Name: Yuhan Shao
Student Number: 1002281327
UTORid: shaoyuha

CSC485 Assignment3

1. (a)

stack	buffer	new dependency	transition
[ROOT]	[Nadia, rode, the, old, donkey, with, dexterity]		Initial Config
[ROOT, Nadia]	[rode, the, old, donkey, with, dexterity]		SHIFT
[ROOT, Nadia, rode]	[the, old, donkey, with, dexterity]		SHIFT
[ROOT, rode]	[the, old, donkey, with, dexterity]	rode $\xrightarrow{\text{nsubj}}$ Nadia	LEFT-ARC
[ROOT, rode, the]	[old, donkey, with, dexterity]		SHIFT
[ROOT, rode, the, old]	[donkey, with, dexterity]		SHIFT
[ROOT, rode, the, old, donkey]	[with, dexterity]		SHIFT
[ROOT, rode, the, donkey]	[with, dexterity]	donkey $\xrightarrow{\text{amod}}$ old	LEFT-ARC
[ROOT, rode, donkey]	[with, dexterity]	Donkey $\xrightarrow{\text{det}}$ the	LEFT-ARC
[ROOT, rode]	[with, dexterity]	rode$\xrightarrow{\text{dobj}}$ donkey	RIGHT-ARC
[ROOT, rode, with]	[dexterity]		SHIFT
[ROOT, rode, with, dexterity]			SHIFT
[ROOT, rode, with]		with $\xrightarrow{\text{pobj}}$ dexterity	RIGHT-ARC
[ROOT, rode]		rode $\xrightarrow{\text{prep}}$ with	RIGHT-ARC
[ROOT]		ROOT $\xrightarrow{\text{pred}}$rode	RIGHT-ARC

(b)

Answer: $2n$ steps.

One step to shift into stack and one step to popped from stack with LEFT-ARC or RIGHT-ARC for each word.

(c)

In the described parsing, when we add left or right edge between stack[-2] and stack[-1], we removed all words between stack[-2] and stack[-1]. Thus we can't have two edges crossing each other.

For example, if we first add edge saw -> yesterday, a dog is already removed and no way to add edge a dog -> was. Or if we first add edge a dog -> was, then yesterday is already removed and not way to add edge saw -> yesterday.

2 (b)

I have finished all the functions in model.py. The most difficult part I think is to add embedding function. I made a mistake at first. I don’t create the tensorflow variables and the performance is very poor. After debugging, I realize that if I don’t create the variables, then the training can’t optimize the embedding. After correcting this, my performance improves dramatically and achieves the desired performance.

Loss at last epoch: 0.1038

Validation on dev set LAS: 0.88

Validation on dev set UAS: 0.90

Test on test set LAS: 0.88

Test on test set UAS: 0.90