Class- CS 6957
Mini Project -2
Name - Trupti Mohanty

1. Implemented the model and the parser. Submitted the result.txt file for the hidden data.

   Best Model Details
   Glove 840B, 300d, Concatenate
   Best learning rate: 0.001
   UAS: 0.73
   LAS: 0.68

2.

Embeddings	Mean		Concatenate	
	UAS	LAS	UAS	LAS
Glove 6B 50d	0.37	0.30	0.70	0.64
Glove 6B 300d	0.42	0.37	0.71	0.65
Glove 42B 300d	0.44	0.38	0.71	0.66
Glove 840B 300d	0.45	0.39	**0.73**	**0.68**

3. The results indicate a significant improvement in accuracy, approximately 75%, when concatenating word embeddings (as well as POS tags) compared to using the mean of these embeddings. This suggests that the order of words plays a crucial role in enhancing the accuracy. In the case of taking the mean, the model does not consider the order of words, which results in a lower level of accuracy.

   While using different Glove word embeddings does not lead significant changes in concanate models' accuracy. However, with mean, changes have been noticed while using different dimension. 300 dimension works better.

4.

(a) Marry had a little lamb.

```
[['SHIFT', 'SHIFT', 'SHIFT', 'SHIFT', 'SHIFT', 'REDUCE_L_amod', 'REDUCE_L_det',
'REDUCE_L_aux', 'REDUCE_R_parataxis', 'SHIFT', 'REDUCE_R_punct']]
```

Stack	Buffer	Actions
[Null Null]	[Marry had a little lamb. Null Null]	'SHIFT'
[Null Null Marry]	[had a little lamb. Null Null]	'SHIFT'
[Null Null Marry had]	[a little lamb. Null Null]	'SHIFT'
[Null Null Marry had a]	[little lamb. Null Null]	'SHIFT'
[Null Null Marry had a little]	[lamb. Null Null]	'SHIFT',
[Null Null Marry had a little lamb]	[. Null Null]	'REDUCE_L_amod',
[Null Null Marry had a lamb]	[. Null Null]	'REDUCE_L_det',
[Null Null Marry had lamb]	[. Null Null]	'REDUCE_L_aux'
[Null Null Marry lamb]	[. Null Null]	'REDUCE_R_parataxis'
[Null Null Marry]	[. Null Null]	'SHIFT'
[Null Null Marry .]	[Null Null]	'REDUCE_R_punct'
[Null Null Marry]	[Null Null]	

(b) I ate the fish raw .

```
[['SHIFT', 'SHIFT', 'REDUCE_L_nsubj', 'SHIFT', 'SHIFT', 'REDUCE_L_det',
'REDUCE_R_obj', 'SHIFT', 'REDUCE_R_ccomp', 'SHIFT', 'REDUCE_R_punct']]
```

Stack	**Buffer**	**Actions**
[Null Null]	[I ate the fish raw . Null Null]	'SHIFT'
[Null Null I]	[ate the fish raw . Null Null]	'SHIFT'
[Null Null I ate]	[the fish raw . Null Null]	'REDUCE_L_nsubj'
[Null Null ate]	[the fish raw . Null Null]	'SHIFT'
[Null Null ate the]	[fish raw . Null Null]	'SHIFT'
[Null Null ate the fish]	[raw . Null Null]	'REDUCE_L_det'
[Null Null ate fish]	[raw . Null Null]	'REDUCE_R_obj'
[Null Null ate]	[raw . Null Null]	'SHIFT'
[Null Null ate raw]	[. Null Null]	'REDUCE_R_ccomp'
[Null Null ate]	[. Null Null]	'SHIFT'
[Null Null ate .]	[Null Null]	'REDUCE_R_punct'
[Null Null ate]	[Null Null]	

(c) With neural networks , I love solving problems .

```
[['SHIFT', 'SHIFT', 'SHIFT', 'REDUCE_L_amod', 'REDUCE_L_case', 'SHIFT',
'REDUCE_R_punct', 'SHIFT', 'SHIFT', 'REDUCE_L_nsubj', 'REDUCE_L_nsubj', 'SHIFT',
'SHIFT', 'REDUCE_R_obj', 'REDUCE_R_ccomp', 'SHIFT', 'REDUCE_R_punct']]
```

Stack	Buffer	Actions
[Null Null]	[with neural networks , I love solving problems . Null Null]	'SHIFT'
[Null Null with]	[neural networks , I love solving problems .Null Null]	'SHIFT'
[Null Null with neural]	[networks , I love solving problems . Null Null]	'SHIFT'
[Null Null with neural networks]	[, I love solving problems . Null Null]	'REDUCE_L_amod'
[Null Null with networks]	[, I love solving problems . Null Null]	'REDUCE_L_case'
[Null Null networks]	[, I love solving problems .]	'SHIFT'
[Null Null networks ,]	[I love solving problems .]	'REDUCE_R_punct'
[Null Null networks]	[I love solving problems .]	'SHIFT'
[Null Null networks I]	[love solving problems .]	'SHIFT',
[Null Null networks I love]	[solving problems .]	'REDUCE_L_nsubj',
[Null Null networks love]	[solving problems .]	'REDUCE_L_nsubj'
[Null Null love]	[solving problems .]	'SHIFT'
[Null Null love solving]	[problems .]	'SHIFT'
[Null Null love solving problems]	[. Null]	'REDUCE_R_obj'
[Null Null love solving]	[. Null]	'REDUCE_R_ccomp'
[Null Null love]	[. Null]	'SHIFT'
[Null Null love .]	[Null]	'REDUCE_R_punct'
[Null Null love]		

5. A Fast and Accurate Dependency Parser using Neural Networks

1. **Homework** - Word embeddings and POS tags as input layer.
   Paper – Along with word embedding and POS tags, arc labels have also been considered in the input layer.

2. **Homework** - One hidden layer with relu activation followed by softmax
   Paper - Input layer is mapped to hidden layer by a cube activation function to better capture the interaction between three embeddings (word/POS/labels) and a softmax layer.

3. **Homework** – Used 2 words from the stack and 2 words from the buffer and their corresponding POS tags.
   Paper - Used top 3 words from stack and top 3 from buffer, the first and second left most and right most children of two words on the stack. The leftmost of leftmost and rightmost of rightmost children for the two words on stack. Total 18 elements.
   POS tags for all these 18, and the arc labels of 12, excluding for 6 words (stack and buffer)

4. **Homework** – Only Cross entropy loss no regularization
   Paper - Cross entropy loss + regularization, also used drop out with 0.5 rate

5. **Homework** – Different Glove embedding.
   Paper - Pre trained Embeddings from Collobert et al, 130M dictionary

6. **Pape**r – Pre-computed matrix multiplication for faster implementation.