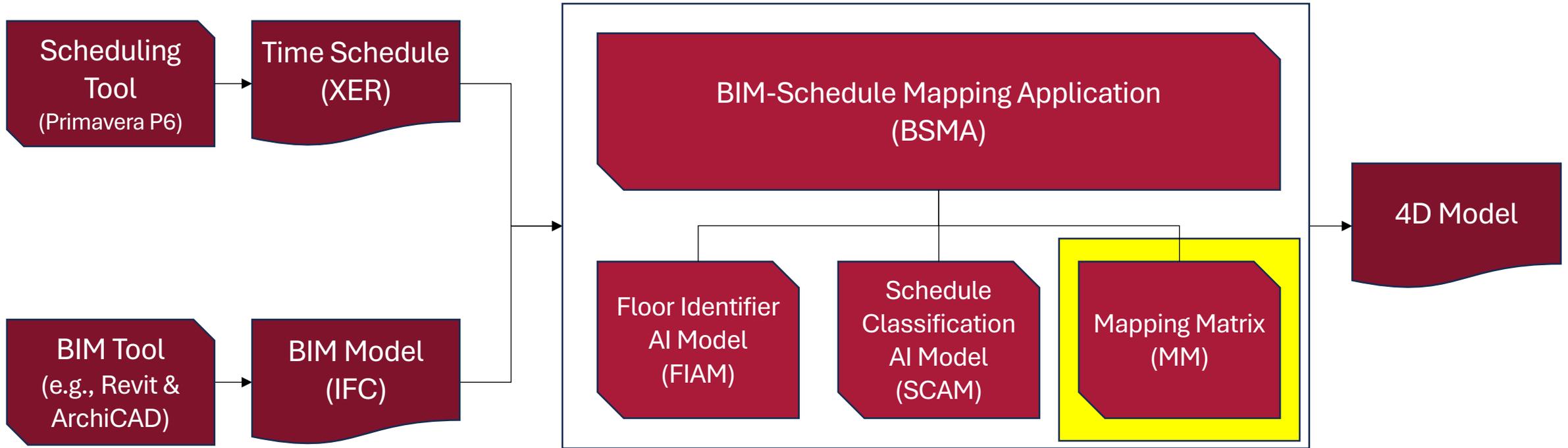
```
{"messages": [{"role": "user", "content": "Analyze the following WBS chain and identify the floor. If no floor is found, classify it as 'Generic'. WBS Chain: High-Rise > Floor 10 > Interior Walls > Drywall Installation"}, {"role": "assistant", "content": "Floor: Floor 10"}]}
```

```
{"messages": [{"role": "user", "content": "Analyze the following WBS chain and identify the floor. If no floor is found, classify it as 'Generic'. WBS Chain: Project X > Site Preparation > Excavation"}, {"role": "assistant", "content": "Floor: Generic"}]}
```

```
{"messages": [{"role": "user", "content": "Analyze the following WBS chain and identify the floor. If no floor is found, classify it as 'Generic'. WBS Chain: Tower B > Level 3 > Plumbing > Rough-in"}, {"role": "assistant", "content": "Floor: Level 3"}]}
```

Expression	Clarification
Messages	A list that simulates a conversation history
1. User Message	This is what the model must learn to process.
<ul style="list-style-type: none">role (User)	The input source
<ul style="list-style-type: none">content (User)	The instruction including WBS chain
2. Assistant Message	The response that the model is being trained to generate
<ul style="list-style-type: none">role (Assistant)	The desired model output
<ul style="list-style-type: none">content (Assistant)	The target output in a simple, standardized format (Floor: [Value]).

