TRIOMICS

Objective

You have been provided with a set of **Electronic Health Record (EHR) notes** for patients diagnosed with various types of cancer. Your goal is to:

1. **Retrieve relevant chunks** of text from these EHR documents based on a user query (e.g., “Has the patient undergone chemotherapy?”).
2. **Extract structured medical data** focusing on:
    a. The **cancer diagnosis** (type, date, histology, stage).
    b. The **cancer-related medications** prescribed to the patient (drug name, start/end dates, intent).

This assignment tests your ability to:

1. Handle unstructured text.
2. Implement or leverage **Large Language Models (LLMs)** for **information retrieval** and **medical data extraction**.
3. Produce well-structured **JSON** output summarizing the patient’s key oncology data.

We understand that the tasks may be broad. **Submit whatever you can**—we will evaluate based on completeness, code quality, and clarity of thought.

Dataset

The dataset (in JSON format) includes multiple EHR notes per patient. Each note is an object with the following structure:

```python
patient_data = [
  {
    "docDate": "MM-DD-YYYY", # The date of the document in MM-DD-YYYY format.
    "docTitle": "Some Title", # A short title describing the type of note (e.g., “Pathology Report,” “Progress Note,” “Chemotherapy Cycle #2”).
    "docText": "Text content of the EHR Note", # The full text of the EHR note, typically 400–500 words, containing relevant clinical information.
  },
  # ... more documents for the same patient ...
]
```

Task 1: Information Retrieval

Prepare a pipeline to extract relevant information chunks/sentences from the EHR data for a given query. An approach could involve first breaking down the data into smaller semantically meaningful chunks, then using a text-embedding model to calculate similarity between the query and chunks from EHR notes or re-ranking models to extract relevant chunks. Example:

Input Query/Question: *"Has the patient undergone chemotherapy?"*
Output Retrieved Chunks:

1. *"On her follow-up on 02/15/2022 post her first cycle of chemotherapy, her response to Doxorubicin and Cyclophosphamide was evaluated."*
2. *"A follow-up mammogram conducted on 02/15/2022 showed reduced tumor size in the left breast, indicating a positive response to chemotherapy."*
3. *"Chemotherapy Cycle 3 Follow-up (05/05/2022):*
   - Assessing response to Docetaxel and Trastuzumab
   - Vital Signs: Stable
   - The patient reported manageable side effects, including mild fatigue and neuropathy."

Note: You can use any approach you wish (keyword matching, semantic embeddings, a specialized retrieval system, etc.)

Task 2: Medical Data Extraction

In this task, you will build a **Large Language Model (LLM)-based pipeline** to extract **key medical information** from a patient's EHR notes. Specifically, we are focusing on data relevant to **cancer diagnoses** and **cancer-related medications**. The final pipeline should return a structured **JSON output** for each patient.

Why Is This Important?
Clinical notes (EHRs) often contain critical information scattered across multiple documents, dates, or paragraphs. Healthcare providers and researchers need this information in a **structured** format (e.g., JSON) to:

1. Quickly **review** a patient's cancer diagnosis details and staging.
2. Identify what **treatments/medications** were given, when they started, when they ended, and why they were used.
3. Incorporate these structured data points into **analytical or clinical decision support systems** (e.g., for clinical trial matching, treatment monitoring, etc.).

Task 2.1: Cancer Diagnosis Characteristics

The first part focuses on extracting **diagnosis details** from the EHR. These details help establish the **type** and **stage** of the cancer.

Data To Extract:

1. **Primary Cancer Condition:** Example “Breast Cancer”, “Lung Cancer”, etc
2. **Diagnosis Date:** Earliest date on which the cancer got confirmed
3. **Histology:** Histological classification of the primary cancer condition
4. **Stage:** TNM and group stage of the cancer diagnosis

Detailed Definitions:

1. **Primary Cancer Condition:**
    a. This is the type of cancer the patient has, often listed in diagnoses as “Breast Cancer,” “Lung Adenocarcinoma,” “Prostate Adenocarcinoma,” etc.
    b. More resources: https://www.cancer.gov/types/common-cancers
2. **Diagnosis Date:**
    a. This is the earliest date on which a definitive diagnosis is mentioned.
    b. How to Find: Typically in sentences such as “The biopsy on 01/12/2020 confirmed invasive ductal carcinoma.” or “Pathology Report (02/17/2020): Invasive breast cancer.”
    c. You may see multiple references to diagnosis across notes; pick the earliest one that specifically confirms the cancer.
3. **Histology:**
    a. Describes the microscopic subtype of the tumor. Common examples: “Adenocarcinoma,” “Invasive ductal carcinoma,” “Squamous cell carcinoma,” etc.
    b. How to Find: In pathology reports or biopsy results. Terms like “Histologically consistent with adenocarcinoma” or “Invasive ductal carcinoma, Grade 2.”
4. **Stage:**
    a. T: Indicates Tumor size/extent. E.g., T2 means a moderate-sized tumor, T4 might mean a larger or invasive tumor.
    b. N: Indicates lymph Nodes involvement. N0 means no nodal involvement, N1/N2 means progressively more nodes involved.
    c. M: Indicates Metastasis. M0 means no distant spread; M1 means present.
    d. Group Stage: A single label (Stage I, Stage IIB, Stage IV, etc.) summarizing T, N, and M combined.
    e. How to Find: In imaging reports, pathology final reports, or physician notes, e.g. “Stage IIB (T2 N1 M0).” or “pT2 N1 M0.”
    f. More resources: https://www.cancer.gov/about-cancer/diagnosis-staging/staging

