

Problem Statement:

Consider a hypothetical health tech company, BeHealthy, which aims to connect medical communities with millions of patients through a web platform. The platform enables doctors to list their services, manage patient interactions, and provide services such as booking appointments and ordering medicines online. Using this platform, doctors can easily organize appointments, track a patient's past medical records, and provide e-prescriptions.

Companies like BeHealthy generate large volumes of data by providing medical services, prescriptions, and online consultations. Let's take a look at the following snippet of medical data that may be generated when a doctor writes notes to their patient or prepares reviews of their patient.

“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 observe, a person with a non-medical background may not understand the various medical terms used in the text. We have taken a simple sentence from a medical data set to understand the problem. Here, you can understand the meaning of the terms “cancer” and “chemotherapy.”

Suppose you have been asked to identify a disease name and its possible treatment from a given data set and list it out in a table or a dictionary like this.

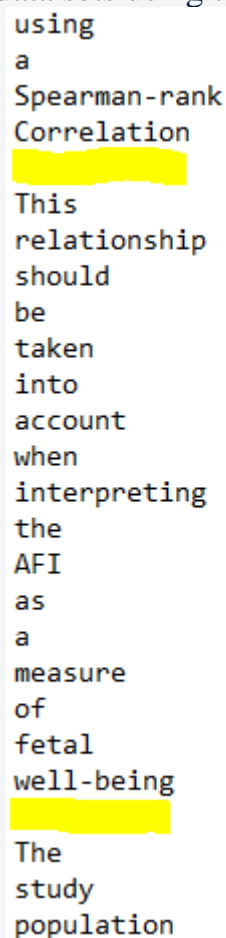
KEY	VALUE
Disease_1	treatment_1, treatment_2, treatment_3...
Disease_2	treatment_4, treatment_1, treatment_5...
Disease_3	treatment_3, treatment_4, treatment_7...
...	...

After analyzing the problem given above, you must build a custom NER to get the list of diseases and their treatment from the data set. Let's first download the data set given below.

You need to process four data sets:

- train_sent
- test_sent
- train_label
- test_label

You have the train and test data sets. The train data set is used to train the CRF model, and the test data set is used to evaluate the model you have built. First, you will understand the “**train_sent**” and “**test_sent**” data sets. Let's take a look at the structure of these data sets 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

You need to change the sentences in the image above in the following way.

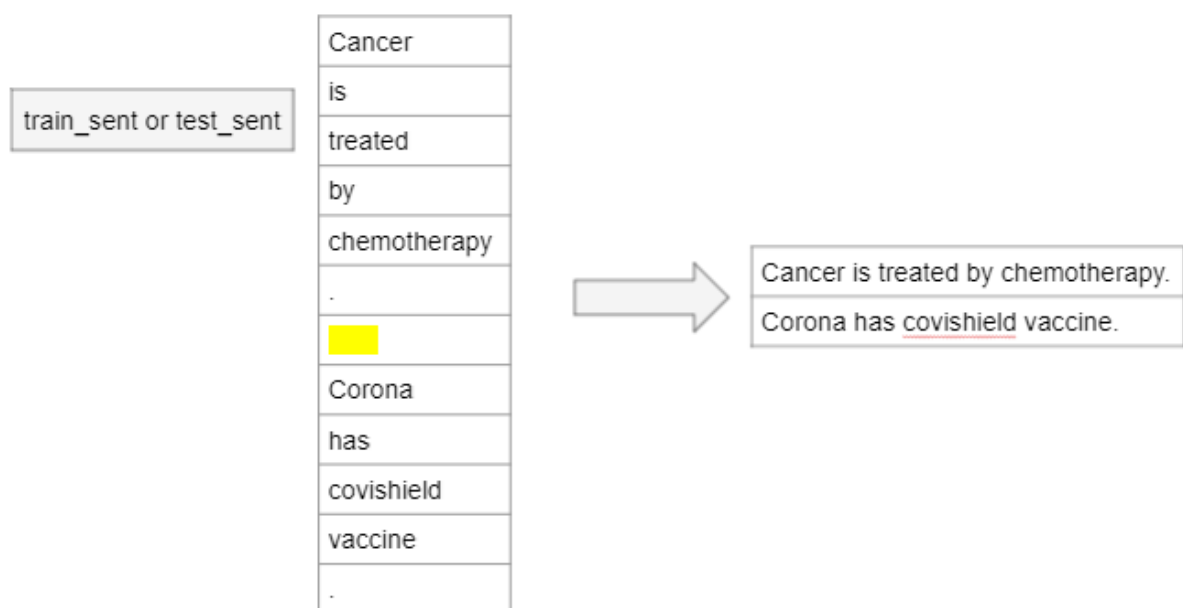
Sentence1: ...using a Spearman-rank Correlation