FRAMEWORK MODEL



Mapping Matrix (MM)

- In order to map the BIM elements to the standard building elements, a rule-based database has been created to identify which IFC elements will be mapped to classified activities.
- To create this matrix, the IFC documentation was explored, extracting all the entities that represent building elements (e.g., IfcSlab, IfcWall, etc.)
- However, building elements from different disciplines are represented using the same IFC entity in IFC3x2 (e.g., HVAC ducts, plumbing pipes, and electrical wires are all represented using IfcFlowSegment).
- Likewise, HVAC terminals, valves, sanitary devices, and electrical switches are all represented using IfcFlowTerminal

Mapping Matrix (MM)

- Due to this issue, the mapping matrix has been broken down into 2 sections.
- The first section represents the elements that can be directly mapped using the IFC entity.
- The second section represents the elements that require an additional criterion. This was the name of the element

Mapping Type	Class	IFC Element/ Keyword
Direct Mapping	Columns	IfcColumn
	Exterior_Walls	IfcWall
	Doors	IfcDoor
	Elevators	IfcTransportElement
Indirect Mapping	Foundations	footing
	Tile_Flooring	tile
	Plumbing_Third_Fix	sink drain water closet flush
	Electrical_First_Fix	conduit cable tray

Case Study Results

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Hardware accelerator e.g. GPU is available in the environment, but no `device` argument is passed to the `Pipeline` object. Model will be on CPU.
#####
#                                                                 #
#                                                                 #
#           Step 1 Completed. The trained BERT model has been loaded successfully!           #
#                                                                 #
#                                                                 #
#####
Press Enter to extract the activities from the XER file
```



< 0 0

Ln 14, Col 34

Case Study Results

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

(164139, 'Installation of window', datetime.datetime(2026, 5, 13, 8, 0), datetime.datetime(2026, 5, 24, 16, 0))
(164140, 'Protection Topping with Screed', datetime.datetime(2025, 12, 2, 8, 0), datetime.datetime(2025, 12, 25, 16, 0))
(164141, 'Installation of the water pipes', datetime.datetime(2026, 2, 23, 8, 0), datetime.datetime(2026, 3, 5, 16, 0))
(164142, 'Installation of the Sanitary fixtures', datetime.datetime(2026, 5, 13, 8, 0), datetime.datetime(2026, 5, 28, 16, 0))
(164143, 'Installation of AC Ductwork', datetime.datetime(2026, 2, 23, 8, 0), datetime.datetime(2026, 3, 10, 16, 0))
(164144, 'Erection of AC Grills and Diffusers', datetime.datetime(2026, 5, 13, 8, 0), datetime.datetime(2026, 5, 24, 16, 0))
(164145, 'Electrical Conduits and cable trays', datetime.datetime(2026, 1, 17, 8, 0), datetime.datetime(2026, 1, 27, 16, 0))
(164146, 'Pulling electrical wires', datetime.datetime(2026, 2, 23, 8, 0), datetime.datetime(2026, 3, 1, 16, 0))
(164147, 'installation of electrical lighting, outlets, and switches', datetime.datetime(2026, 5, 13, 8, 0), datetime.datetime(2026, 5, 23, 16, 0))
(164148, 'Testing & Commissioning of MEP systems', datetime.datetime(2026, 5, 30, 8, 0), datetime.datetime(2026, 6, 14, 16, 0))
(164149, 'Handover - Snagging and Issuance of Taking Over Certificate', datetime.datetime(2026, 6, 15, 8, 0), datetime.datetime(2026, 6, 22, 16, 0))
#####
#
#
#           Step 2 Completed. Activities have been extracted from the P6 File
#
#
#####
Press Enter to extract the IFC elements from the BIM Model
```

