ECS763P: Natural Language Processing
Assignment 1: Sequence classification

Animesh Devendra Chourey

210765551

1 Split the training data

Splitting the data into training data and testing data

```
    train_data,test_data=train_test_split(raw_training_data,test_size=0.2,
                             random_state=42)
```

The two parts train data and test data contain 80% and 20% of the original data set respectively. This is done by train_test_split function present in the sklearn library. In order to maintain same fixed split of data, random state is specified.

2 Error analysis 1: False positives

The classification report is getting stored in the variable class report which is in the form of a dictionary. From this report, precision values are getting extracted and being stored in the sorted manner. We will be analysing on five of the classes with the lowest precision. False positives are the ones where the tag predicted by the classifier for the word is but in reality it is not that tag and the tag is not present in corresponding ground truth i.e. classifier incorrectly predicts the positive tag.

3 Error analysis 2: False negatives

Now the recall values are getting extracted from the classification report and being stored in the sorted manner. We will be analysing on five of the classes with the lowest recall. False negatives are the ones where the actual truth values do not have the tag but in the classifier predicted to be that tag i.e. classifier incorrectly predicts the negative class.

4 Incorporating POS tags as features

To add more features to the words POS tagging is used. This is done in order to boost up the classification accuracy percentage. POS tags are extracted from crf_pos.tagger file. Amongst every example, firstly words and bio tags are tokenized, and then POS tags are extracted and appended to the words only. The new entity becomes word@Tag with @ acting as a separator. This continues until every word is assigned a tag.

5 Feature experimentation

For this part more features are added so that the model is able to classify the words more accurately. These features are added to boost up the values of f-score, macro. The following new features have been added to the words : Punctuation, Number, Capitalization, Hyphen, Suffix. All these features captures various aspects of the language and determines the correct order of usage. The order of appearance determines the action and misplacing the order completely changes the meaning and motive behind the said sentence. Proper understanding of punctuation help in better understanding behind the meaning of the text. Capitalization helps in specifying the beginning of the sentence, serves as a signal for names and titles.

The meaning of a sentence depends on its words and it is extremely important to capture all the features behind the word to get better understanding of the language. We can opt to tune the L1 and L2 regularization coefficients parameter. Adjusting these hyperparameters can help optimize the algorithm even better for better performance. Here for this dataset setting the parameters c1 = 0.2 and c2 = 0.5 along with feature.minfreq = 2 generates the best possible result.

Without Features:

	precision	recall	f1-score
macro avg	0.62	0.51	0.54
weighted avg	0.81	0.82	0.81

With Features:

	precision	recall	f1-score
macro avg	0.68	0.60	0.63
weighted avg	0.84	0.85	0.84

Here we can clearly see that after adding the features the results in better classification.