#### **Assignment: Research Paper Analysis & Classification Pipeline**

----------------------------------------------------------------------------------------------------------------------------

### **Objective**

Your task is to fine tune an open-source small language model or quantized large language model to extract disease data and classify reaserch article abstracts into cancer and non-cancer categories.

Your goal is to:

1. **Use model** such as Gemma, Phi, etc. and fine-tune it using methods such as LoRA for cancer and non-cancer classification.
2. Use provided set of labelled abstracts. For each abstract identify mentioned diseases e.g. lung cancer, breast cancer etc.
3. Provide classification task performance of baseline and finetuned models in the form of confusion matrix and assess if finetuning improved performance.

### **Advanced Requirements**

1. **Model Selection & Justification**  
   1. Choose a pre-trained **multi-label text classification model** and justify your selection.
   2. Fine-tune a research paper dataset if necessary.
2. Compare different models' performance.
3. **Dataset Usage**

* The dataset will be provided as an **Excel file** containing a list of **PubMed IDs along with the abstracts.**Implement data pre-processing:
  + - Remove redundant metadata.
    - Normalize citations.
    - Handle missing abstracts.

1. **Programming & Libraries**
   1. Use **Hugging Face Transformers, PyTorch, TensorFlow** for model interaction.
   2. Use **NumPy, Pandas, Scikit-learn** for data handling.
   3. Implement **LangChain for structured prompt-based querying**.
2. **Model Output Structuring**
   1. Organize and structure model-generated outputs efficiently.
   2. Use an LLM for citation analysis.

### **Expected Output**

For each research paper abstract, the pipeline should return:

1. **1. Multi-Label Classification Output**

* **Predicted Categories:**

{

"predicted\_labels": ["Cancer", "Non-Cancer"]

}

* **Confidence Scores for Each Category:**

{  
 "Cancer": 0.92,  
 "Non-Cancer": 0.85  
}

### **2. Disease-Specific Identification from Abstract**

For each abstract, the model should extract specific diseases mentioned.

**Example Output:**

{

"abstract\_id": "12345",

"extracted\_diseases": ["Lung Cancer", "Breast Cancer"]

}

### **3. Model Performance Evaluation**

* **Baseline Model Performance:**
  + Accuracy: **85%**
  + F1-score: **0.78**
  + Confusion Matrix:

|  |  |  |
| --- | --- | --- |
|  | **Predicted Cancer** | **Predicted Non-Cancer** |
| **Actual Cancer** | 320 | 80 |
| **Actual Non-Cancer** | 50 | 550 |

* **Fine-Tuned Model Performance:**
  + Accuracy: **92%**
  + F1-score: **0.86**
  + Confusion Matrix:

|  |  |  |
| --- | --- | --- |
|  | **Predicted Cancer** | **Predicted Non-Cancer** |
| **Actual Cancer** | 350 | 50 |
| **Actual Non-Cancer** | 30 | 570 |

**Performance Improvement Analysis:**

* **Accuracy increased by 7%** after fine-tuning.
* **Reduction in false negatives, improving model reliability.**
* **Fine-tuned model provides better classification confidence.**

### **Bonus**

### **Agentic Workflow and Orchestration**

* **Can this pipeline be orchestrated as an agentic workflow solution?**

**Cloud Deployment**

* Deploy the pipeline as a REST API using **FastAPI** or **Flask**.
* Host on **AWS Lambda, Google Cloud Run, or Hugging Face Spaces**.
* **Containerize using Docker** and include a deployment script.

**Scalability Enhancements**

* Implement **batch processing** for multiple papers.
* Add **streaming capabilities** using **Apache Kafka or Redis Streams**.

### **Evaluation Criteria**

|  |  |
| --- | --- |
| **Criteria** | **Description** |
| **Code Quality** | Modular, well-structured, follows best practices. |
| **Functionality** | Multi-label classification, confidence scoring, topic extraction. |
| **Model Performance** | Accuracy, F1-score, and justification of model selection. |
| **Insight Extraction** | Relevance and correctness of topic breakdown, citations, and summaries. |
| **Scalability & Deployment (Bonus)** | API hosting, cloud integration, batch processing. |

### **Timeline**

**One Week** to complete the assignment.

Submit your work as a **GitHub repository link** or a **zip file** with clear instructions.