Case Study Results

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

('14ofPoRPz5xxGNigMC_Fpn', 'M_Footing-Rectangular:1800 x 1200 x 450mm:1672655', 'IfcSlab')
('14ofPoRPz5xxGNigMC_Fpl', 'M_Footing-Rectangular:1800 x 1200 x 450mm:1672657', 'IfcSlab')
('14ofPoRPz5xxGNigMC_Fpj', 'M_Footing-Rectangular:1800 x 1200 x 450mm:1672659', 'IfcSlab')
('1XHufJjtH6HwKSyuEjep9F', 'Foundation Slab:Conc 200mm:1676909', 'IfcSlab')
('1XHufJjtH6HwKSyuEjepk5', 'Foundation Slab:Conc 200mm:1679271', 'IfcSlab')
('09zK8e1nvFZh1FDGpPf50k', 'Foundation Slab:150mm Foundation Slab:1682279', 'IfcSlab')
('09zK8e1nvFZh1FDGpPf56V', 'Foundation Slab:150mm Foundation Slab:1682646', 'IfcSlab')
('09zK8e1nvFZh1FDGpPf56n', 'Foundation Slab:150mm Foundation Slab:1682680', 'IfcSlab')
('1EbXocQLX2HqKsI2S0Q9s0', 'Basic Roof:Generic - 225mm:1730936', 'IfcCovering')
('09zK8e1nvFZh1FDGpPf59C', 'Cast-In-Place Stair:Stair:1683205', 'IfcStair')
('1ntkqXKinCX023V6DL8lgK', 'Cast-In-Place Stair:Stair:1684783', 'IfcStair')
#####
#                                                                                                     #
#                                                                                                     #
#           Step 3 Completed. A total of 904 elements have been extracted from the BIM Model.           #
#                                                                                                     #
#                                                                                                     #
#####
Press Enter to start activity classification using the trained BERT Model
```



< 0 0

Case Study Results

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Task ID: 164139, Activity: Installation of window -> Class: Exterior_Windows
Task ID: 164140, Activity: Protection Topping with Screed -> Class: Roof_Decking
Task ID: 164139, Activity: Installation of window -> Class: Exterior_Windows
Task ID: 164139, Activity: Installation of window -> Class: Exterior_Windows
Task ID: 164140, Activity: Protection Topping with Screed -> Class: Roof_Decking
Task ID: 164141, Activity: Installation of the water pipes -> Class: Plumbing_Second_Fix
Task ID: 164142, Activity: Installation of the Sanitary fixtures -> Class: Plumbing_Third_Fix
Task ID: 164143, Activity: Installation of AC Ductwork -> Class: HVAC_Second_Fix
Task ID: 164144, Activity: Erection of AC Grills and Diffusers -> Class: HVAC_Third_Fix
Task ID: 164145, Activity: Electrical Conduits and cable trays -> Class: Electrical_First_Fix
Task ID: 164146, Activity: Pulling electrical wires -> Class: Electrical_Second_Fix
Task ID: 164147, Activity: installation of electrical lighting, outlets, and switches -> Class: Electrical_Third_Fix
Task ID: 164148, Activity: Testing & Commissioning of MEP systems -> Class: test
Task ID: 164149, Activity: Handover - Snagging and Issuance of Taking Over Certificate -> Class: no_mapping
#####
#                                                                 #
#                                                                 #
#      Step 4 Completed. Activities have been classified as per the trained BERT Model      #
#                                                                 #
#                                                                 #
#####
Press Enter to start the mapping process
```



