Report:

Part1:

My approach to this problem was simple but robust. Utilizing the BioBERT model, which is specifically trained on PubMed abstracts and PMC full-text articles. The model was tasked with extracting entities from clinical trials datasets across three distinct but related tasks, T1, T2, and T3. Each task was associated with one of three provided datasets: G1, G2, and G3. This approach leveraged BioBERT's robust understanding of biomedical language, enabling precise entity recognition across the different datasets and tasks. Below are the steps used to train a BioBERT model.

1. Data Preparation

The datasets used for training and testing were sourced from specialized biomedical texts. Key preprocessing steps included:

- **Train-test split using stratification:** To enhance model robustness and ensure balanced representation of various entity types, datasets were stratified based on the label combinations.
- **Data Loading:** Training and testing data were loaded from CSV files.
- **Data Cleaning:** Unnecessary columns were removed, simplifying the datasets for NER tasks.
- **Tag Adjustment:** The entity tags were adjusted to align with token indices, ensuring accurate entity recognition.
- **Tag Parsing:** Raw tag strings were parsed into structured formats, facilitating easier manipulation and processing.

2. Model Training and Evaluation

I fine-tuned the BioBERT model on our processed datasets. The following steps were undertaken:

- **Tokenization:** Text data were tokenized using the BioBERT tokenizer to convert text into a format suitable for model input.
- **Label Preparatio**n: Entity tags were converted into BIO (Beginning, Inside, Outside) format, and a custom function was applied to map these tags to corresponding token labels.
- **Model Setup:** A BioBERT model was initialized with a specified number of labels and loaded onto a GPU for efficient training.
- **Training Process:** The model was trained over multiple epochs, with gradient clipping implemented to stabilize training. Losses were recorded and plotted to visualize the training progress.

- **Model Saving**: The trained model and tokenizer were saved for future inference and validation.

3. Results

The trained model was evaluated using a test dataset. The main results included:

- **Prediction Performance:** The model's ability to accurately predict entity tags was assessed using a DataLoader to process the test set.
- **Classification Repo**rt: A detailed classification report was generated, providing metrics such as precision, recall, and F1-score for each entity type.

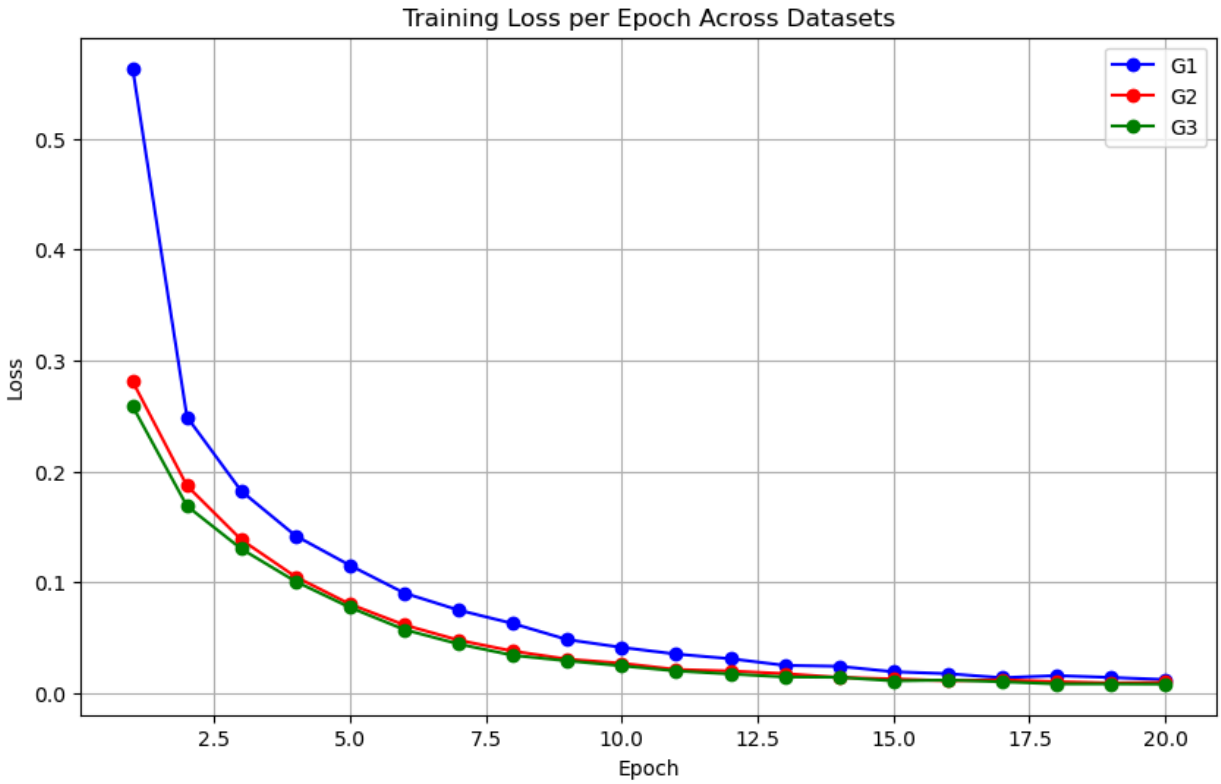
Here are the requested results from the Models.

