# **Report**

This is an annotated dataset for Named Entity Recognition (NER) problem.

Named Entity Recognition (NER) is a subtask of Information Extraction

This dataset is divided into train.txt, test.txt and valid.txt  
The Tokens are labelled under one of the following

tags [I-LOC, B-ORG, O, B-PER, I-PER, I-MISC, B-MISC, I-ORG, B-LOC]

Down loaded the Conll2003 data to test and conll.py for using helper functions

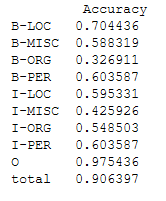
### **1a) Token-level performance (per class and total)**

First, we have to generated the parsed elements form Conll2003 dataset using Spacy

Here the function CalAcc compare the tags. The input to this function is generated form spacy2conll where alignment and processing is done using Token.whitespace attribute provided by package Spacy.

The after generation of Correct Sentence the 2 list are parsed using CalAcc function

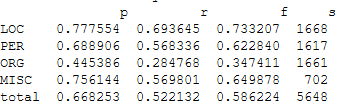
And output we get is Token and Class Level Accuracy.



### **1b) CoNLL chunk-level performance (per class and total)**

Here we have to use the module evaluation of conll.py to process the two datasets conll2003, docs\_corrected the above function is called using function conllEvaluation where conll and provided doc is compared.

Below we can see the Precision, recall, f-measure of correctly recognizing all the named entities in a chunk



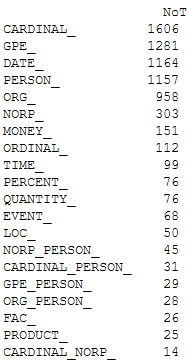
### **2) Grouping of Entities**

Here we are calling statFreq which is calling the groupEntities Function.

To get the Frequency List by grouping the Entities using the non\_chunks.

We are generating all the grouped entities for every sentence and after compute the frequency list for every group of them.

Below is the Output for top 20 entities:



### **3) Extend the entity span to cover the full noun-compounds**

### **(Fix segmentation errors)**

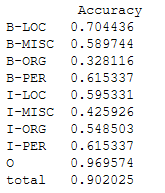
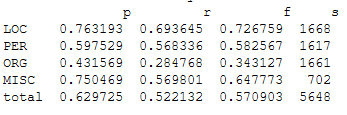
Her Basically we are Re-Evaluating the Tokens after fixing the segmentation Problem using the compound dependency.

As we know we have to fix error after So we are assuming to have a compound dependency with a second token and then we will re-assign it to first.

For this We are using compoundRectification function to get the corrected compound doc.

Then Accuracy is calculated using CalAcc by comparing the corrected doc with the Conll sentences. To get the Precision, recall, f-measure we are using the previously using conllEvaluation function.

Here there is no surety that Accuracy may increase, we can observe both increase or decrease

As we can see, using this method, the performance slightly decreases.