Case Study Results

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

('0YYWYoM_f3A8096DxEEEbK', 'M_Downlight - Rectangle - LED:18W Square 30x30:1729130', 'IfcFlowTerminal')
('0YYWYoM_f3A8096DxEEEwa', 'M_Downlight - Rectangle - LED:18W Square 30x30:1729306', 'IfcFlowTerminal')
('0YYWYoM_f3A8096DxEEES$', 'M_Downlight - Rectangle - LED:18W Square 30x30:1729762', 'IfcFlowTerminal')
('0YYWYoM_f3A8096DxEEEvf', 'M_Downlight - Rectangle - LED:18W Square 30x30:1729879', 'IfcFlowTerminal')
('0YYWYoM_f3A8096DxEEESB', 'M_Downlight - Rectangle - LED:18W Square 30x30:1729781', 'IfcFlowTerminal')
=====
164148 Testing & Commissioning of MEP systems
Dates: 2026-05-30 08:00:00 - 2026-06-14 16:00:00
-----
- This activity ('Testing & Commissioning of MEP systems') was not mapped due to exclusion (Class: 'test').
=====
164149 Handover - Snagging and Issuance of Taking Over Certificate
Dates: 2026-06-15 08:00:00 - 2026-06-22 16:00:00
-----
- This activity ('Handover - Snagging and Issuance of Taking Over Certificate') was not mapped due to exclusion (Class: 'no_mapping').
#####
#                                                                                                     #
#                                                                                                     #
#           Step 5 Completed. Activity Mapping has been completed                                     #
#                                                                                                     #
#                                                                                                     #
#####
Press Enter to add the tasks to the IFC model
```



Case Study Results

The screenshot displays the Synchro software interface, titled "Untitled - Synchro". The interface is divided into several sections:

- Top Menu Bar:** File, Configure, Plan, Assign Resources, 3D, 4D Review, Analyse, Reports, Windows, Navigator, Help.
- Toolbar:** Contains various icons for project management, including "Above", "Indent", "Link As Chain", "On Screen", "Start", "Project Start", "Calendars", "Cut Task(s)", "Copy Task(s)", "Paste Task(s) As Text", "ID", "Reschedule", "Reschedule Selected", "Reschedule Report", "Compute Critical Path", "Settings", "Baseline Selected Tasks", "Baselines and Scenarios", "WBS", "WBS Layout", "Activity Codes", "Use Status Colours", "Show Spotlight", "Find and Replace...", "Select All", "Up to previous selected", and "Filters".
- 3D Objects Panel (Left):** Lists objects in the scene, including "Objects of scene" and "PN_type_c.ifc".
- Task List (Center):** A table with columns for ID, Name, and Duration. Below it is a Gantt chart showing task scheduling from October to January 2026.
- 3D View Properties Panel (Right):** A configuration panel for the 3D view, including sections for General, Visibility, Effects, Indicators, and Selection.
- 3D View (Bottom Center):** A 3D rendering of a building with a green roof and white walls, titled "3D Using Dates [Best] [1099x398]". A "FRONT" orientation indicator is visible.
- Status Bar (Bottom):** Displays system information such as "Never expire", "Enough memory", "[Filter Off] Selected[0][1][0]", "20:53 16/08/25", "Private Project", and "Transactions: 1".

Case Study Results

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

- Added task: 'Erection of AC Grills and Diffusers' (ID: 164144) and linked to 48 elements.
- Added task: 'Electrical Conduits and cable trays' (ID: 164145) and linked to 37 elements.
- Added task: 'Pulling electrical wires' (ID: 164146) but no elements found to link.
- Added task: 'installation of electrical lighting, outlets, and switches' (ID: 164147) and linked to 50 elements.
Skipping task 'Testing & Commissioning of MEP systems' (ID: 164148) due to exclusion (Class: 'test').
Skipping task 'Handover - Snagging and Issuance of Taking Over Certificate' (ID: 164149) due to exclusion (Class: 'no_mapping').

Tasks successfully added and saved to 'd:\Knowledge\Python\202504_workspace\Projects\PNC\PN_type_c_with_tasks.ifc'
#####
#                                                                 #
#                                                                 #
#           Step 6 Completed. Tasks have been added to the BIM Model.           #
#                                                                 #
#                                                                 #
#####

Activities added to IFC file but not linked to BIM elements:
- Pulling electrical wires
```



