# Problem Statement

Clinical studies often require detailed patients' information documented in clinical narratives. Named Entity Recognition (NER) is a fundamental Natural Language Processing (NLP) task to extract entities of interest (e.g., disease names, medication names and lab tests) from clinical narratives, thus to support clinical and translational research. Clinical notes have been analyzed in greater detail to harness important information for clinical research and other healthcare operations, as they depict rich, detailed medical information.

In this challenge, hackers are invited to extract all disease names from a given set of 20000 paragraphs/documents in the test set provided the labelled entities (diseases) for 30000 documents in the train set.

For example, here is a sentence from a clinical report:

*We compared the inter-day reproducibility of post-occlusive* ***reactive hyperemia*** *(PORH) assessed by single-point laser Doppler flowmetry (LDF) and laser speckle contrast analysis (LSCI).*

In the sentence given, reactive hyperemia (in bold) is the named entity with the type disease/indication.

# Data Description

The train file has the following structure:

| Variable | Definition |
|---|---|
| id | Unique ID for a token/word |
| Doc_ID | Unique ID for a Document/Paragraph |
| Sent_ID | Unique ID for a Sentence |
| Word | Exact word/token |
| tag | (Target) Named Entity Tag |

The target 'tag' follows the Inside-outside-beginning (IOB) tagging format. The **IOB format** (short for inside, outside, beginning) is a common tagging format for tagging tokens in named-entity recognition.

- The B-indications (beginning) tag indicates that the token is the beginning of a disease entity (disease name in this case)
- An I-indications (inside) tag indicates that the token is inside an entity
- An O (outside) tag indicates that a token is outside a disease entity

## Example

For more clarity, let's look at the same sample in the given tabular format, each row here corresponds to a word/token:

| id | Doc_ID | Sent_ID | Word | tag |
|---|---|---|---|---|
| 242 | 3 | 12 | We | O |
| 243 | 3 | 12 | compared | O |
| 244 | 3 | 12 | the | O |
| 245 | 3 | 12 | inter-day | O |
| 246 | 3 | 12 | reproducibility | O |
| 247 | 3 | 12 | of | O |
| 248 | 3 | 12 | post-occlusive | O |
| 249 | 3 | 12 | reactive | B-indications |
| 250 | 3 | 12 | hyperemia | I-indications |
| 251 | 3 | 12 | ( | O |
| 252 | 3 | 12 | PORH | O |
| 253 | 3 | 12 | ) | O |
| 254 | 3 | 12 | assessed | O |
| 255 | 3 | 12 | by | O |
| 256 | 3 | 12 | single-point | O |
| 257 | 3 | 12 | laser | O |
| 258 | 3 | 12 | Doppler | O |
| 259 | 3 | 12 | flowmetry | O |
| 260 | 3 | 12 | ( | O |
| 261 | 3 | 12 | LDF | O |
| 262 | 3 | 12 | ) | O |

The disease 'reactive hyperemia' is labelled using 'B-indications' for the word 'reactive' and 'I-indications' for the word 'hypermia'. All the other words that are outside 'reactive hyperemia' are labelled with 'O'.

Test file contains all the documents from which the participants need to extract the diseases. The format is as shown below:

| id | Doc_ID | Sent_ID | Word |
|---|---|---|---|
| 242 | 3 | 12 | We |
| 243 | 3 | 12 | compared |
| 244 | 3 | 12 | the |
| 245 | 3 | 12 | inter-day |
| 246 | 3 | 12 | reproducibility |
| 247 | 3 | 12 | of |
| 248 | 3 | 12 | post-occlusive |
| 249 | 3 | 12 | reactive |
| 250 | 3 | 12 | hyperemia |
| 251 | 3 | 12 | ( |
| 252 | 3 | 12 | PORH |
| 253 | 3 | 12 | ) |
| 254 | 3 | 12 | assessed |
| 255 | 3 | 12 | by |
| 256 | 3 | 12 | single-point |
| 257 | 3 | 12 | laser |
| 258 | 3 | 12 | Doppler |
| 259 | 3 | 12 | flowmetry |
| 260 | 3 | 12 | ( |
| 261 | 3 | 12 | LDF |
| 262 | 3 | 12 | ) |

## Submission Format

Submission must contain only the ID of the word (id), Sentence ID (Sent_ID) and predicted tag (B-indications/I-indications/O) as shown below:

| id | Sent_ID | tag |
|---|---|---|
| 242 | 12 | O |
| 243 | 12 | O |
| 244 | 12 | O |
| 245 | 12 | O |
| 246 | 12 | O |
| 247 | 12 | O |
| 248 | 12 | O |
| 249 | 12 | B-indications |
| 250 | 12 | I-indications |
| 251 | 12 | O |
| 252 | 12 | O |
| 253 | 12 | O |
| 254 | 12 | O |
| 255 | 12 | O |
| 256 | 12 | O |
| 257 | 12 | O |
| 258 | 12 | O |
| 259 | 12 | O |
| 260 | 12 | O |
| 261 | 12 | O |
| 262 | 12 | O |

- No other field is required in the submission
- Each tag must have one of B-indications/I-indications/O only
- Upload the submission only in **a zipped format** containing only the submission file
- sample_submission.zip is provided for reference **(Please submit only zipped submission file)**

# Evaluation Metric

The evaluation for this contest is based on **modified F1-Score** as explained below:

Suppose the ground truth has the following entities (mentioned in square brackets) for the given sentence

```
[Malaria] and [Yellow Fever] remain more deadly than [Hepatitis B] today
```

This has 3 entities.

Supposing the actual prediction has the following

```
[Malaria] [and] [Yellow] Fever remain more deadly than Hepatitis B [today]
```

We have an exact match for `Malaria`, false positives for `and` and `today`, a false negative for `Hepatitis B` and a substring match for `Yellow`. We compute precision and recall by first defining matching criteria. We are also trying to reward partial match here and not just exact entity match.

Here, True positives are of 2 types - Exact match and partial match and we are giving a weight of 1 to Exact Match and 0.5 to partial match. The computations are as follows:

Exact Match = 1 (`Malaria`) and Partial Match = 1 ( `Yellow` which overlaps `Yellow Fever`), False Positives =2 (`and`, and `today`), False Negatives = 1 (`Hepatitis B`)

Precision = (Exact Match + 0.5 * Partial Match) / (Exact Match + Partial Match + False Positives) = (1 + 0.5)/(1+1+2) = 0.375

Recall = (Exact Match + 0.5 * Partial Match) / (Exact Match + Partial Match + False Negatives) = (1 + 0.5)/(1+1+1) = 0.50

**F1 Score = (2 * Precision * Recall)/(Precision + Recall) = 0.428**

The counts of exact match, partial match, false positives and false negatives is summed across all sentences in the test set and overall F1 Score is the leaderboard score.

Please find the script for the evaluation metric implemented in Python at this link.

# Public and Private Split

The 20000 documents in test data are further randomly divided into Public (50%) and Private (50%) data.

Your initial responses will be checked and scored on the Public data. The final rankings would be based on your private score which will be published once the competition is over.