	Performance on the test set of T1	**Performance on the test set of T1 and T2.**	**Performance on the test set of T1, T2 and T3.**	**Performance on combined G1+G2+G3**
Treatment F1	0.904	0.9	0.905	0.915
Chronic Disease F1	0.907	0.897	0.902	0.902
Cancer F1	0.906	0.898	0.899	0.882
Allergy F1	0.809	0.828	0.803	0.826
Other F1	0.952	0.946	0.947	0.946
Weight averaged F1	0.932	0.925	0.927	0.927

Conclusion and observation:

The model’s ability to identify medical entities changes based on how it’s trained. When trained progressively on datasets G1 through G3, it retains and even enhances its ability to recognize 'Treatment' and 'Chronic Disease' categories, suggesting it effectively holds onto previous knowledge. In contrast, its capability to detect 'Cancer' diminishes with more data, hinting at some challenges with memory retention. 'Allergy' recognition, on the other hand, gets better, likely benefiting from the richer variety of examples in the training process. The 'Other' categories maintain strong performance throughout, showing that the model is reliably robust across different training scenarios.

Model Training Loss:



If you carefully observe the the initial training losses for datasets G1, G2, and G3 start at different levels, with G1 beginning the highest, indicating it was the most challenging initially with all the random weights, followed by G2, and then G3, which had the lowest starting loss, suggesting it was the easiest for the model to initially adapt to.

A Brief discussion on model selection:

In this assignment, I have used the BioBERT,, a domain-specific adaptation of BERT for biomedical text, has shown significant improvements in performance on various biomedical NLP tasks, including Named Entity Recognition (NER) in clinical texts.

BioBERT is particularly one of the best models for clinical trials and biomedical data based on the literature available:

- BioBERT is pre-trained on large-scale biomedical corpora, which include PubMed abstracts and PMC (PubMed central) full-text articles. This domain-specific pre-training allows BioBERT to capture the unique language, terminology, and contextual understanding of biomedical literature better than general-domain models like BERT. BioBERT significantly outperforms BERT on several biomedical NLP tasks which involve

recognizing complex entity names and relationships that are common in clinical trials and medical records.

- In a systematic review conducted by researchers from texas and Hong Kong university revealed that pre-trained language models in the biomedical domain, particularly focusing on BioBERT and its related variants (BioALBERT, BioRoBERTa) outperforms ClinicalBERT, BlueBERT and achieves the best performance.
- In the same review, researchers showed that PubMedBERT outperformed BioBERT by very little margin in most of the NER tasks except NCBI Disease.

Reference:

Lee, Jinhyuk, et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." *Bioinformatics* 36.4 (2020): 1234-1240. Bioinformatics

Benyou Wang, Qianqian Xie, Jiahuan Pei, Zhihong Chen, Prayag Tiwari, Zhao Li, and Jie Fu. 2023. Pre-trained Language Models in Biomedical Domain: A Systematic Survey. ACM Comput. Surv. 56, 3, Article 55 (March 2024), 52 pages. Pre-trained Language Models in Biomedical Domain: A Systematic Survey.

Part 2. Suggest other approaches that can be taken to improve the performance of this process with respect to performance measures listed in Table 2.

To improve the model performance in continual learning and minimize the issues like catastrophic forgetting, we can apply a variety of techniques:

Regularization Techniques such as Elastic Weight Consolidation (EWC) help in maintaining crucial weights from prior tasks, preventing significant loss of previously learned information.

Rehearsal Techniques can be used, where the model is periodically trained on old or synthetically generated data to prevent existing knowledge.

Data Augmentation can improve generalization by introducing diverse or synthetic examples, especially for underperforming entities.

Training BioBERT on Facebook Research data:

I have also trained a BioBERT model on facebook research training data available on github (provided in the assignment) to see the performance of the model on each entity.

Here are the results:

	F1-SCORE
'age'	0.942

allergy_name	0.834
'bmi'	0.980
'cancer'	0.877
'chronic_disease'	0.886
'clinical_variable'	0.871
'contraception_consent'	0.892
'ethnicity'	0.869
'gender'	0.925
'language_fluency'	0.940
'lower_bound'	0.950
'pregnancy'	0.860
'technology_access'	0.877
'treatment'	0.892
'upper_bound'	0.952
‘other’	0.936
weighted avg	0.919

Detailed Explanation of the Data understanding and Stratified sampling in TASK 1, 2, and 3:

Data understanding: As a first step, I have gone through data. Here are some important observations:

1. In the ‘tags’ column the start and end Index for an entity was starting from 1 and going til +2 at the end.
    a. ***For example:***
        ('8:16:chronic_disease,20:32:treatment')
        ('portal fibrosis by liver biopsy')

        In this example, if we extract out words from the given indexes (8:16) – 'ibrosis '
        and (20:32) – 'iver biopsy '.

So, I have updated these indexes with correct values.

b. **I figured it out that there are only 4 labels present in the dataset:**

‘Unique labels found: {'treatment', 'cancer', 'chronic_disease', 'allergy_name'}
Number of unique label categories: 4’

Train test split using stratification:

I have divided our data into train test split using stratified sampling to have data balance from each category. I faced few challenges in doing stratified sampling because for one data row there might be more than one label present. There were few ways to handle this:

a. **Data Duplication:** In this approach I could have considered each label as a single row and duplicated the data. This approach is good for the small dataset but I have enough dataset so, I did not go with this technique.
b. Unique-label combination stratification: So, I figured out unique labels combinations present in the whole dataset like:

Unique label combinations found:
('cancer', 'chronic_disease', 'treatment')
('allergy_name',)
('allergy_name', 'treatment')
('allergy_name', 'chronic_disease')
('allergy_name', 'cancer', 'chronic_disease', 'treatment')
('treatment',)
('chronic_disease',)
('allergy_name', 'cancer', 'treatment')
('chronic_disease', 'treatment')
('cancer', 'treatment')
('cancer', 'chronic_disease')
('allergy_name', 'cancer')
('allergy_name', 'chronic_disease', 'treatment')
('cancer',)
Number of unique label combinations: 14

Then I have stratified our dataset based on unique label combinations after merging the combinations as ‘other’ which are having counts ‘<=10’.