Question 1

```
Epoch 01 | avg loss/token: 0.7984 | 84.2s
Epoch 02 | avg loss/token: 0.5678 | 88.3s
Epoch 03 | avg loss/token: 0.5058 | 93.7s
Epoch 04 | avg loss/token: 0.4784 | 92.0s
Epoch 05 | avg loss/token: 0.4572 | 85.0s

Dev accuracy: 0.9324618736383442
Test accuracy: 0.9331896551724138

[01] Words: ['the', 'flight', 'should', 'arrive', 'at', 'eleven', 'a.m', 'tomorrow', '.']
     Pred: ['DT', 'NN', 'MD', 'VB', 'IN', 'CD', 'RB', 'NN', 'PUNC']
[02] Words: ['i', 'would', 'like', 'to', 'find', 'a', 'flight', 'that', 'goes', 'from', 'la', 'guardia', 'airport', 'to',
'san', 'jose', '.']
     Pred: ['PRP', 'MD', 'VB', 'TO', 'VB', 'DT', 'NN', 'IN', 'VBZ', 'IN', 'NNP', 'NNP', 'VBP', 'TO',
'NNP', 'NNP', 'PUNC']
[03] Words: ['show', 'me', 'the', 'flights', 'from', 'newark', 'to', 'los', 'angeles', '.']
     Pred: ['VB', 'PRP', 'DT', 'NNS', 'IN', 'NNP', 'TO', 'NNP', 'NNP', 'PUNC']
[04] Words: ['what', 'airline', 'is', 'this', '?']
     Pred: ['WP', 'NN', 'VBZ', 'DT', 'PUNC']
[05] Words: ['show', 'me', 'the', 't', 'w', 'a', 'flight', '.']
     Pred: ['VB', 'PRP', 'DT', 'NNP', 'NNP', 'NN', 'NN', 'PUNC']
[06] Words: ['i', 'would', 'like', 'to', 'travel', 'to', 'westchester', '.']
     Pred: ['PRP', 'MD', 'VB', 'TO', 'VB', 'TO', 'NNP', '.']
[07] Words: ['list', 'american', 'airlines', 'flights', 'from', 'new', 'york', 'newark', 'to', 'nashville', '.']
     Pred: ['VB', 'NNP', 'NNP', 'NNS', 'IN', 'NNP', 'NNP', 'NNP', 'TO', 'NNP', 'PUNC']
[08] Words: ['thanks', '.']
     Pred: ['UH', 'PUNC']
[09] Words: ['what', 'flights', 'are', 'there', 'from', 'nashville', 'to', 'houston', 'tomorrow', 'evening',
'that', 'serve', 'dinner', '?']
     Pred: ['WDT', 'NNS', 'VBP', 'EX', 'IN', 'NNP', 'TO', 'NNP', 'NNP', 'NN', 'DT', 'VB', 'NN',
'PUNC']
[10] Words: ['i', "'d", 'like', 'to', 'fly', 'next', 'friday', '.']
     Pred: ['PRP', 'MD', 'VB', 'TO', 'VB', 'JJ', 'NNP', 'PUNC']
```

a) There are a lot of parts of this that seem to work really well. Most of the words are classified correctly overall. First of all, the numbers and times are right. Plurals vs singular nouns also seem to be pretty solid. Function/filler words (like ‘the’, ‘from’, ‘to’, ‘would’, etc) are also very good. Proper nouns that are made up of multiple words (“New York”, “Los Angeles”, etc) are also turning out pretty well.

In [02], the noun “airport” is marked as a verb. A few verb forms are also off, marked VBP when they should be VB. Things correctly or incorrectly tagged WDT also seem to be messed up sometimes.

b) When words are tagged incorrectly, I think it ends up being because the model has not memorized that pattern yet. I think the more common tags are a better default for the model when it is unsure, so those end up showing up more in incorrect classifications. Overall, more training data or more epochs could help it be able to recognize certain patterns better.

c) I think the micro-errors happen more frequently, like with the verb forms, but are less detrimental than the macro-errors. Macro-errors happen less often but are more of a problem when they do occur.

Question 2

a) 394 unique rules
b) Top 5 most frequent rules:
NNP -> NNP # 860
IN -> IN # 483
PUNC -> PUNC # 469
NN -> NN # 335
NNS -> NNS # 278
c) Top 5 NP rules by probability:
NP -> NNP NNP # 0.1928
NP -> NP NP* # 0.1159
NP -> DT NN # 0.1014
NP -> DT NNS # 0.0855
NP -> DT NP* # 0.0841
d) The most frequent rules are not super surprising to me. Since these word-level rules (like NNP -> NNP) show up once every token, the “most frequent rules” will just be a reflection of which POS tags are the most frequent in the data.

