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 research 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

- **Model Selection & Justification**
  - Choose a pre-trained multi-label text classification model and justify your selection.
  - Fine-tune a research paper dataset if necessary.
- Compare different models' performance.
- **Dataset Usage**
  - The dataset will be provided, containing a list of PubMed IDs along with the abstracts. Implement data pre-processing:
    - Remove redundant metadata.
    - Normalize citations.
    - Handle missing abstracts.
- **Programming & Libraries**
  - Use Hugging Face Transformers, PyTorch, TensorFlow for model interaction.
  - Use NumPy, Pandas, Scikit-learn for data handling.
  - Implement LangChain for structured prompt-based querying.

- **Model Output Structuring**
  - Organize and structure model-generated outputs efficiently.
  - Use an LLM for citation analysis.

Expected Output

For each research paper abstract, the pipeline should return:

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 (README file).