Task 2.2: Cancer Related Medications

In the second part, you will extract details about **medications** given specifically for the patient’s cancer treatment (as opposed to general meds like antihypertensives).

Data To Extract:

1. **Medication Name:** For example, “Doxorubicin,” “Cyclophosphamide,” “Paclitaxel,” “Trastuzumab,” “Pembrolizumab,” “Letrozole,” etc.
2. **Start Date:** The earliest date this medication was started, in MM-DD-YYYY format, if available.
3. **End Date:** The date the medication was stopped, if mentioned. If the patient is still on the medication, you may leave it blank or mark as null.

4. **Intent:** A free-text field describing why the medication was given. Examples: “Adjuvant therapy post-surgery,” “Neoadjuvant therapy to shrink tumor,” “Maintenance therapy for HER2+ disease,” or “Hormonal therapy to block estrogen in ER+ cancer.”

Expected Output Format

Your final pipeline should return a dictionary (or JSON) with two main keys:

1. **“diagnosis_characteristics”**: A **list** of dictionaries, each describing a primary cancer diagnosis. This can be multiple if a patient has more than one primary cancer (e.g., breast and thyroid).
2. **“cancer_related_medications”**: A **list** of dictionaries, each describing a medication given for the cancer.

```python
output = {
    "diagnosis_characteristics": [{
        "primary_cancer_condition": str,
        "diagnosis_date": str, # In MM-DD-YYYY format
        "histology": [str], # List of histology
        "stage": {
            "T": str, # T Stage,
            "N": str, # N Stage
            "M": str, # M Stage,
            "group_stage": str, # Group Stage,
        }
    }], # List of primary cancers
    "cancer_related_medications": [{
        "medication_name": str,
        "start_date": str, # In MM-DD-YYYY format
        "end_date": str, # In MM-DD-YYYY format
        "intent": str, # Intent/Reason for giving the corresposing medication
    }], # List of cancer related medication given to the patient
}
```

Hints & Modeling

1. You can use the pipeline from Task 1 to locate relevant passages about diagnosis, staging, or medication usage before extracting fields.
2. For an LLM-based approach, consider prompting the model with specific instructions to parse the text and fill in the required JSON fields.
3. For partial or missing data (e.g., no end date for a medication), it’s acceptable to leave those fields empty.

Example LLM Setup (Qwen1.5-7B-Chat with 4-bit Quantization)

You can use the Qwen1.5-7B-Chat model in Google Colab with 4-Bit quantization. We've provided a code snippet to load this model in 4-Bit.

Python

```
!pip install -q bitsandbytes accelerate optimum
```

```
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig

device = "cuda" # the device to load the model onto
quantization_config = BitsAndBytesConfig(
   load_in_4bit=True,
   bnb_4bit_quant_type="nf4",
   bnb_4bit_compute_dtype=torch.float16,
)

model = AutoModelForCausalLM.from_pretrained("Qwen/Qwen1.5-7B-Chat",
   use_safetensors=True,
   low_cpu_mem_usage=True,
   quantization_config=quantization_config,
   device_map=device
)

tokenizer = AutoTokenizer.from_pretrained("Qwen/Qwen1.5-7B-Chat")
```

Submission

Please submit your work either as a **Jupyter notebook file or as a link to Google Colab**. Ensure that each task within the notebook is clearly delineated and separated. Remember to include your essential observations by using comments or markdown cells.

Evaluation

We will evaluate solutions based on:

- **Data Understanding**: How clearly you handle EHR text, potential ambiguities, or missing data.

- **Innovation & Accuracy**: The techniques used for retrieval and extraction; correctness of extracted fields.
- **Code Quality**: Organization, readability, and documentation of your code. Well-commented notebook cells or markdown cells clarifying your approach.
- **Partial Work**: Even if you cannot fully complete both tasks, a strong partial solution can still score well.

Best of luck! We look forward to seeing your work.