Course Project :
COVID-19 Named Entity Recognition

COMP 4901K / MATH 4824B
Fall 2020

Named Entity Recognition (NER)

- NER seeks to find and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, and locations.

https://en.wikipedia.org/wiki/Named-entity_recognition

Named Entity Recognition (NER)

The task: find and classify names in text, for example:

The European Commission [ORG] said on Thursday it disagreed with German [MISC] advice.

Only France [LOC] and Britain [LOC] backed Fischler [PER] 's proposal .

"What we have to be extremely careful of is how other countries are going to take Germany 's lead", Welsh National Farmers ' Union [ORG] (NFU [ORG]) chairman John Lloyd Jones [PER] said on BBC [ORG] radio .

https://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture03-neuralnets.pdf

Dataset in this project

- A corpus containing the scholarly literature about COVID-19, together with the annotation for NER task.
- The dataset contains 64 fine-grained entity types:
  - General types (e.g., person, organization)
  - common biomedical entity types (e.g, genes, chemicals and diseases)
  - COVID-19 related types (e.g., coronaviruse, viral protein, substrates and immune responses)
  - The label ‘O’ means no type

Dataset

- We pre-processed and split the dataset into train, validation and test sets, which are stored in train.pkl, val.pkl and test.pkl respectively.
- Ground labels are provided for training and validation, while omitted for testing set.
- Please refer to **playground.ipynb** file for more details about the dataset (e.g., dataset format, how to load the dataset).

Goal of this project

- You are asked to predict the tag of every token in a given sentence.
  - ``Protection of calves against fatal enteric colibacillosis by orally administered Escherichia coli K99 - specific monoclonal antibody.''

Token	Protection	of	calves	against	fatal	enteric	colibacillosis	by
tag	O	O	LIVESTOCK	O	O	DISEASE_OR_SYNDROME	DISEASE_OR_SYNDROME	O

Token	orally	administered	Escherichia	coli	K99	specific	monoclonal	antibody
tag	GENE_OR_GENOME	GENE_OR_GENOME	GENE_OR_GENOME	GENE_OR_GENOME	GENE_OR_GENOME	CARDINAL	CARDINAL	CARDINAL

Hints

- Simple ways: use one-hot vector to represent each token, and apply a dense layer to predict the tag labels.
- Advanced methods: use lower dimensional embedding, and CNN/RNN to get contextualized representation of tokens, plus a dense layer to predict the labels.
- More fancy ways: use pretrained word embeddings (word2vec, Glove, etc, which you can find online), multiple layers of neural networks, bidirectional RNN/LSTM, etc.

Hints

- Hyper-parameters that you could tune:
  - learning rate
  - Optimizer (e.g., ADAM, SGD)
  - size of hidden layers
  - activation function
  - parameter initializer
  - …

Computational requirement

- For light-weight models, it is OK to only use CPU for this project. For instance, by using only CPU, a simple CNN model takes 10s/epoch, and for onehot+dense model, it takes 30s/epoch.
- If you want to use GPU but your own computer is insufficient for that, you might try the Kaggle Notebook (see the kaggle notebook.pdf file in the zip folder for details).

Evaluation metric

- Accuracy on test set => we provide the code for calculating accuracy in **playground.ipynb** and **evaluate.py.**
- You should **check your output format on validation set** using the provided codes in **playground.ipynb** and **evaluate.py** before submission**.**
- **By calling `evaluate("val_preds.csv", "data/val.pkl")`, it should return the accuracy of the validation set.**
- The .csv file that you submit should be **exactly the same as defined in the playground.ipynb.**

You need to submit the following

- The predictions on **test data**
  - **Make sure your output format is the same as that described in playground.ipynb.**
  - Make sure you can successfully evaluate your validation predictions on the validation data with the help of evaluate.py.
- Report (1~2 pages)
- Code (Frameworks and even programming languages are not restricted.)
- Due: 23:59, Nov 28, 2020
- Each **team leader** is required to submit the groupNo.zip file that contains `preds.csv`, the report, and the code of your team on canvas.
- we will check your report with your code and the accuracy on test set.

Grading Rule

Grade	Model (80%)	Report (20%)	Baseline models' accuracy on validation set
60%		submission	
80%	an easy model that most students can outperform	detailed explanation	83.1%
90%	a competitive model that about half students can surpass	detailed explanation and analysis	87.9%
100%	a very competitive model	excellent visualization and analysis	89.7%

Note: we will grade the performance of your model based on its accuracy score on the **test** set. Since you don't have access to the labels on the test set, you should compare your model's accuracy score on the **validation** set with the values in the rightmost column in the above table to have a rough reference of how your model performs.

Thank You