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 of how to create sentences from words. At every break, you can create a new sentence.

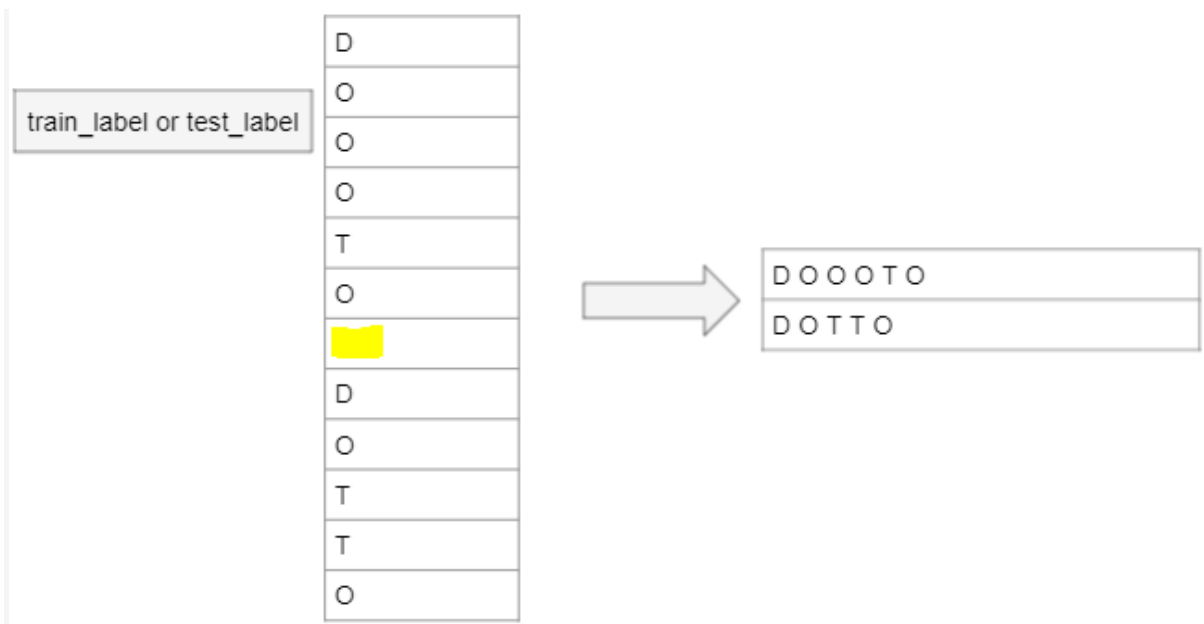


The "**train_sent**" data set contains **2,599** sentences formed from the words, and the "**test_sent**" data set contains **1,056** sentences formed from the words. Next, we will examine the "train_label" and "test_label" data sets.

O
O
O
O
D
O
O
O
O
O
O
O
T
O
O
T
O

O
O
O
O
D

The data sets discussed above pertain to labels associated with diseases and their respective treatments. It includes three labels: O for "Other," D for "Disease," and T for "Treatment." These labels match with the words in the “train_sent” and “test_sent” data sets. Therefore, there is a one-to-one correspondence between the labels in the “train_label” and “test_label” data sets and the words in the “train_sent” and “test_sent” data sets. You need to again create the lines of labels corresponding to each sentence in the “train_sent” and “test_sent” data sets as shown below.



So, you got an overview of the assignment and the structure of the data set. Let's take a look at the steps you need to follow to complete the assignment.

