NER – Named Entity Recognition

Overview of the domain and problem statement

Now, let’s consider a hypothetical example of a health tech company called ‘BeHealthy’. Suppose ‘BeHealthy’ aims to connect the medical communities with millions of patients across the country.

‘BeHealthy’ has a web platform that allows doctors to list their services and manage patient interactions and provides services for patients such as booking interactions with doctors and ordering medicines online. Here, doctors can easily organise appointments, track past medical records and provide e-prescriptions.

So, companies like ‘BeHealthy’ are providing medical services, prescriptions and online consultations and generating huge data day by day.

Let’s take a look at the following snippet of medical data that may be generated when a doctor is writing notes to his/her patient or as a review of a therapy that he or she has done.

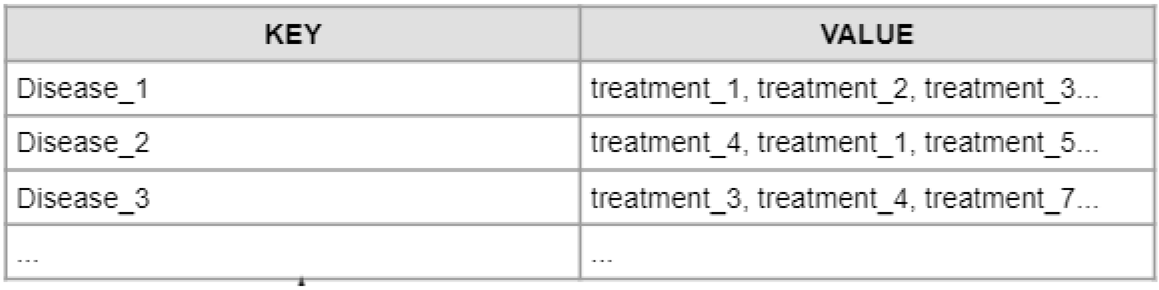
“The patient was a 62-year-old man with squamous cell lung cancer, which was first successfully treated by a combination of radiation therapy and chemotherapy.”

As you can see in this text, a person with a non-medical background cannot understand the various medical terms. We have taken a simple sentence from a medical data set to understand the problem and where you can understand the terms ‘cancer’ and ‘chemotherapy’.

Suppose you have been given such a data set in which a lot of text is written related to the medical domain. As you can see in the dataset, there are a lot of diseases that can be mentioned in the entire dataset and their related treatments are also mentioned implicitly in the text, which you saw in the aforementioned example that the disease mentioned is cancer and its treatment can be identified as chemotherapy using the sentence.

But, note that it is not explicitly mentioned in the dataset about the diseases and their treatment, but somehow, you can build an algorithm to map the diseases and their respective treatment.

Suppose you have been asked to determine the disease name and its probable treatment from the dataset and list it out in the form of a table or a dictionary like this.



KEY	VALUE
Disease_1	treatment_1, treatment_2, treatment_3...
Disease_2	treatment_4, treatment_1, treatment_5...
Disease_3	treatment_3, treatment_4, treatment_7...
...	...

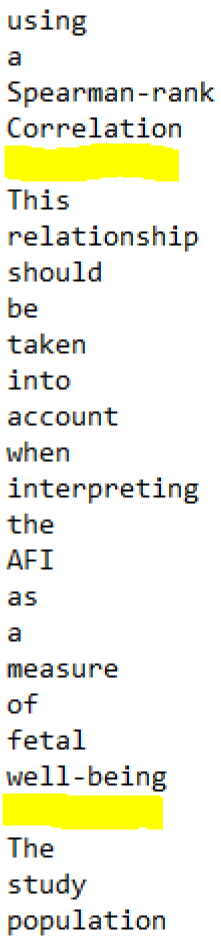
After discussing the problem given above, you need to build a custom NER to get the list of diseases and their treatment from the dataset.

There are four datasets provided to you to process, which are as follows:

- train_sent
- test_sent
- train_label
- test_label

You have the train and the test datasets; the train dataset is used to train the CRF model, and the test dataset is used to evaluate the built model.

First, you will understand the ‘**train_sent**’ and the ‘**test_sent**’ datasets. Let’s take a look at the structure of these datasets using the image provided below.



