Documentation - Take home assignment - Gowshalini

I have chosen Assignment 2 as it aligns with my expertise

Methodology

1. I have Initially gone through the dataset and found some useful information from below links.

   I have downloaded the FHIR JSON samples from the below URL:
   https://build.fhir.org/downloads.html

   Resources
   https://hl7.org/fhir/ - Home page
   https://hl7.org/fhir/documentation.html - All documentation
   https://hl7.org/fhir/json.html - Detail description for JSON format
   https://build.fhir.org/downloads.html - Download examples

2. Experiment phase
   - Initially, I converted the provided pdf "Discharge_Summary_John_Doe_new.pdf into text format using the fitz library and checked its performance of it. It has extracted the text well. (fitz is better for simple pdf)

```
import fitz  # PyMuPDF for PDF text extraction

# Step 1: Extract Text from PDF
def extract_text_from_pdf(pdf_path):
    """Extracts text from a PDF file."""
    doc = fitz.open(pdf_path)
    text = "\n".join([page.get_text("text") for page in doc])
    return text
```

   - Then Developed GeminiAI LLM and via prompt engineering tried to achieve the results. As a response, I have gotten decent results.

   Code:

```
# Step 2: Configure Google Gemini API
def setup_gemini(api_key):
   """Configures the Google Gemini API."""
   genai.configure(api_key=api_key)
```

```python
# Step 3: Send Text to Google Gemini for FHIR Conversion
def generate_fhir_bundle(prompt):
    """Sends extracted text to Google Gemini AI for FHIR Bundle JSON conversion."""
    model = genai.GenerativeModel("gemini-1.5-pro")  # Using the latest Gemini model
    response = model.generate_content(prompt)
    print(response.text)
    return response.text  # Extract generated JSON

# Step 4: Save JSON to a File
def save_json(output_json, filename="fhir_bundle.json"):
    """Saves the JSON output to a file."""
    with open(filename, "w", encoding="utf-8") as f:
        f.write(output_json)
    print(f"FHIR Bundle saved as {filename}")


if __name__ == "__main__":
    # 🔹 Set your Google API key here

    pdf_path = "Discharge_Summary_John_Doe_new.pdf"
    setup_gemini(GOOGLE_API_KEY)


    extracted_text = extract_text_from_pdf(pdf_path)

    prompt = f"""
    Convert the following unstructured medical discharge summary into a structured FHIR Bundle JSON.
    Ensure the FHIR resources include:
    - Patient
    - Condition (Primary and Secondary Diagnoses)
    - Procedure (Procedures performed)
    - Observation (Vital signs and Lab results)
    - MedicationStatement (Medications prescribed)

    Discharge Summary:
    {extracted_text}

    Output the FHIR Bundle JSON:
    """
```

```
fhir_bundle_json = generate_fhir_bundle(prompt)
save_json(fhir_bundle_json)
```

