---

1 Named Entity Recognition using the Constrained Method

For this project, using the F1 score was the primary metric in order to judge whether a linguistic model performed up to par. In this case, the F1 score needed to be .49, and with slight modifications needed to be around .60. With our final model, an F1 score of .65 on the testing data was achieved. The most significant steps taken to achieve this F1 score will now be discussed.

To start building the model, we used one simple feature to gather into a feature vector representation for the words. This was the proper case feature which classified the following: if the word being examined was the first word in a sentence, if the word was in proper case, or if the word was in all capital letters (with these being in order of priority). This feature, alongside many of the features that are going to be discussed, was not determined with a binary value. Rather, their returned values were "first", "proper", and "caps", respectively.

The models were going to be tested on the development data. In order to actually report on the effectiveness of this feature inclusion, several machine learning classes were tested within the scikit-learn package. Of them, a perception model with a iteration of 5 was used, giving us the following results:

```
accuracy:  92.64%; precision:  50.26%; recall:  49.25%; FB1:  49.75
              LOC: precision:  46.72%; recall:  67.99%; FB1:  55.38  1432
             MISC: precision:  21.67%; recall:  31.46%; FB1:  25.66  646
              ORG: precision:  57.76%; recall:  48.18%; FB1:  52.53  1418
              PER: precision:  67.06%; recall:  42.14%; FB1:  51.76  768
```

Next, the AdaBoost, such that perhaps the MISC and PER entities are classified with better precision.

Ada boost processed 52923 tokens with 4351 phrases; found: 1136 phrases; correct: 532.

```
accuracy:  86.65%; precision:  46.83%; recall:  12.23%; FB1:  19.39
              LOC: precision:  33.76%; recall:   8.03%; FB1:  12.97  234
             MISC: precision:  25.93%; recall:   1.57%; FB1:   2.97  27
              ORG: precision:  54.67%; recall:  26.18%; FB1:  35.40  814
              PER: precision:   1.64%; recall:   0.08%; FB1:   0.16  61
```

To no avail. A more detailed look of our experimentation with different sk-learn classifiers can be seen in comparisons.txt, a file provided in the .zip file submitted. To our surprise, the multi-layer perceptron model with an LGBFS solver and 200 iterations worked the best with our sole feature model, giving us the following results.

```
accuracy:  94.77%; precision:  55.29%; recall:  64.44%; FB1:  59.52
              LOC: precision:  53.01%; recall:  76.02%; FB1:  62.46  1411
             MISC: precision:  27.55%; recall:  36.40%; FB1:  31.36  588
              ORG: precision:  56.18%; recall:  63.06%; FB1:  59.42  1908
              PER: precision:  70.62%; recall:  67.27%; FB1:  68.90  116
```

Unfortunately, these results were not satisfying enough. Later on, the sole feature previously mentioned was edited such that the "first word" conditional value was removed. The rationale behind this was that due to the addition of other features, the underlying purpose of "proper case" was to stay as such. There are many other potential features that can be used to signify "first word". Additionally, the "first word" value can potentially damage the training, due to the fact that first words are always capitalized (most of the time). With the removal of this conditional, we achieved an overall F1 score of over .60.

To go beyond this, several further features were implemented. Among them was hyphen-detection, a feature that was suggested early on. However, this had no significant impact. Another feature, called "affix feature", was created. A list of popular Spanish prefixes and suffixes were gathered with a method used by an affected NER model. Going through all of the words in the data set, prefixes and suffixes of length 4 (simply the first 4 and last 4

characters of each word) were gathered. Of these, the ones that occurred at least 100 times in the data set were kept in a list, affixes.txt. All of these affixes had a length of 4, so if a word had a length of 4, it was checked if any affix in the list was contained within it. This feature is determined with a binary value, unlike several others.

Another feature that was added was short word shape. The shapes added were the following:

- Xx., e.g., "Corp."
- X. e.g., "INC."
- X., e.g., "INC"
- Xx, e.g., "Noah"
- x, e.g., "sam"
- D, e.g., (something that contains numbers)

With all of these features added, another test was performed with the following results.

```
accuracy:  94.75%; precision:  55.52%; recall:  63.69%; FB1:  59.32
              LOC: precision:  52.88%; recall:  75.61%; FB1:  62.23  1407
             MISC: precision:  25.61%; recall:  30.79%; FB1:  27.96  535
              ORG: precision:  57.39%; recall:  62.59%; FB1:  59.88  1854
              PER: precision:  69.12%; recall:  67.59%; FB1:  68.35  1195
```

As we can see, not much improvement. Another feature was added in, that being a gazetteer. However, this was not a run-of-the-mill document or corpus. Rather, a data set provided by GeoNames.com was used. Here, the list of all available locations provided in Mexico was used. This data was cleaned such that any word in the set that contained non-alphabetical characters was discarded. Additionally, all of these words were set to lowercase. Words from the test set were temporarily lowercased when within this feature function, such that if any of the words appeared within our "gazetteer", a binary value was given.

With all of these features combined, the following scores were retrieved.

```
accuracy:  94.82%; precision:  55.59%; recall:  64.77%; FB1:  59.83
              LOC: precision:  54.53%; recall:  75.91%; FB1:  63.47  1370
             MISC: precision:  25.80%; recall:  34.38%; FB1:  29.48  593
              ORG: precision:  57.65%; recall:  63.59%; FB1:  60.48  1875
              PER: precision:  67.99%; recall:  68.49%; FB1:  68.24  1231
```

Again, not so good. A final addition to our NER model was made. The number of features would be multiplied by 3: For the given word in question, all of the feature values named here would be decided. In addition, feature values for the preceding word would be decided as features by themselves. The same was done for the proceeding word. The reason for this was because upon examination of the output file, a very particular pattern was observed. It appeared that when for example, let's say a 3-word grouping, was misclassified, the misclassification occurred at the first word, and carried on towards the end the the grouping with the same label. For example, let's say there exists a 3-word grouping labeled as B-PLACE, I-PLACE, I-PLACE. Our model could, for example, misclassify it as B-ORG, I-ORG, I-ORG. As we can see, this type of misclassification occurrs in such an intuitive manner, that perhaps the model did not give weights to the 1st words when we are currently trying to classify the second. This way, there is a guarantee that groups of words will have more influence on each other, since they are all going to be factored (on a 3-word grouping bases) into each other's features. With this addition, we got the following results:

accuracy: 95.16LOC: precision: 53.16MISC: precision: 24.76ORG: precision: 58.85PER: precision: 77.05
And we are over .60 for sure!