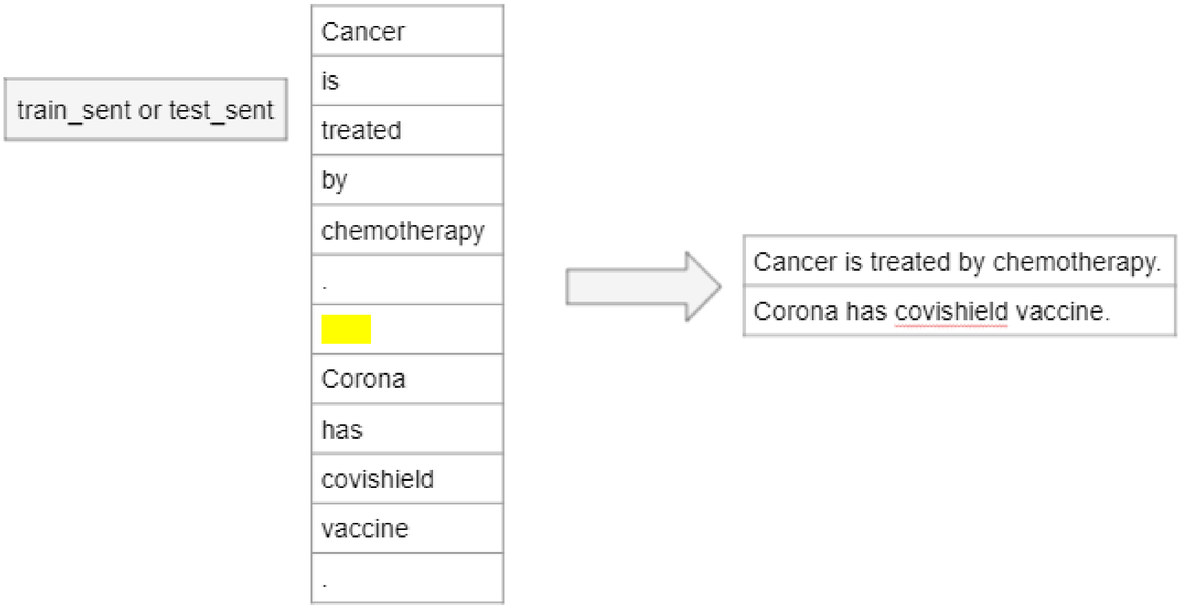
```
using
a
Spearman-rank
Correlation

This
relationship
should
be
taken
into
account
when
interpreting
the
AFI
as
a
measure
of
fetal
well-being

The
study
population
```

Here, you need to understand that each word in this dataset is provided in a single line. So, first, you need to club all these words together to form the sentences. Moreover, there are blank lines given in the dataset that have been highlighted in the image given above. These blank lines indicate that a new sentence is starting from the next line onwards to the next blank line.

In the image provided above, you need to make the sentences in the following way:

Sentence1: …using a Spearman-rank Correlation
Sentence2: This relationship should be taken into account when interpreting the AFI as a measure of fetal well-being.
Sentence3: The study population…
...and so on.

You can also refer to the image given below to get a better idea on how to create sentences from words.



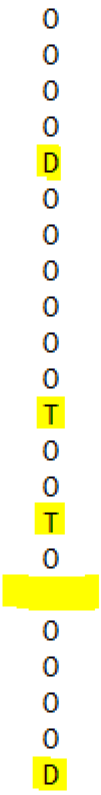
train_sent or test_sent

Cancer
is
treated
by
chemotherapy
.
🟨
Corona
has
covishield
vaccine
.

Cancer is treated by chemotherapy.
Corona has covishield vaccine.

In this ‘train_sent’ dataset, there are a total of 2,599 sentences when you form the sentences from the words. Similarly, there are a total of 1,056 sentences in the ‘test_sent’ dataset when you form the sentences from the words.