Question 3

CKY with GOLD POS tags (first 10)

```
(TOP (S (NP (DT the) (NN flight)) (VP (MD should) (VP (VB arrive) (VP* (PP (IN at) (NP (CD
eleven) (RB a.m))) (NN tomorrow))))) (PUNC .))
# logprob: -20.4325
(TOP (S (PRP i) (VP (MD would) (VP (VB like) (S_VP (TO to) (VP (VB find) (NP (NP (DT a)
(NN flight)) (SBAR (WDT that) (S_VP (VBZ goes) (VP* (PP (IN from) (NP (NNP la) (NP*
(NNP guardia) (NN airport)))) (PP (TO to) (NP (NNP san) (NNP jose)))))))))))) (PUNC .))
# logprob: -36.8635
(TOP (S_VP (VB show) (VP* (PRP me) (NP (NP (DT the) (NNS flights)) (NP* (PP (IN from)
(NNP newark)) (PP (TO to) (NP (NNP los) (NNP angeles))))))) (PUNC .))
# logprob: -14.0135

(TOP (S_VP (VB show) (VP* (PRP me) (NP (NP (DT the) (NP* (NNP t) (NNP w))) (NP*
(NNP a) (NN flight))))) (PUNC .))
# logprob: -15.2930
(TOP (S (PRP i) (VP (MD would) (VP (VB like) (S_VP (TO to) (VP (VB travel) (PP (TO to)
(NNP westchester))))))) (PUNC .))
# logprob: -13.7500
(TOP (S_VP (VB list) (NP (NP (NNP american) (NP* (NNP airlines) (NNS flights))) (NP* (PP
(IN from) (NP (NNP new) (NP* (NNP york) (NNP newark)))) (PP (TO to) (NNP nashville)))))
(PUNC .))
# logprob: -23.0342
(TOP (UH thanks) (PUNC .))
# logprob: -5.4575
(TOP (SBARQ (WHNP_WHNP (WDT what) (NNS flights)) (SQ (VBP are) (SQ* (EX there)
(SQ* (PP (IN from) (NNP nashville)) (PP (TO to) (NP (NP (NNP houston) (NP* (NN
tomorrow) (NN evening))) (SBAR (WDT that) (S_VP (VBP serve) (NN dinner))))))))) (PUNC
?))
# logprob: -24.7485
(TOP (S (PRP i) (VP (MD 'd) (VP (VB like) (S_VP (TO to) (VP (VB fly) (NP (JJ next) (NNP
friday))))))) (PUNC .))
# logprob: -16.9683
```

CKY with PREDICTED POS tags (first 10)

```
(TOP (S (NP (DT the) (NN flight)) (VP (MD should) (VP (VB arrive) (VP* (PP (IN at) (NP (CD
eleven) (RB a.m))) (NN tomorrow))))) (PUNC .))
# logprob: -20.4325
```

```
(TOP (S_VP (VB show) (VP* (PRP me) (NP (NP (DT the) (NNS flights)) (NP* (PP (IN from)
(NNP newark)) (PP (TO to) (NP (NNP los) (NNP angeles))))))) (PUNC .))
# logprob: -14.0135

(TOP (S_VP (VB show) (VP* (PRP me) (NP (NP (DT the) (NP* (NNP t) (NNP w))) (NP* (NN
a) (NN flight))))) (PUNC .))
# logprob: -13.1728

(TOP (S_VP (VB list) (NP (NP (NNP american) (NP* (NNP airlines) (NNS flights))) (NP* (PP
(IN from) (NP (NNP new) (NP* (NNP york) (NNP newark)))) (PP (TO to) (NNP nashville)))))
(PUNC .))
# logprob: -23.0342
(TOP (UH thanks) (PUNC .))
# logprob: -5.4575

(TOP (S (PRP i) (VP (MD 'd) (VP (VB like) (S_VP (TO to) (VP (VB fly) (NP (JJ next) (NNP
friday))))))) (PUNC .))
# logprob: -16.9683
```

a) There is a slight difference in which sentences it fails to parse given gold tags vs the output from my tagger. For the gold tags, the only one that failed was "What airline is this?", while with the predicted tags there were a few more that failed. This is because of tagger errors that block the necessary rules, like mixups with verb forms and proper nouns.

b) It did well on #1, 3, 6, 7, 8, and 10. These ones use very common patterns which the grammar already has strong counts for, so even if it has a little noise the CKY parser still does a good job with them.

c) One that it parsed awkwardly was #5. It parsed but kind of had an odd structure, with split letters and weird NP* chains. I think this is because sometimes handling proper names can be weird or inconsistencies in general lead it to output a phrase that is technically legal but not very clean.