Miimansa NLP Scientist: Hands-on Task

This hand-on evaluation has2 parts to it. First, the dataset is described briefly and the three parts of the problem are explained.

Dataset
Start with the Facebook clinical trial dataset publicly available here: https://github.com/facebookresearch/Clinical-Trial-Parser. We'll focus on the Named Entity Recognition (NER) task. The table below gives a brief overview of the dataset.

	Class	**Count**	**Examples**
Entity	Treatment	31K	surgery, remdesivir
	Chronic disease	26K	kidney failure, AD
	Cancer	9.3K	leukemia
	Gender	3.7K	—
	Pregnancy	2.8K	—
	Allergy	1.9K	allergy to aspirin
	Contraception consent	1.6K	—
	Language literacy	482	—
	Technology access	132	email, cellphone
	Ethnicity	82	—
Attribute	Clinical variable	13K	ECOG, Hgb count
	Age	2.6K	—
	Body mass index	289	—
Limit	Upper bound	14K	< 25 кд/m2
	Lower bound	14K	≥ 18

Table: 1
Source: Tseo, Yitong, et al. "Information extraction of clinical trial eligibility criteria." arXiv preprint arXiv:2006.07296 (2020).

Part 1. We have provided you with three datasets (G_1, G_2, G_3) with minimum overlap between them. These three datasets will define three different NER tasks T_1, T_2, and T_3, with the same set of labels but different entities. Concretely, in task T_1 the NER should recognize the entities in dataset G_1, assigning each a label from {Treatment, Chronic Disease, Cancer, Allergy, Other}. In task T2, the NER should recognize the entities in G_2, assigning each a label from the aforementioned set, and so on. You can keep aside some part of each dataset as a test dataset of that task. Eg, you might keep aside 20 percent of the data as the test set. All performance measures should be reported on the respective test set.
Continual learning: Train a model for the aforementioned tasks in a continual learning fashion, i.e., first, train the model for task T_1. Second, further train the same model on task T_2, keeping only **100** examples from task T_1. Now, train the same model the third time for task T_3. Again, you can keep only **100** examples from previous tasks(T_1, T_2). The objective would be to achieve the following properties:

Property	*Definition*
Knowledge retention	The model is not prone to catastrophic forgetting.
Forward transfer	The model learns a new task while reusing knowledge acquired from previous tasks.
Backward transfer	The model achieves improved performance on previous tasks after learning a new task.
Fixed model capacity	Memory size is constant regardless of the number of tasks and the length of a data stream.

Table: 2
Source: Biesialska, Magdalena, Katarzyna Biesialska, and Marta R. Costa-Jussa. "Continual lifelong learning in natural language processing: A survey." *arXiv preprint arXiv:2012.09823* (2020).

The training process provided above is one of a basic continual learning setup. Report the performance of the above steps and compare it with the model allowed to train on the combined dataset (G_1+G_2+G_3). At the very least report the metrics given in table 3 below on the test set that you create for each task.

You are free to use any other continual learning approach. Submit a brief report and your code.

Upload models after every episode on the huggingface hub. In each episode you train only for task in that episode, without looking back at the training dataset of previous episodes except a fixed set of training examples (100 examples from previous episode, in our case). We'll train your model on a similar held-out task (T_4 with dataset G_4) with100 examples provided by you from previous episodes and the dataset G_4. Provide a separate code from training and evaluation on T_4. Your code should produce all measures after training on task T_4.

	Performance on the test set of T_1	**Performance on the test set of T1 and T_2.**	**Performance on the test set of T1, T2 and T_3.**	**Performance on combined G_1+G_2+G_3**
Treatment F1				
Chronic Disease F1				
Cancer F1				
Allergy F1				
Other F1				
Weight averaged F1				

Table 3: Entity-wise and weight averaged strict F1 scores for each of the three tasks as well as the performance of the model trained on G_1+G_2+G_3

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.

Submission checklist: Submit a report that clearly demonstrates your approach, results as well as a discussion on these that you might wish to include. We highly recommend submitting this as a jupyter notebook with markdown text, but you are free to use other similar tools that you might prefer. Please verify that your submission includes the following:

a) Report and code for model training. Please include a function that takes a data set and a model as the input arguments and produces the metrics described in Table 3.
b) Link to your models on Huggingface
c) Table 3 populated with your results and submitted as a comma separated value file