Question 1

1.b

Stack	Buffer	New dependency	Transition
[ROOT]	[I, parsed, this, sentence, correctly]		Initial configuration
[ROOT, I]	[parsed, this, sentence, correctly]		Shift
[ROOT, I, parsed]	[this, sentence, correctly]		Shift
[ROOT, parsed]	[this, sentence, correctly]	parsed→I	Left-Arc
[ROOT, parsed, this]	[sentence, correctly]		Shift
[ROOT, parsed, this, sentence]	[correctly]		Shift
[ROOT, parsed, sentence]	[correctly]	sentence→this	Left-Arc
[ROOT, parsed]	[correctly]	parsed→sentence	Right-Arc
[ROOT, parsed, correctly]	[]		Shift
[ROOT, parsed]	[]	parsed→correctly	Right-Arc
[ROOT]	[]	ROOT→parsed	Right-Arc

1.c

Parsing will always take 2n steps. At each stage either the buffer size decreases by one or the stack size decreases by one. Each word is removed from the buffer, and then removed from the stack later, and so we have 2n steps.

1.g

Highest dev UAS: 88.37

Test UAS: 89.00

1.h

Question 2

2.a

There are very long outputs since some rules can be recursive with high probability. (And in Chomsky Normal Form each such rule must expand the total output by at least one word).

The reponsible rule is NP -> NP PP, this is a recursive rule that is capable of inflating the sentence.

2.b

All the rules have same weight and Noun has a 5/6 chance to become a terminal, and only a 1/6 chance to create an ADJ. This is the only source for ADJ.

2.c

for the problem in a - decrease the weight of the rule: NP NP PP (or increase NP Det Noun)

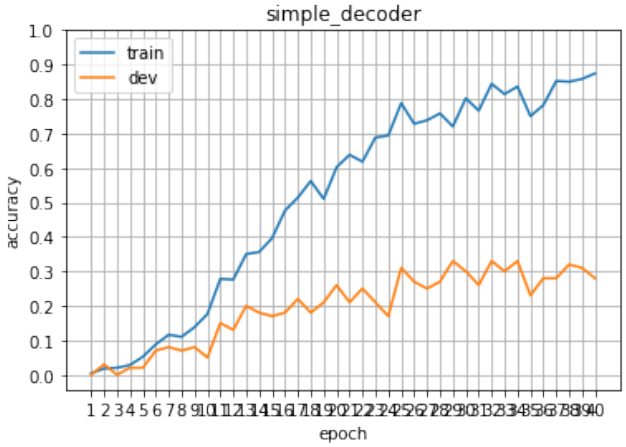
for the problem in b - increase the probability of the rule Noun Adj Noun

2.e

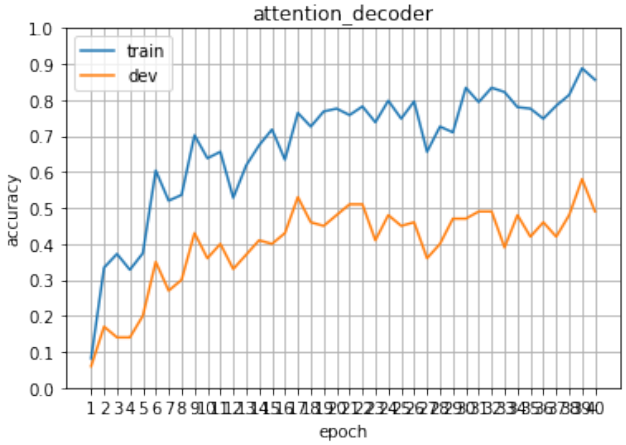
(S (NP (Det the) (Noun president)) (VP (Verb ate) (NP (Det the) (Noun (Adj delicious) (Noun sandwich)))))

Question 3

3.1



3.2



3.3