In this assignment

- You are required to process and modify the given data into the sentence format. You need to do this for the “**train_sent**” and “**train_label**” data sets and for test data sets.
- Next, you need to define the features to build the **CRF model**.
- Further, you must apply these features in each sentence in the train and test data set to get the feature values.
- Once the features are computed, you need to define the target variable and then build the CRF model.
- Next, you need to evaluate the model's performance using the test data set.
- Finally, you are required to create a dictionary in which diseases are the keys and treatments are the values.

NOTE

You are advised to use **Google Collab** for performing your demonstrations.

1. Reading the csv file command:

```
#give google drive permission

from google.colab import drive

drive.mount('/content/drive')

//read file

df = pd.read_csv("path in google drive/filename.csv")
```

2. Opening and reading the text file command:

```
#import the data from the drive here by uploading the dataset
and then reading it

from google.colab import drive

drive.mount('/content/drive')

with open('/content/drive/My Drive/Samsung.txt', 'r',
encoding="utf-8") as con:

    samsung_reviews = con.read()

    con.close()
```

Tasks

You must perform the following eight tasks to complete the assignment:

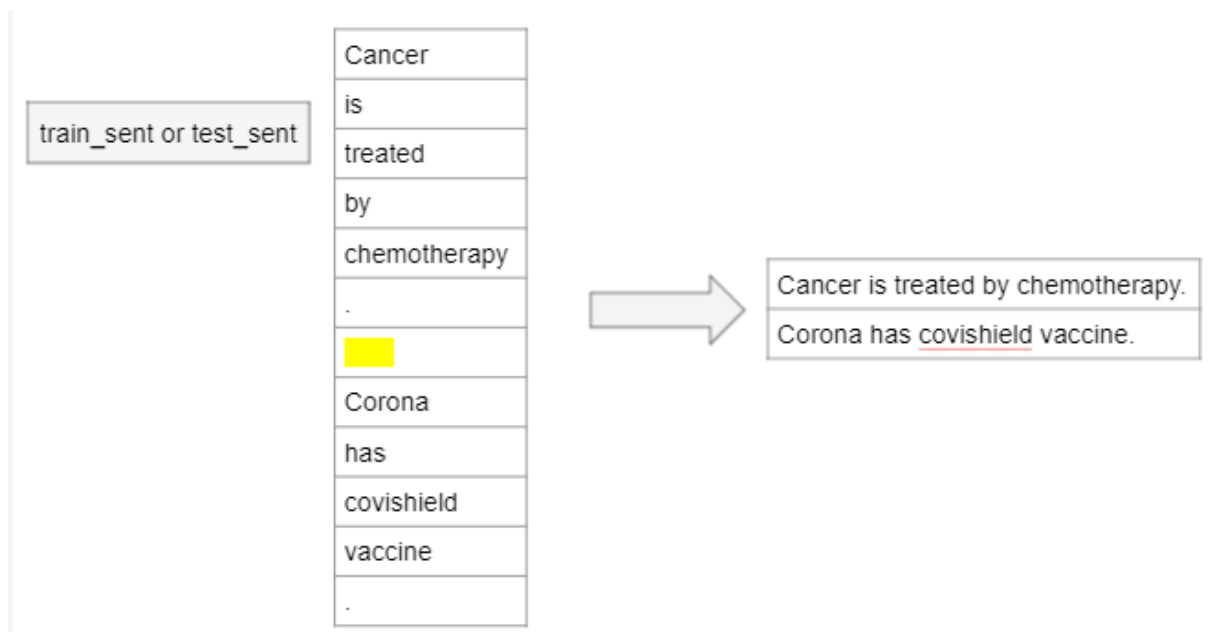
1. Data preprocessing
2. Concept identification
3. Defining the features of CRF
4. Getting the features, words, and sentences
5. Defining input and target variables
6. Building the model
7. Evaluating the model
8. Identifying the diseases and predicted treatments using a custom NER

Let's break down these tasks into subtasks.

