4. Maximum Entropy Markov Model (MEMM) tagger

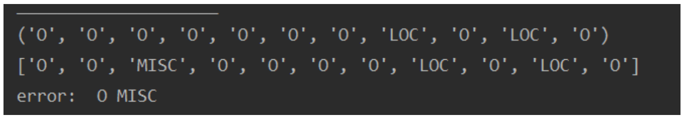
d) For the greedy decoding speedup, we used a dictionary with keys (curr_word, next_word, prev_word, prevprev_word, prev_tag, prevprev_tag) and the most probable tag according to the trained model as values. This way we do not need to calculate predictions and features to words-tags combinations we have already seen before. This gave better results than not using a dictionary at all and from using an extra dictionary with a whole sentence as key.

For the Viterbi decoding, we achieved speed up in running time by vectorizsing all the features vectors of each pair u, t "together" (i.e. calling the vectorizer only once, on a sequence of feature lists), and calculating the prediction probabilities vectors for each such pair together (in one call to the model prediction method).

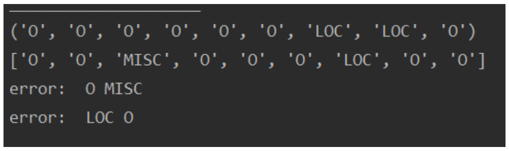
e) F1 score for greedy decoding: 0.85

F1 score for Viterbi decoding: 0.84

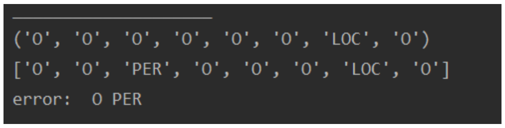
f) Analyzing the errors the model made, we can see that it tends to confuse 'O' with other tags, especially when the surrounding environment contains a lot of 'O's. For instance:



```
('O', 'O', 'O', 'O', 'O', 'O', 'O', 'LOC', 'O', 'LOC', 'O')
['O', 'O', 'MISC', 'O', 'O', 'O', 'O', 'LOC', 'O', 'LOC', 'O']
error:  O MISC
```

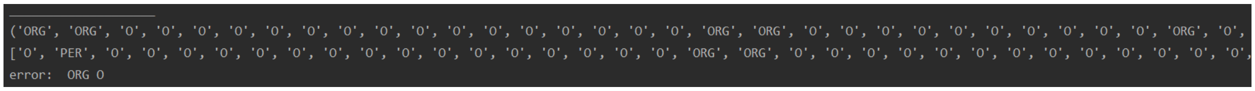


```
('O', 'O', 'O', 'O', 'O', 'O', 'LOC', 'LOC', 'O')
['O', 'O', 'MISC', 'O', 'O', 'O', 'LOC', 'O', 'O']
error:  O MISC
error:  LOC O
```

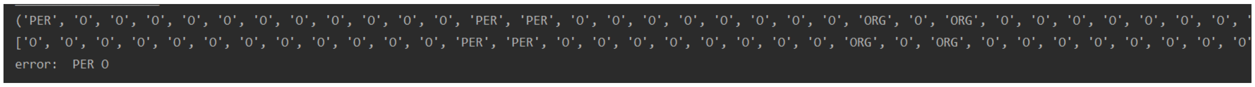


```
('O', 'O', 'O', 'O', 'O', 'O', 'LOC', 'O')
['O', 'O', 'PER', 'O', 'O', 'O', 'LOC', 'O']
error:  O PER
```

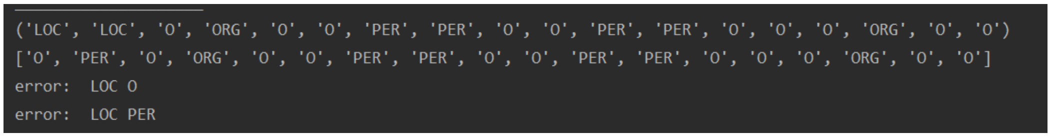
It is also noticeable that if a certain sentence's prediction contains a mistake it is likely to occur in the beginning of the sequence.



```
('ORG', 'ORG', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'ORG', 'ORG', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'ORG', 'O', 
['O', 'PER', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'ORG', 'ORG', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
error:  ORG O
```



```
('PER', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'PER', 'PER', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'ORG', 'O', 'ORG', 'O', 'O', 'O', 'O', 'O', 'O', 'O', '
['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'PER', 'PER', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'ORG', 'O', 'ORG', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O'
error:  PER O
```



```
('LOC', 'LOC', 'O', 'ORG', 'O', 'O', 'PER', 'PER', 'O', 'O', 'PER', 'PER', 'O', 'O', 'O', 'ORG', 'O', 'O')
['O', 'PER', 'O', 'ORG', 'O', 'O', 'PER', 'PER', 'O', 'O', 'PER', 'PER', 'O', 'O', 'O', 'ORG', 'O', 'O']
error:  LOC O
error:  LOC PER
```

(In the above examples, the first line is the gold tags and the second line is the greedy-prediction).

5. BiLSTM tagger

b) (i) If we hadn’t used masking, we would be including in our loss the “error” of the predictions our model gave to the extra zero-labeled positions, hence affecting the gradient and parameters learning. i.e. we could be updating the parameters in directions that are “not relevant” to the training data. Using the mask zeros out these unnecessary penalties from the loss function.