NLP Class Project (CWI)

Features:

- Part of speech tag *(pos)*
- Length of word *(length)*
- Word's Scrabble score *(scrabble)*
- Word exists on Simple Wiktionary's top 1000 word list *(common)*

This list contains some classic NLP features (PoS tag) as well as some more novel features like Scrabble score, a well known scoring system for the complexity of a word given it's letter composition.

Method:

Using a simple perceptron over these features, including randomisation and averaging.

The code can be found on: https://github.com/robjtede/com4513-cwi

The common to run the program is:
```
python3 assignment.py -sag -e 25 -v
```

Results:

Even with this reasonably simplistic method, the results still look good.

PoS + Common:

Macro-F1: 0.25 *(something went wrong in the scoring of this one)*

Label	Precision	Recall	F1
0	0.00	0.00	0.0
1	0.38	1.00	0.55

Common + Scrabble:

Macro-F1: 0.52

Label	Precision	Recall	F1
0	0.95	0.28	0.4
1	0.46	0.98	0.62

PoS + Length + Common:

Macro-F1: 0.71

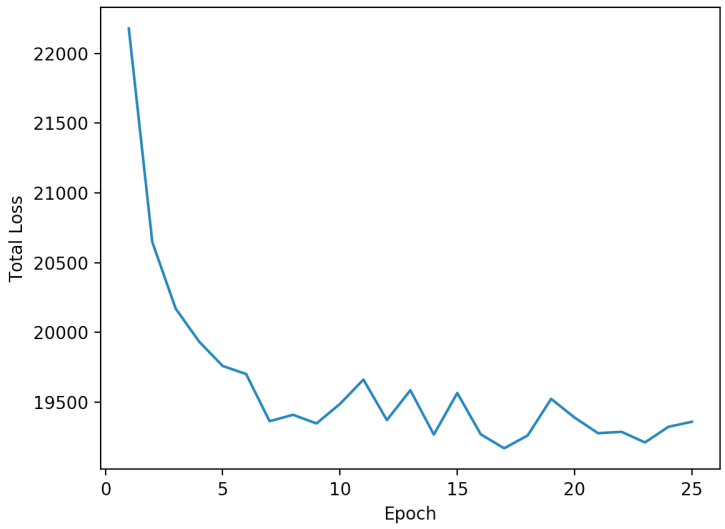
Label	Precision	Recall	F1
0	0.89	0.60	0.7
1	0.58	0.88	0.70

All Features:

Macro-F1: 0.75

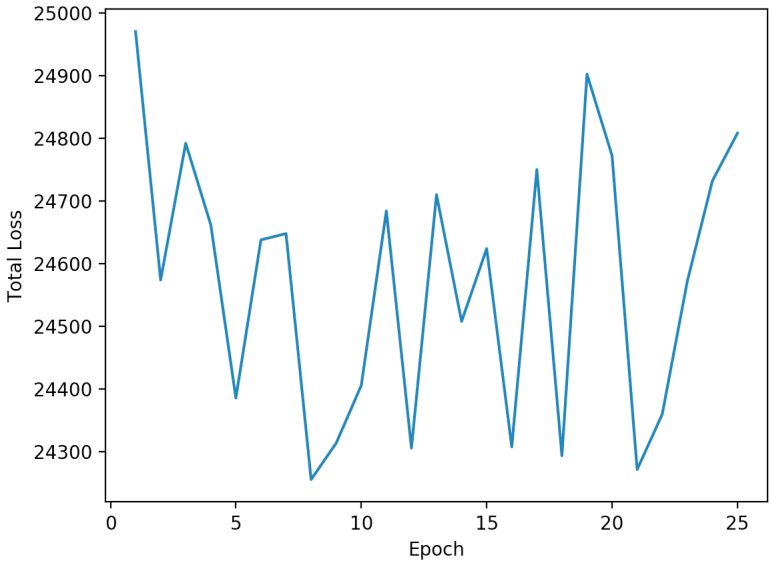
Label	Precision	Recall	F1
0	0.83	0.78	0.8
1	0.68	0.74	0.71

The corresponding loss function for this run is shown below:



Conclusions:

- The perceptron seems to respond quite significantly to the length of the word in question, with longer words being correctly classified as complex. When length is removed as a feature, the loss function starts to vary massively without converging on a value:



- The scrabble scoring system does seem to provide some benefit to the classifier, improving the Macro-F1 score by 0.04 over the run without it.
- In all cases (except the failed scoring run) the precision for the 0 label (non-complex word) is much higher. This is likely due to non-complex being the more common classification.
- Giving the perceptron more features to work with generally yields a more accurate classifier.