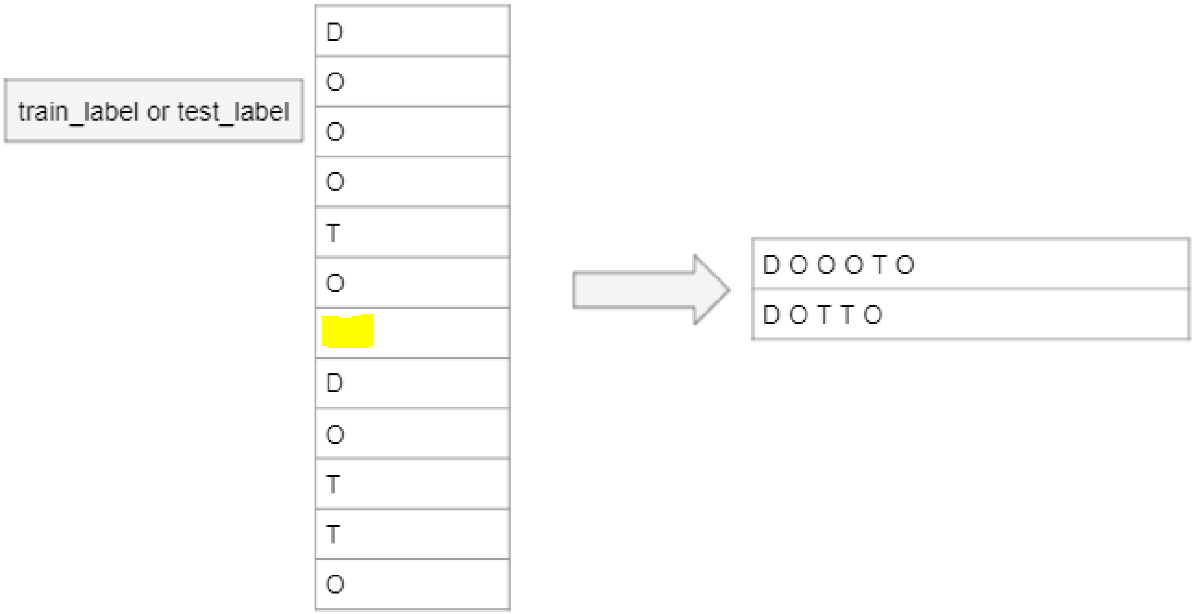
Now, let’s take a look at the next datasets that are named ‘**train_label**’ and ‘**test_label**’.



```
O
O
O
O
D
O
O
O
O
O
O
T
O
O
T
O

O
O
O
O
D
```

The above dataset is about the labels corresponding to the diseases and the treatment. There are three labels that have been used in this dataset: O, D and T, which are corresponding to ‘Other’, ‘Disease’ and ‘Treatment’, respectively.

These labels correspond to each word that is available in the ‘train_sent’ and 'test_sent' datasets. So, there is one-to-one mapping of each label available in the 'train_label' and 'test_label' datasets with the words that are available in the 'train_sent' and 'test_sent' datasets, respectively. You need to again create the lines of labels corresponding to each sentence in the ‘train_sent’ and the ‘test_sent’ datasets as shown below.



train_label or test_label

D
O
O
O
T
O
D
O
T
T
O

D O O O T O
D O T T O

So, in this ‘train_label’ dataset, there are a total of 2,599 lines of labels when you form the lines from the label dataset. Similarly, there are a total of 1,056 lines of labels in the ‘test_label’ dataset when you form the lines from the label dataset.

In this assignment, you need to perform the following broad steps:

- You need to process and modify the data into sentence format. This step has to be done for the 'train_sent' and ‘train_label’ datasets and for test datasets as well.
- After that, you need to define the features to build the CRF model.
- Then, you need to apply these features in each sentence of the train and the test dataset to get the feature values.
- Once the features are computed, you need to define the target variable and then build the CRF model.
- Then, you need to perform the evaluation using a test data set.
- After that, you need to create a dictionary in which diseases are keys and treatments are values.

In the next segment, you will get the exact idea on how to approach this assignment.

