

Research Paper Analysis & Classification Pipeline

1. Tools & Libraries Used

- Hugging Face Transformers
- PyTorch
- Pandas, NumPy, Scikit-learn
- LangChain for structured prompts

2. Repository Contents

- preparing_dataset_csv.py: Script to convert raw abstract files to CSV
- fine_tuning_data.csv: Output CSV for training
- Question_1_and_3_Research_Paper_Analysis_&_Classification_Pipeline_Velsera.ipynb: Fine-tuning notebook
- Disease_Specific_Identification_from_Abstacts.ipynb: Disease NER extraction notebook

 Project Structure

```
|—— dataset/
|   |—— cancer/          # Text files containing cancer-related abstracts
|   |—— non_cancer/      # Text files containing non-cancer abstracts
|—— preparing_dataset_csv.py # Script to preprocess text files and generate CSV
|—— fine_tuning_data.csv    # Generated dataset CSV for classification
|—— Question_1_and_3_Research_Paper_Analysis_&_Classification_Pipeline_Velsera.ipynb
|                           # Notebook for classification model training & evaluation
|—— Disease_Specific_Identification_from_Abstacts # Notebook for disease name
extraction
|—— README.md              # Project documentation
```

1. Dataset Preparation

Input Format: Raw .txt files in two folders: cancer and non_cancer, each with id, title, and abstract.

Processing:

Combined into a single CSV using preparing_dataset_csv.py

Output CSV: fine_tuning_data.csv with fields: id, text, label

Labels: 1 for Cancer, 0 for Non-Cancer

2. Model Selection

Classification Model

Model: DistilBERT from Hugging Face

Justification:

Retains ~97% of BERT's performance

60% faster, 40% smaller – ideal for environments like Google Colab

Seamless integration with Hugging Face's Trainer API

Supports LoRA fine-tuning for efficient parameter updates

Outperforms larger models in terms of training speed and ease for general classification

Disease Extraction Model

Model: en_ner_bc5cdr_md (SciSpaCy)

Justification:

Specialized for biomedical named entity recognition

Trained on the BioCreative V CDR corpus

Outperforms general NER models for disease entity detection

3. Fine-Tuning Process

Approach: LoRA-based fine-tuning of DistilBERT on binary classification task.

Notebook:

Question_1_and_3_Research_Paper_Analysis_&_Classification_Pipeline_Velsera.ipynb

Libraries Used:

transformers, datasets, scikit-learn, pandas, LoRA, PyTorch

4. Multi-Label Classification Output Format

Example:

```
{  
  "predicted_labels": ["Cancer"],  
  "confidence_scores": {  
    "Cancer": 0.92,  
    "Non-Cancer": 0.08  
  }  
}
```

5. Disease Extraction Output Format

Notebook: Disease_Specific_Identification_from_Abstracts.ipynb

Example:

```
{  
  "abstract_id": "PMC1234567",  
  "extracted_diseases": ["Lung Cancer", "Breast Cancer"]  
}
```