Results:

```
{
"resourceType": "Bundle",
"type": "document",
"identifier": {
  "system": "urn:uuid",
  "value": "discharge-summary-123456789"
},
"timestamp": "2025-02-08T12:00:00Z",
"entry": [
  {
    "fullUrl": "urn:uuid:patient-123456789",
    "resource": {
      "resourceType": "Patient",
      "id": "patient-123456789",
      "identifier": [
        {
          "system": "http://hospital.example.org/mrn",
          "value": "123456789"
        }
      ],
      "name": [
        {
          "use": "official",
          "family": "Doe",
          "given": ["John"]
        }
      ],
      "gender": "male",
      "birthDate": "1985-03-15"
    }
  },
  {
    "resource": {
      "resourceType": "Condition",
      "id": "condition-primary",
      "subject": {
        "reference": "urn:uuid:patient-123456789"
      },
      "code": {
        "coding": [
          {
            "system": "http://snomed.info/sct",
          "code": "410429000",
          "display": "Acute myocardial infarction"
        }
      ],
      "text": "Acute Myocardial Infarction (ST-Elevation)"
    },
    "clinicalStatus": {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
          "code": "active",
          "display": "Active"
        }
      ]
    }

  }
},
{
  "resource": {
    "resourceType": "Condition",
    "id": "condition-secondary1",
    "subject": {
      "reference": "urn:uuid:patient-123456789"
    },
    "code": {
      "coding": [
        {
          "system": "http://snomed.info/sct",
          "code": "38341003",
          "display": "Hypertension"
        }
      ]
    },
     "clinicalStatus": {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
          "code": "active",
          "display": "Active"
        }
      ]
    }
  }
},
{
  "resource": {
    "resourceType": "Condition",
      "id": "condition-secondary2",
      "subject": {
        "reference": "urn:uuid:patient-123456789"
      },
      "code": {
        "coding": [
          {
            "system": "http://snomed.info/sct",
            "code": "44054006",
            "display": "Type 2 diabetes mellitus"
          }
        ]
      },
       "clinicalStatus": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
            "code": "active",
            "display": "Active"
          }
        ]
      }
    }
  },
      {
    "resource": {
      "resourceType": "Condition",
      "id": "condition-secondary3",
      "subject": {
        "reference": "urn:uuid:patient-123456789"
      },
      "code": {
        "coding": [
          {
            "system": "http://snomed.info/sct",
            "code": "55822004",
            "display": "Hyperlipidemia"
          }
        ]
      },
       "clinicalStatus": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
            "code": "active",
            "display": "Active"
          }
        ]
    }
  }
},
{
   "resource": {
    "resourceType": "Condition",
    "id": "condition-secondary4",
    "subject": {
      "reference": "urn:uuid:patient-123456789"
    },
    "code": {
      "coding": [
        {
          "system": "http://snomed.info/sct",
            "code": "414916001",
            "display": "Obesity"
        }
      ]
    },
     "clinicalStatus": {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
          "code": "active",
          "display": "Active"
        }
      ]
    }
  }
},

 {
  "resource": {
    "resourceType": "Procedure",
    "id": "procedure-angiography",
    "subject": {
      "reference": "urn:uuid:patient-123456789"
    },
    "code": {
      "coding": [
        {
          "system": "http://snomed.info/sct",
          "code": "72863005",
          "display": "Coronary angiography"
        }
      ],
      "text": "Coronary Angiography"
    },
        "performedDateTime": "2025-02-02"
      }
    },
    {
      "resource": {
        "resourceType": "Procedure",
        "id": "procedure-pci",
        "subject": {
          "reference": "urn:uuid:patient-123456789"
        },
        "code": {
          "coding": [
            {
               "system": "http://snomed.info/sct",
              "code": "225363005",
              "display": "Percutaneous transluminal coronary angioplasty"
            }
          ],
          "text": "Percutaneous Coronary Intervention (PCI) with Drug-Eluting
Stent Placement"
        },
        "performedDateTime": "2025-02-03"
      }
    },
    {
      "resource": {
        "resourceType": "Observation",
        "id": "observation-bp-admission",
        "subject": {
          "reference": "urn:uuid:patient-123456789"
        },
        "code": {
          "coding": [
            {
              "system": "http://loinc.org",
              "code": "85354-9",
              "display": "Blood Pressure"
            }
          ]
        },
        "valueQuantity": {
          "value": 160,
          "unit": "mmHg",
          "system": "http://unitsofmeasure.org",
          "code": "mm[Hg]"
        },
        "component": [
          {
```

```
        "code": {
          "coding": [
             {
              "system": "http://loinc.org",
              "code": "8480-6",
              "display": "Diastolic Blood Pressure"
            }
          ]
        },
        "valueQuantity": {
          "value": 95,
          "unit": "mmHg",
          "system": "http://unitsofmeasure.org",
          "code": "mm[Hg]"
        }
      }
    ],
     "effectiveDateTime": "2025-02-01"
  }
},

{
  "resource": {
    "resourceType": "Observation",
    "id": "observation-hr-admission",
    "subject": {
      "reference": "urn:uuid:patient-123456789"
    },
    "code": {
       "coding": [
        {
          "system": "http://loinc.org",
          "code": "8867-4",
          "display": "Heart rate"
        }
      ]
    },
    "valueQuantity": {
      "value": 102,
      "unit": "/min",
      "system": "http://unitsofmeasure.org",
      "code": "/min"
    },
      "effectiveDateTime": "2025-02-01"
  }
},
{
    "resource": {
      "resourceType": "MedicationStatement",
      "id": "medication-aspirin",
      "medicationCodeableConcept": {
        "coding": [
          {
            "system": "http://www.nlm.nih.gov/research/umls/rxnorm",
            "code": "1191",
            "display": "Aspirin"
          }
        ]
      },
      "subject": {
        "reference": "urn:uuid:patient-123456789"
      },
      "dosage": [
        {
          "text": "81 mg daily",
          "doseAndRate": [
            {
              "doseQuantity": {
                "value": 81,
                "unit": "mg",
                "system": "http://unitsofmeasure.org",
                "code": "mg"
              }
            }
          ]
        }
      ],
        "status": "active"
    }
  }
  // ... (Rest of the Observation and MedicationStatement resources)
 ]
}
```

Evaluation

How to evaluate it?