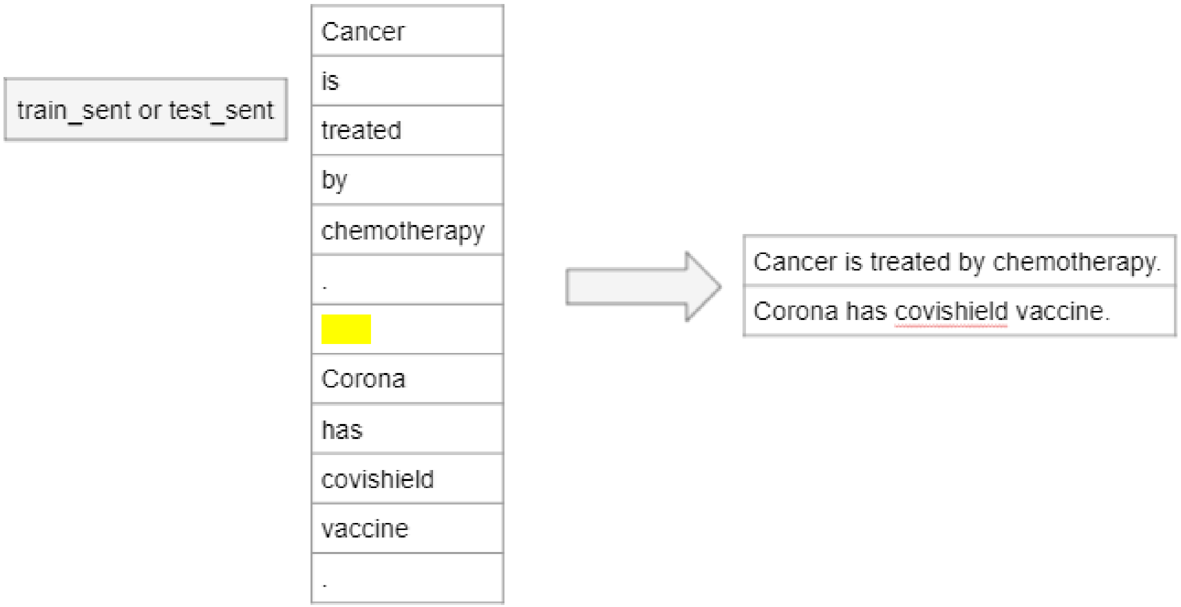
Tasks

There are eight major tasks that you need to perform to complete the assignment. They are as follows:

1. Data preprocessing
2. Concept identification
3. Defining the features for CRF
4. Getting the features words and sentences
5. Defining input and target variables
6. Building the model
7. Evaluating the model
8. Identifying the diseases and predicted treatment using a custom NER

Let's break down the steps into subtasks to understand this better.

Data preprocessing: As you are already aware that the dataset is in the token format instead of sentences, you need to construct the sentences from the words. There are blank lines after the completion of each sentence or a set of labels in label files ('train_label' and 'test_label') and you need to build a logic to arrange them into sentences or a sequence of labels in the case of label files. You can refer to the following two images to understand this better.

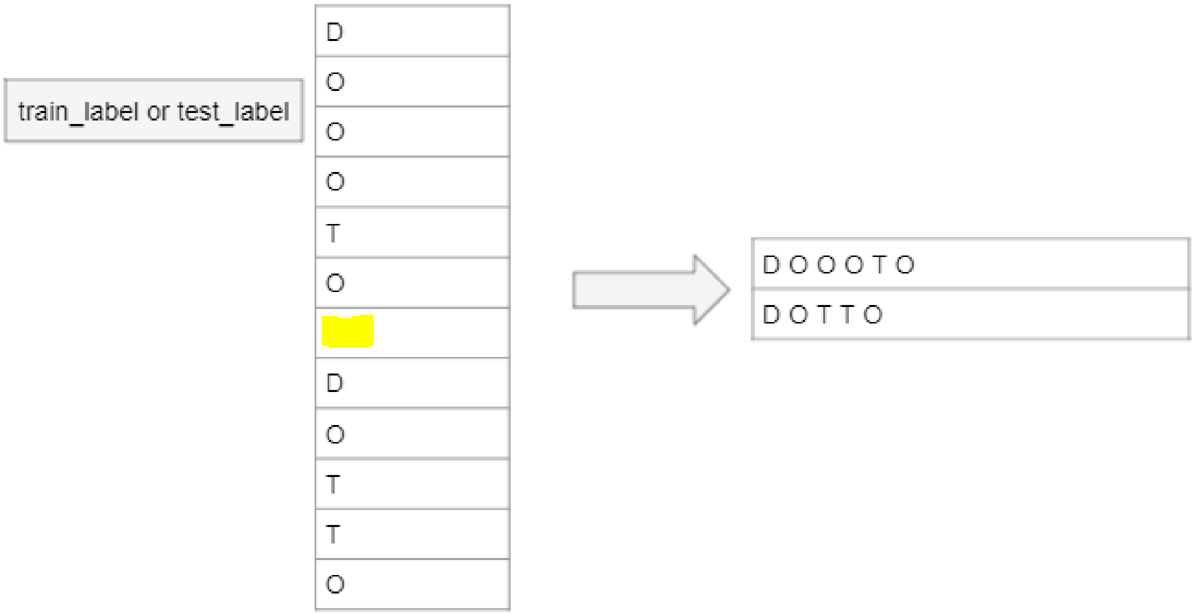


Cancer
is
treated
by
chemotherapy
.

Corona
has
covishield
vaccine
.

Cancer is treated by chemotherapy.
Corona has covishield vaccine.

A similar step is to be performed for the 'train_label' and 'test_label' datasets.



D
O
O
O
T
O

D
O
T
T
O

D O O O T O
D O T T O

You need to do the following three tasks after processing and modifying the datasets:

- Construct proper sentences from individual words and print five sentences along with their labels.
- Print the correct count of the number of sentences in the processed train and test dataset.
- Correctly count the number of lines of labels in the processed train and test dataset.

