Feature Design and improvements on Logistic Regression:

1>'FIRST NAME' and 'LAST NAME'

The twitter data evidently consists of a lot of names and so, the 'FIRST NAME' and 'LAST NAME' features will be very useful to distinguish a word that represents a person from another word which is not a person. These features will have boolean true value only for this word which helps the classifier to learn that the feature is an attribute of the 'PERSON' entity. I used the corresponding lexicons to find if the word occurs as a First or Last name.

The average FB1 score went up from 20.05 to 21.7

The average F1 score on dev and dev-test was 0.915

eg: "Michael" is a first name and it would have "FIRST NAME" feature set to true where as "Book" would not have this feature set to true.

2>'LOCATION'

Similar to the above argument, I used the location based lexicons like location,country and venue to predict if the word belongs to one of these lexicons.If so, the feature 'LOCATION' was set to true.

Also, by intuition we can think that all the words with "field" in word can be indicative of a location

eg: "Bakersfield" is a location and the feature would be true for this word

The average FB1 score went up from 21.7 to 22.3

The average F1 score on dev and dev-test remained 0.915

3>'SPORTS'

On the same lines, I used the sports lexicons sports league and sports teams to set the feature true for words that indicate a sports context

The average FB1 score went up from 22.3 to 24.1

The average F1 score on dev and dev-test remained 0.915

4>'MUSIC'

The twitter data seems to have content about various music artistes of today's generation. And so,I gathered a list of all the Pop artists of the today and obtained the lexicon with this list. I used this to have 'MUSIC' feature set to true if they are associated with a word in this lexicon.

The average FB1 score went up from 24.1 to 25.6

The average F1 score on dev and dev-test remained 0.915

5>'PRODUCT'

The data also had a lot of commercial products. And so, like above approaches I used the 'automotive.model' and 'product' lexicons to classify the word as a product by setting this feature to true.

The average FB1 score went up from 25.6 to 26.5

The average F1 score on dev and dev-test remained 0.915

6>'TV SHOWS'

And on the same lines, used the TV programs network and channel lexicons to classify the word as a TV related word using the 'TV SHOWS' feature.

The average FB1 score went up from 26.5 to 28.1

The average F1 score on dev and dev-test remained 0.915

7>'COMPANY'

As we did for the product feature, I used the product makers lexicon automotive.make to identify the brand that makes these products and the feature is set to true only for those words that are found in the lexicon.

The average FB1 score went up from 28.1 to 29.00

The average F1 score on dev and dev-test remained 0.915

8>'STOP WORDS'

Also,I used the english.stopwords lexicon to identify the stop words in the twitter data and set this feature to true.

The average FB1 went up from 29.00 to 31.2

The average F1 score on dev and dev-test remained 0.915

9>'POS TAGGING'

I did realize that the POS labelling can help identify the word to be of a particular category. For example, The Nouns such as names and places are tagged as 'NN' where as another word like 'walking' would not have this tag and hence it would be useful to have the POS TAGS themselves as features to uniquely identify the words.

The average FB1 score went up from 31.2 to 33.4

The average F1 score on dev and dev-test remained 0.915

10>'HYPERNYMS'

Also, we can view NER task as somewhat similar to identifying the category of a given word ,in other words, it is trying to find the parent/ancestor class to which it belongs to. And therefor I used the NLTK's WORDNET to find the nearest hypernym and use the same as a feature.

eg: "pizza" and "pasta" both words have the same Hypernym "dish". This can help classify all the words of a particular category based on its parent class name.

The average FB1 score went up from 33.4 to 35.65

The average F1 score on dev and dev-test increased to 0.92

11>'WORD SHAPES'

Another feature that can be useful to generalize words into a category is the 'word shape' feature. I used 'X' to encode [A-Z] and 'x' to encode [a-z] and 'd' to encode [0-9]. This feature converts a date like 29th to 'ddxx' and thus identify all the dates to be a particular category for which this feature would be true.

The average FB1 score finally went up from 35.65 to 37.57

And the final FB1 score on the test was 29.97

The average F1 score on dev and dev-test remained 0.92

The other features tried but did not increase the accuracy are :-

12>Identifying the special characters .

The idea was to identify the ! and other characters that people use and mark them as a group. The accuracy increased marginally by an average difference of (20.05-19.355=0.695)

13>Identifying internet links starting with "http"

The average FB1 decreased from 20.05 to 19.24

14>Identifying Emoticons:

This decreased the average as well and this reinforced our understanding that, this entity cannot be tagged and ends up in the 'other' category