>< 0 0

Case Study Results

```
D:\Knowledge\Python\202504_Workspace\Projects\PNC\PN_type_c_with_tasks.ifc - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
PN_type_c_with_tasks.ifc
122502 #122676=IFCTASK('3RHJYsIsL9QPeBfPi0GnZ7',#18,'Formwork and reinforcement for the isolated footings','C
122503 #122677=IFCRELASSIGNSTASKS('3gSfvbFsLDJf0CIg_w_zqE',#18,$,$,(#122676),$,#27,#122675);
122504 #122678=IFCRELASSIGNSTOPROCESS('1uiWRDp3v2cB5gyeqDl5x1',#18,$,$,(#5937,#943,#111,#4286,#3157,#2818,#37
122505 #122679=IFCCALENDARDATE(20,9,2025);
122506 #122680=IFCLOCALTIME(8,0,0.,$, $);
122507 #122681=IFCDATEANDTIME(#122679,#122680);
122508 #122682=IFCCALENDARDATE(20,9,2025);
122509 #122683=IFCLOCALTIME(16,0,0.,$, $);
122510 #122684=IFCDATEANDTIME(#122682,#122683);
122511 #122685=IFCSCHEDULETIMECONTROL('2SbpqYRjXFb8nhBqRVu81X',#18,'Pouring the concrete for the foundation S
122512 #122686=IFCTASK('0$g7BDK0X6rAq2XsEDDQXp',#18,'Pouring the concrete for the foundation','Construction s
122513 #122687=IFCRELASSIGNSTASKS('3NpkxQ$obBeuvlcEAieuee',#18,$,$,(#122686),$,#27,#122685);
122514 #122688=IFCRELASSIGNSTOPROCESS('0rM$q_NMr3_gp4K09ejP92',#18,$,$,(#5937,#943,#111,#4286,#3157,#2818,#37
122515 #122689=IFCCALENDARDATE(21,9,2025);
122516 #122690=IFCLOCALTIME(8,0,0.,$, $);
122517 #122691=IFCDATEANDTIME(#122689,#122690);
122518 #122692=IFCCALENDARDATE(28,9,2025);
122519 #122693=IFCLOCALTIME(16,0,0.,$, $);
122520 #122694=IFCDATEANDTIME(#122692,#122693);
122521 #122695=IFCSCHEDULETIMECONTROL('ldQ5NTwhjCcPKW6fMwAMv4',#18,'Curing for the RC foundation Schedule Con
```

Case Study Results

Line #	IFC Entity	Clarification
122676	IFCTASK	Declaring the task (activity)
122677	IFCRELASSIGNSTASKS	Activity resources.
122678	IFCRELASSIGNSTOPROCESS	Connecting the BIM elements to the task.
122679	IFCCALENDARDATE	Start date.
122680	IFCLOCALTIME	Start time.
122681	IFCDATEANDTIME	Combining date and time.
122682	IFCCALENDARDATE	End date.
122683	IFCLOCALTIME	End time.
122684	IFCDATEANDTIME	Combining end date and end time.

Case Study Results

The screenshot displays the Synchro software interface for a construction project. The main window is titled 'Untitled - Synchro' and features a ribbon menu with tabs for File, Configure, Plan, Assign Resources, 3D, 4D Review, Analyse, Reports, Windows, and Navigator. The Plan tab is active, showing various task management tools like 'Cut Task(s)', 'Copy Task(s)', and 'Paste Task(s) As Text'. Below the ribbon, there are dropdown menus for 'Install', 'FS', 'Project Calendar', 'Resource', and 'Task'. The interface is divided into several panels:

- 3D Objects:** A tree view on the left showing the hierarchy of objects in the scene, including 'PN_type_c_with_tasks.ifc'.
- Task List:** A table in the center listing tasks with columns for ID, Name, Duration, and Start/End times. The tasks are:

