Jaykumar Patel Homework2
110255934

1. (i) Since we used MLE for this part it was obvious that tags like ‘,’ , ‘ ‘’ ‘, ‘$’ had 100% precision. If we look at the tag was predicted most amount of time it was NN, this makes sense because this we always picked the tag with highest probability and since NN shows up many times train data with many different words, this could happen.
(ii) This method has Accuracy higher than regular MLE which made sense because we not have more information about each word and tag. Since we are taking more stuff into account. We have additional information about the tag previous to this one. One of the problems I notice was since we were taking logs to prevent underflow, we had to take log of 0. Which is not possible so we took log very small number but this might cause some problems in terms of ties. This one also has same problem as the previous method, tags that shows up least are the ones to appear least on the output. This might be something we can’t fix.
(iii) This method surprisingly has lower accuracy from all the methods. This is weird because unlike MLE we are smoothing the zeros, this is not the best method of smoothing but it should have gave some sort of improvement it is prob because we didn’t have big enough training data. Also we haven’t picked a way to break ties during the process of finding max.

2.

```
['-RRB-', 'RBR', 'JJS', '-LRB-', 'PDT', 'SYM', 'RBS', 'JJR', 'NNPS', 'WP$']
-RRB- {'NN': 1}
RBR {'NN': 2, 'JJ': 2, 'DT': 2, 'IN': 3, 'TO': 1}
JJS {'NN': 2, 'IN': 3, 'DT': 2, 'JJ': 5}
-LRB- {'IN': 1}
PDT {'IN': 1}
SYM {'NNP': 1}
RBS {'JJ': 1}
JJR {'RBR': 3, 'NN': 2, 'DT': 2, 'NNP': 2, 'JJ': 3, 'IN': 2, '-NONE-': 2}
NNPS {'MD': 2, 'NN': 10, 'VBZ': 2, 'DT': 3, 'NNP': 11, 'NNS': 4, 'JJ': 3}
WP$ {'IN': 1}
```

NNPS, is confused with NN and NNP which makes sense. Since the count of these are almost same and making it hard to tell the difference. NN is one of the that showed up in place of many different tags reason being we have observed NN many times in the train data.