1. Find publicly available datasets with pdf and corresponding FHIR Bundles. Then compare the generated output with the actual output for each pdf and calculate the accuracy
    a. I was searching for this, I couldn’t find dataset.
2. Then I found the JSON example datasets from below link and downloaded it.

https://build.fhir.org/downloads.html
3. From the downloaded dataset I filtered the bundles related json files.
4. Then using chat-gpt I generated the pdfs
5. Then I tried to generate the FHIR bundles for these pdf and compare the actual vs generated output.

   For the comparison I again used LLM to compare. Because the depending on python function based on rules didn't fit for all scenarios.

   Code:

```
# Function to load JSON files
def load_json(file_path):
     """Loads a JSON file."""
     with open(file_path, "r", encoding="utf-8") as file:
         return json.load(file)

# Main Execution
if __name__ == "__main__":
   # 🔹 Set your Google API key here

   pdf_path = "/content/drive/MyDrive/canada company
assignment/bundle/bundle-lipids.pdf"
   setup_gemini(GOOGLE_API_KEY)

   json_1 = load_json("/content/drive/MyDrive/canada company
assignment/bundle/bundle-lipids.json")
   json_2 = load_json("/content/drive/MyDrive/canada company
assignment/bundle/bundle-lipids_output.json")


   prompt = f"""
   You are an expert in JSON structure analysis. Your task is to
compare two JSON documents and determine how similar they are.

### **Instructions:**
1. **Compare both JSONs field by field**, considering:
  - Missing or extra fields.
  - Differences in values (including numbers, strings, and
lists).
  - Structural differences (nested objects, ordering).

2. **Generate a similarity score (0% to 100%)** based on:
  - **Matching fields and values**: Higher score.
  - **Minor format differences (case sensitivity, spacing, 
ordering)**: Ignore if the meaning is unchanged.
  - **Significant differences in keys or values**: Lower score.

3. **Provide an explanation** of what is different:
  - List fields that are **identical**.
  - Highlight fields that have **small differences** (e.g., 
`"value": "120"` vs `"value": "125"`).
  - Show **missing fields** in one JSON but present in the other.
  - Highlight incorrect or extra information.

### **Example Output:**
✅ **Similarity Score: 92%**
🔹 **Identical Fields:** 45/50 match
⚠️ **Minor Differences:** `patient.name = "John Doe" vs "Doe, 
John"`
❌ **Missing Fields:** `"followUpDate"` is missing in the 
generated JSON.

Now, compare the following two JSONs and provide a similarity 
score along with a summary of differences.

### **JSON 1:**
{json_1}

### **JSON 2:**
{json_2}

   """

   response = generate_fhir_bundle(prompt)
   print(response)
```

Results

1. The results gave me 25% accuracy overall. But this doesn't mean the results generation is poor in performance. To actually improve the evaluation criteria I have to study each parameter and what is needed and what is not. Then I can improve the prompt to generate the FHIR bundles and also the evaluation criteria. If I get help from the people who are experts in this dataset may be helpful. If not I may need more time to study and experiment with the dataset.

Improvements

1. I have tried to build a RAG application where I can pass the dataset that contains the representations of FHIR JSON. Details can be found from below link.
   https://hl7.org/fhir/json.html

   I have created a text file based on this and passed it to the LLM prompt. But it reduced the accuracy. Need more time to tune the prompt.

   The text file I have created is named: FHIR_JSON_Representation.txt

   Please note that to calculate the accuracy I took only 5 files from “bundle” folder. There are 14 files where we can create the pdf manually or using convertors if available and use it to evaluate.

   Still, I used ChatGPT to generate the pdf from JSON files in the dataset. So it may also not be 100% accurate. So need to find another accurate way.

Future work

1. Study the dataset well and get more idea of the FHIR JSON
2. Find a way to prepare the dataset for evaluation
3. Do research on the ways of evaluating this problem
4. If we get more data try training a new LLM model for this problem and do the inference.
5. Try self-supervised learning based approach to build a model. (knowledge distillation)
6. Create an interface to upload pdf and get generated output.