ID	Name	Duration	Start	End
1	164... Formwork and reinforcement for the isol...	7d, 7h	09:0	
2	164... Pouring the concrete for the foundation	1d	09:0	
3	164... Curing for the RC foundation	5d	09:0	
4	164... Formwork and reinforcement for the Col...	4d, 7h	09:0	
5	164... Pouring the concrete for the Columns	1d	09:0	
6	164... Curing for the Columns	5d, 7h	09:0	
7	164... Formwork and reinforcement for the slab	14d, 7h	09:0	
8	164... Curing for the slab	5d, 7h	09:0	
9	164... Implementation of Brickwork for the walls	13d, 7h	09:0	
10	164... Door Frames	6d, 7h	09:0	
- 3D View Properties:** A panel on the right with settings for 'General', 'Visibility', 'Effects', 'Indicators', and 'Selection'. The '3D Window Title' is '3D', and the 'Render mode' is 'Gouraud Shaded'.
- 3D Model:** A 3D rendering of a building with a green roof and white walls, viewed from a perspective. A 'FRONT' orientation indicator is visible.
- Gantt Chart:** A Gantt chart at the bottom right shows the project schedule from October to January 2026, with tasks represented by colored bars.

The status bar at the bottom indicates system information: 'For Help, press F1', 'Never expire', 'Enough memory', '[Filter Off] Selected[26][1][0]', '16:00 28/05/26', 'Private Project', 'Transactions: 1', and the system clock 'ENG 10:06 PM'.

REFLECTIONS AND IMPLICATIONS

LESSONS LEARNED FROM THIS CASE STUDY

1. Inference Vs Training

Interacting with the LLM (Inference) does not modify its core competence. True domain adaptation requires Fine-Tuning—a dedicated process utilizing specialized datasets, training techniques, and tools like the HuggingFace API to update the neural network's weights.

LESSONS LEARNED FROM THIS CASE STUDY

2. Data Preparation

The success of fine-tuning relies heavily on the quality and volume of the domain-specific training data. A key reflection is the significant time and effort required to clean the schedules to achieve high accuracy in activity-to-model mapping.

LESSONS LEARNED FROM THIS CASE STUDY

3. Fine-Tuning and Knowledge Transfer

The study successfully validated the power of Transfer Learning, confirming that we don't need to train a model from scratch, which would take massive time and computing. We simply benefit from the model's existing understanding of language and adapt it to our field, Project Controls.

FINE-TUNING OPPORTUNITIES IN PROJECT CONTROL

Topic	Fine-Tuning Opportunity	Application
Claims Management	Training on the project communications and documents of previous projects where claims were initiated, to understand the relation between the claims and their context.	Support avoiding claims in future projects.
Risk Management	Training on the meeting minutes, correspondences, communications, and documents of previous projects where issues occurred, to understand the relation between these issues and their context.	Constantly scan communication to predict potential risks and automatically assign them to the risk register with a suggested mitigation action.
Cost Estimation	Training on BOQs, project specifications, and historical cost data to understand the relation between project scope and the line items of BOQ.	Quickly scan the project documents to find the inconsistencies or missing scope.

FINE-TUNING OPPORTUNITIES IN PROJECT CONTROL

Topic	Fine-Tuning Opportunity	Application
Scheduling	Training on project schedules of previous projects (including the XER files, narratives, productivity sheets, etc.) and the consultant comments after reviewing the schedule, to understand the relation between the schedule and the comments.	Support the review and assess the time schedules submitted by the Contractor for review.
Delay Analysis	Training on schedule baselines, schedule updates, project communications, periodical reports, and delay analysis reports, to understand the relation between the root causes of the delay and the project status.	Help predict the root causes of delay events and analyze them and their impact.
Technical Proposals	Training on the bidding documents (Request for Proposals, Invitation to Bid, Terms of Reference, etc.) and the prepared technical proposal, including any disclaimers or qualifications, to understand their relation.	Support in preparing accurate technical proposal, highlighting the potential liabilities.



Mohamed ElSaadany

Senior Manager, PMC | Khatib & Alami



ilshawy@hotmail.com

mohamed.saadany@khatibalami.com



[linkedin.com/in/msaadany](https://www.linkedin.com/in/msaadany)



[youtube.com/@msaadanytube](https://www.youtube.com/@msaadanytube)



+966 54 55 483 77



**Thank
You**