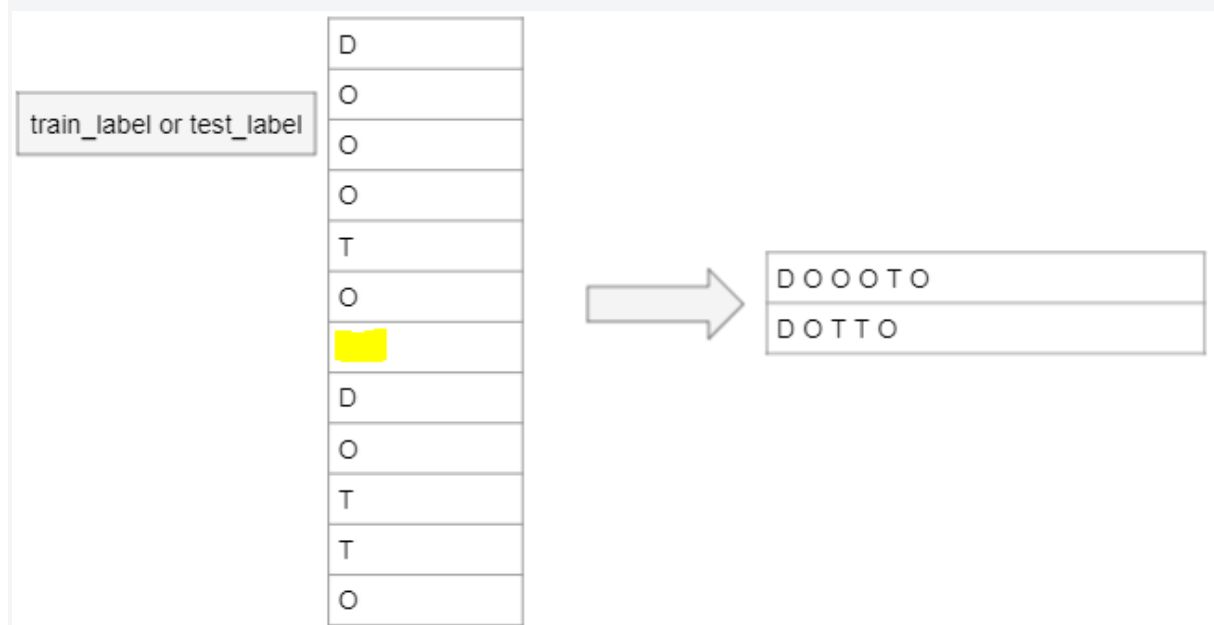
1. Data preprocessing: As you are already aware, the available data set is in the token format instead of sentences. So, you need to construct sentences from the words in the data set. There are blank lines after each sentence or a set of labels in label files (“train_label” and “test_label”), so you need to build the 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.

Hint: You can read the txt file using the following command.

```
train_sentences = process_file('filepath/train_sent.txt')
```



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



You need to complete the following three tasks after processing the data set:

- First, you need to organize the individual words in the data set into coherent sentences and then display a sample of these sentences by printing five of them along with their corresponding labels.
- You must ensure that the correct count of sentences is printed for the processed train and test data sets.

- Finally, you must accurately determine the number of lines of labels in the processed train and test data sets and correctly report this count.

2. Concept identification: After preprocessing the given data set, we will first explore the various concepts in the data set. For this, we will use PoS tagging. PoS tagging works well to identify all the words from a corpus with **NOUN or PROP**N (nouns) tag and prepare a dictionary of their counts. We will then output the **top 25 most frequently** discussed concepts in the 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 of the whole data set. In this step, you need to perform the following two tasks by considering the train and the test data set as a single unit of data:

- Use a toolkit like spaCy to extract the tokens with a NOUN or PROP
- N (nouns) tag and find the frequency of their occurrence in the entire data set (including train and test data sets).
- Print the top 25 most common tokens with NOUN or PROP
- N (nouns) tags for the entire data set (including the train and test data sets).

Defining the features for CRF: Here, you are building a CRF model, which you learned about in the previous session. For this task, you need to perform the following three steps. You learned how to define feature values in the session on NER.

- Define the features with the **PoS tag** as one of the features.
- While defining the features in which you have used the PoS tags, you also need to consider the word preceding the current word. The preceding word's information can be used to make the **CRF model more accurate and exhaustive**.
- Mark the first and last words in a sentence correctly in the form of features.

Getting the features and the labels of sentences: For this task, you need to perform the following two steps:

- 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 for a given preprocessed label line you created earlier.

Defining input and target variables: For this task, you need to perform the following two steps:

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

Building the model: You must build the CRF model (using the sklearn library) for a custom NER application using the features and target variables.

