Fine-Tuning Objective for SAPCAD LLM

Context

The SAPCAD project involves developing an AI assistant that understands user instructions (in English or German) and BIM standards to generate structured JSON outputs for modifying IFC (Industry Foundation Classes) building models using IfcOpenShell.

To bridge the gap between natural language and BIM element manipulation, we aim to fine-tune a large language model (LLM) to specialize in architectural tasks.

Objective

The goal is to fine-tune the LLM to:

1. Understand design instructions in natural language (English/German)

2. Interpret building standard constraints when available (e.g., GEG, passive housing regulations)

3. Identify relevant IFC elements (e.g., IfcWindow, IfcWall, IfcRoof)

4. Generate structured JSON output to:

- Modify existing elements

- Remove elements

- Add new elements

This JSON is directly parsed and applied using IfcOpenShell.

Example Output Schema

```
{
  "element_updates": [
    {
      "type": "<IfcElement>",
      "action": "modify" | "remove" | "add",
      "target_properties": {
            "<property_name>": <value or "auto" or null>
        },
      "unit": "<unit_from_spec>"
    }
  ],
  "update_scope": "<describe_scope>"
}
```

Dataset Design

To train the model effectively, we will prepare a dataset with examples that include:

Fields per training sample (in JSONL format):

```
{
  "instruction": "Make the window next to the entrance door wider.",
  "bim_spec": {
    "IfcWindow": {
      "editable_properties": ["OverallWidth", "OverallHeight"],
      "unit": "mm"
    }
  },
  "output": {
    "element_updates": [
      {
        "type": "IfcWindow",
        "action": "modify",
        "target_properties": {
          "OverallWidth": "auto"
        },
        "unit": "mm"
      }
    ],
    "update_scope": "window near entrance door"
  }
}
```

Required dataset coverage:

- Modify instructions: "Change color", "Resize wall"

- Add elements: "Add a window on south wall"

- Remove elements: "Remove roof above kitchen"

- Location-aware: "Next to the main entrance"

- BIM-rule aware: "Comply with U-values < 0.15"

Output Usage

The model's output will be used by a backend FastAPI route called modify_ifc_endpoint, which applies changes to the IFC file. Based on the action field, it routes to the appropriate handler:

- modify → _modify_property()

- remove → delete_element()

- add → add_ifc_element()

Status

- Prompt-response samples reviewed

- Project tested using new prompt structure

- Task "Enable Add and Remove IFC Elements via IfcOpenShell" created in Kanban

- Task "Update modify_ifc_endpoint to support add/remove" defined and underway