Concept identification: After preprocessing, we will first explore what are the various concepts present in the dataset. For this task, we will use PoS tagging. It is good to identify all the words from the corpus that have a tag of NOUN or PROPN (nouns) and prepare a dictionary of their counts. We will then output the top 25 most frequently discussed concepts in the entire corpus.

An important point to note here is that we are using both test and train sentences for concept identification. This is an exploratory analysis on the complete data. In this step, you need to perform the following two tasks by considering the train and the test dataset as a single unit of data:

- Use a toolkit like spaCy to extract those tokens that have NOUN or PROPN as their PoS tag and find their frequency from the **entire dataset** that comprises both the train and the test datasets.
- Print the top 25 most common tokens with NOUN or PROPN PoS tags for the **entire dataset** that comprises both the train and the test datasets.

Defining the features for CRF: Here, you need to perform the following three steps:

1. Define the features with the PoS tag as one of the features.
2. While defining the features in which you have used the PoS tags, you also need to consider the preceding word of the current word. The use of the information of the preceding word makes the CRF model more accurate and exhaustive.
3. Mark the beginning and the end words of a sentence correctly in the form of features.

Getting the features and the labels of sentences: In this step, you need to perform the following two tasks:

- Write the code to get the features' value of a sentence after defining the features in the previous step.
- Write the code to get a list of labels of a given preprocessed label line that you have created earlier.

Defining input and target variables: In this step, you need to perform the following two tasks:

- Extract the features' values for each sentence as an input variable for the CRF model in the test and the train dataset.
- Extract the labels as the target variable for the test and the train dataset.

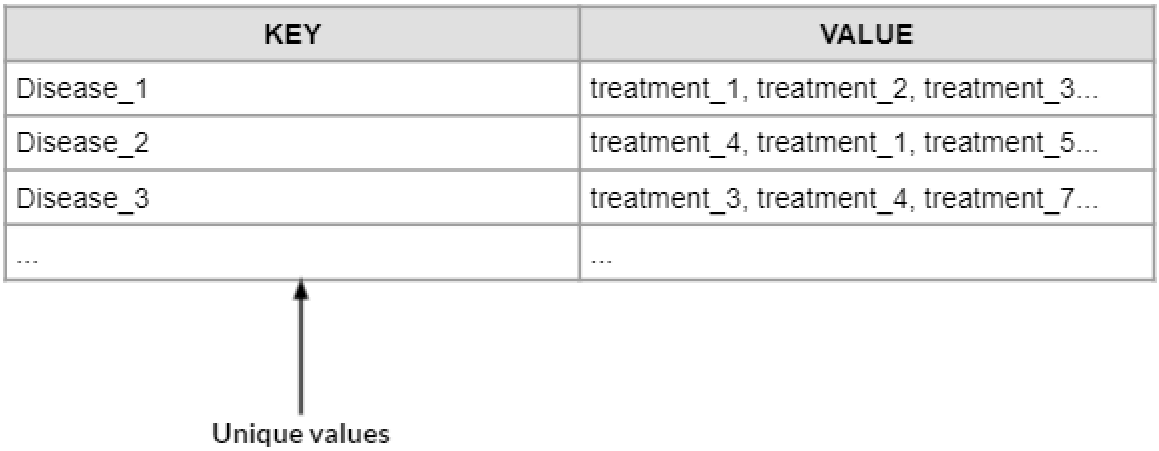
Building the model: You need to build the CRF model for a custom NER application using the features and the target variables.

Evaluation: Evaluate the model using the following two steps:

- Predict the labels of each of the tokens in each sentence of the test dataset that has been preprocessed earlier.
- Calculate the f1 score using the actual and the predicted labels of the test dataset.

Identifying the diseases and treatment using a custom NER:

- Create the code or logic to get all the predicted treatments (T) labels corresponding to each disease (D) label in the test dataset. You can refer to the following image to get an idea on how to create a dictionary where diseases are working as keys and treatments are working as values.



KEY	VALUE
Disease_1	treatment_1, treatment_2, treatment_3...
Disease_2	treatment_4, treatment_1, treatment_5...
Disease_3	treatment_3, treatment_4, treatment_7...
...	...

Unique values

- Predict the treatment for the disease named 'hereditary retinoblastoma'.

In this way, you will be able to finish this assignment. Let’s download the well-commented notebook that you can refer to solve this assignment.

You have been given the data in the form of tokens instead of sentences, and you need to process the data to get the sentences.

Please note that here, we are assuming that if there is a disease in the sentences, then the treatment mentioned in that sentence can be assumed to be the treatment for that disease. Also, there is an assumption that the same treatment can work for different diseases.

The next segment will help you understand the evaluation scheme for the assignment.