

# COMS 4705 NLP

## Sec 01

### HW 3

Name:	Yu Wang
Uni:	yw3025
Date:	April. 23, 2018

## ***Instructions:***

### **To train the model:**

Note: If you want to retrain my network, the Q3's network is in network\_3.py file, Q1 and Q2's network is in network.py file. The default setting in driver.py file is Q3's network, to change back to Q1 and Q2's network, make sure to ``from network import Network`` instead of ``from network_3 import Network`` in driver.py. I have spent days for this problem that sample code in `dynet` save weights instead of model.

```
python2 src/driver.py --model=./model/model_q1 --vocab=./data/vocabs_q1.data
```

```
python2 src/driver.py --model=./model/model_q2 --vocab=./data/vocabs_q2.data --hidden1=400 -  
-hidden2=400
```

```
python2 src/driver.py --model=./model/model_q3 --vocab=./data/vocabs_q3.data --hidden1=400 -  
-hidden2=400 --we=100 --pre_trained="./glove.6B.100d.txt" (make sure to download this file  
beforehand)
```

### **To validate the model (based on Canvas Instructions):**

Note: I put output file in the outputs folder as in Canvas.

Q1:

```
python2 src/depModel1.py trees/dev.conll outputs/dev_part1.conll
```

```
python2 src/eval.py trees/dev.conll outputs/dev_part1.conll
```

Q2:

```
python2 src/depModel2.py trees/dev.conll outputs/dev_part2.conll
```

```
python2 src/eval.py trees/dev.conll outputs/dev_part2.conll
```

Q3:

```
python2 src/depModel3.py trees/dev.conll outputs/dev_part3.conll
```

```
python2 src/eval.py trees/dev.conll outputs/dev_part3.conll
```

### **To test the model (based on hw3 Instructions) :**

Note: I put output file in the outputs folder as in Canvas.

Q1:

```
python2 src/depModel1.py trees/test.conll outputs/test_part1.conll
```

Q2:

```
python2 src/depModel2.py trees/test.conll outputs/test_part2.conll
```

Q3:

```
python2 src/depModel3.py trees/test.conll outputs/test_part3.conll
```

## *Result and Observation*

### Q1

```
yuwang nlp_hw3 $ python2 src/eval.py trees/dev.conll outputs/dev_part1.conll
Unlabeled attachment score 83.18
Labeled attachment score 79.95
```

This is the baseline as instructed in the HW3 instructions

### Q2

```
yuwang nlp_hw3 $ python2 src/eval.py trees/dev.conll outputs/dev_part2.conll
Unlabeled attachment score 83.72
Labeled attachment score 80.47
```

We get improvement in both unlabelled attachment and labelled attachment. This is because we have more complicated models (more hidden layer dimensions) and more parameters so that the model can capture more information.

Note the training time increase a lot (\*2) with respect to Q1

### Q3

```
yuwang nlp_hw3 $ python2 src/eval.py trees/dev.conll outputs/dev_part3.conll
Unlabeled attachment score 85.82
Labeled attachment score 82.76
```

Network for this part is in network\_3.py

For this part I use the **per-trained vector** `glove.6B.100d.txt` all other settings are the same as Q2 and set the word embedding dimension to 100.

I choose this option because:

1. I find a little bit overfitting in Q2, since we double the dimensions of the two hidden layers and only minimal improvement. Also, mini-batch loss in Q2 already goes down to 0.0\* in 5's epochs.
2. I first use dropout, and the result don't show much improvement (Unlabeled attachment score 83.67, Labeled attachment score 80.59)
3. So I think the reason is that, the information in the training set is not that universal to give improvement on the dev set. The pre-trained vectors, however, have information on a much broader set apart from the training set, can give us more information to capture between the words and the sentence context, so we get much improvement at last.