Evaluation: Evaluate the model by performing the following two steps:

- Predict the labels of each of the tokens in each sentence in the test data set preprocessed earlier.
- Calculate the f1 score using the actual and predicted labels in the test data set.

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 data set. You can refer to the following image to get an idea of how to create a dictionary where diseases are the keys and treatments are the 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 “**hereditary retinoblastoma.**”

In this way, you will be able to complete this assignment. Download the **well-commented notebook**. You can refer to the notebook to solve this assignment. The data provided is in the form of tokens, not sentences, so you need to process the data to get sentences.

Note: Here, we are assuming that if a disease is mentioned in the sentences, then the treatment mentioned in those sentences can be assumed to be the treatment for that disease. Also, we are assuming that the same treatment can work for different diseases.

Evaluation Rubrics

Criteria	Meets Expectations	Does Not Meet Expectations
Task 1 (10%)	<p>Constructs proper sentences from individual words and prints five sentences</p> <p>Correctly counts the number of sentences in the processed train and test data sets</p> <p>Correctly counts the number of lines of the labels in the processed train and test data sets</p>	<p>Does not process the data properly and does not print five sentences along with their labels</p> <p>Does not count the number of sentences in the processed train and test data sets</p> <p>Does not count the number of lines of the labels in the processed train and test data sets</p>
Task 2 (10%)	<p>Uses a toolkit like spaCy to extract those tokens that have NOUN or PROPN as their PoS tag and finds the frequency of their occurrence in the entire data set that comprises both the train and the test data sets</p> <p>Prints the top 25 most common tokens with NOUN or PROPN PoS tags for the entire data set that comprises both the train and the test data sets</p>	<p>Does not extract those tokens that have NOUN or PROPN as their PoS tag and does not find the frequency of their occurrence in the entire data set that comprises both the train and the test data sets</p> <p>Does not print the top 25 most common tokens with NOUN or PROPN PoS tags for the entire data set that comprises both the train and the test data sets</p>

Task 3 (25%)	<p>Defines the features with the PoS tag as one of the features</p> <p>While defining the features in which the PoS tags are used, the word preceding the current word is considered. The previous word's information can be used to make the CRF model more accurate and exhaustive.</p> <p>Marks the beginning and ending words in a sentence correctly in the form of features</p>	<p>Does not define the features with the PoS tag as one of the features</p> <p>Does not use the previous word while defining the features with the PoS tag</p> <p>Does not mark the beginning and ending words in a sentence correctly in the form of features</p>
Task 4 (10%)	<p>Writes the code to compute the features' value of a sentence</p> <p>Writes the code to get a list of labels of a given preprocessed label line created earlier</p>	<p>Does not write the code to compute the features' value of a sentence</p> <p>Does not write the code to get the labels of a sentence</p>
Task 5 (10%)	<p>Extracts the features' values for each sentence as an input variable for the CRF model in the test and train data sets</p> <p>Extracts the labels as the target variable for the test and train data sets</p>	<p>Does not define the features' values for each sentence as an input variable for the CRF model in the test and train data sets</p> <p>Does not define the labels as the target variable for the test and train data sets</p>
Task 6 (5%)	<p>Builds the CRF model for a custom NER application</p>	<p>Does not build the CRF model for a custom NER application</p>
Task 7 (10%)	<p>Predicts the labels of each of the tokens in each sentence in the test data set that has</p>	<p>Does not predict the labels of each of the tokens in each sentence in the test data set</p>

	<p>been preprocessed earlier</p> <p>Calculates the f1 score using the actual and predicted labels in the test data set</p>	<p>that has been preprocessed earlier</p> <p>Does not calculate the f1 score using the actual and predicted labels in the test data set</p>
<p>Task 8 (20%)</p>	<p>Creates the code or logic to get all the predicted treatment (T) labels corresponding to each disease (D) label in the test data set</p> <p>Predicts the treatment for the disease “hereditary retinoblastoma”</p>	<p>Does not create the code or logic to get all the predicted treatment (T) labels corresponding to each disease (D) label in the test data set</p> <p>Does not predict the treatment for the disease “hereditary retinoblastoma”</p>