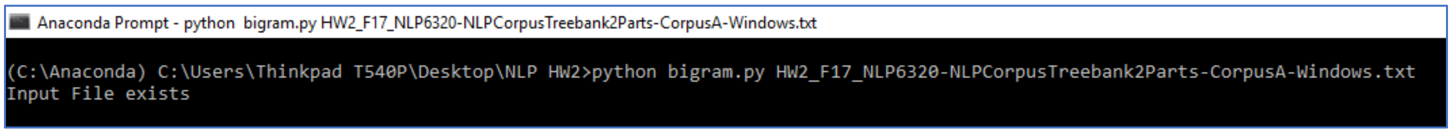
READ ME NLP HW2 adj160230

**Question #1**

**Command:** python biagram.py <input-file>

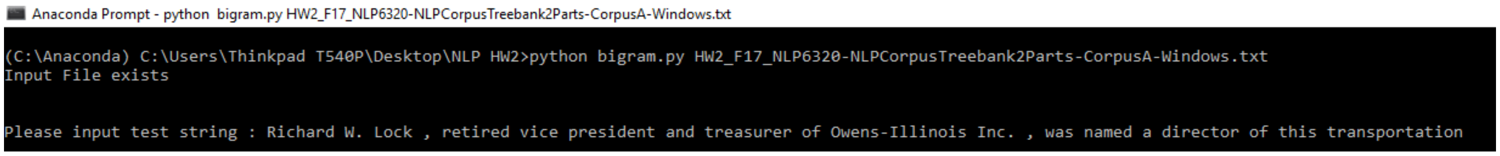
1. python bigram.py HW2_F17_NLP6320-NLPCorpusTreebank2Parts-CorpusA-Windows.txt



```
Anaconda Prompt - python  bigram.py HW2_F17_NLP6320-NLPCorpusTreebank2Parts-CorpusA-Windows.txt

(C:\Anaconda) C:\Users\Thinkpad T540P\Desktop\NLP HW2>python bigram.py HW2_F17_NLP6320-NLPCorpusTreebank2Parts-CorpusA-Windows.txt
Input File exists
```

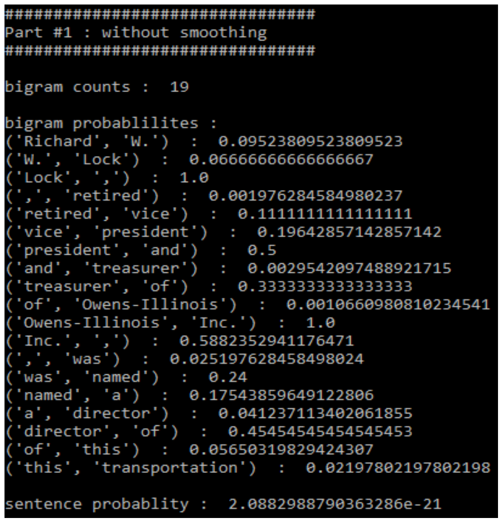
2. Enter input test string

   Example: “Richard W. Lock , retired vice president and treasurer of Owens-Illinois Inc. , was named a director of this transportation”

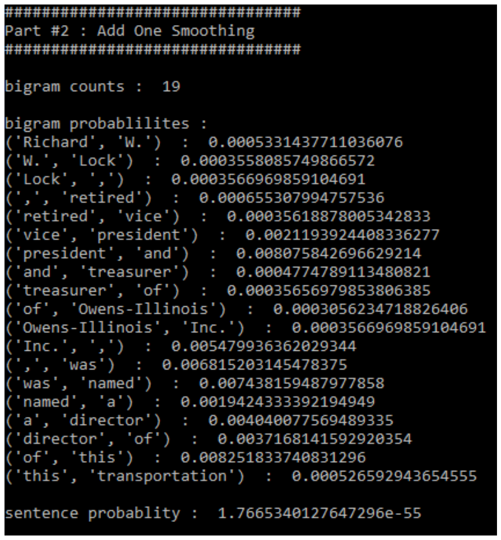


```
Anaconda Prompt - python  bigram.py HW2_F17_NLP6320-NLPCorpusTreebank2Parts-CorpusA-Windows.txt

(C:\Anaconda) C:\Users\Thinkpad T540P\Desktop\NLP HW2>python bigram.py HW2_F17_NLP6320-NLPCorpusTreebank2Parts-CorpusA-Windows.txt
Input File exists

Please input test string : Richard W. Lock , retired vice president and treasurer of Owens-Illinois Inc. , was named a director of this transportation
```

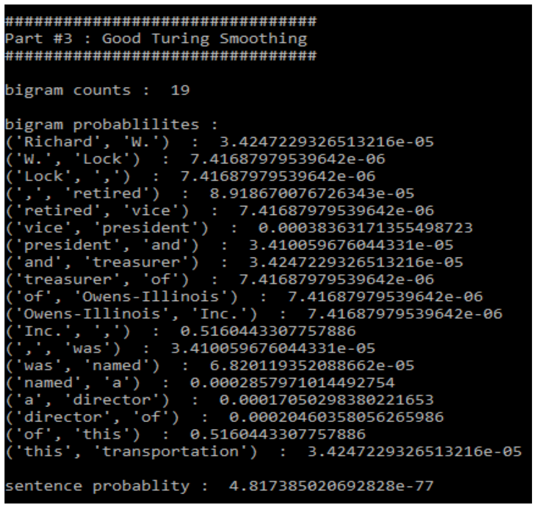
3. Output will be displayed as follows:



```
################################
Part #1 : without smoothing
################################

bigram counts :  19

bigram probablilites :
('Richard', 'W.')  :  0.09523809523809523
('W.', 'Lock')  :  0.06666666666666667
('Lock', ',')  :  1.0
(',', 'retired')  :  0.001976284584980237
('retired', 'vice')  :  0.1111111111111111
('vice', 'president')  :  0.19642857142857142
('president', 'and')  :  0.5
('and', 'treasurer')  :  0.0029542097488921715
('treasurer', 'of')  :  0.3333333333333333
('of', 'Owens-Illinois')  :  0.0010660980810234541
('Owens-Illinois', 'Inc.')  :  1.0
('Inc.', ',')  :  0.5882352941176471
(',', 'was')  :  0.025197628458498024
('was', 'named')  :  0.24
('named', 'a')  :  0.17543859649122806
('a', 'director')  :  0.041237113402061855
('director', 'of')  :  0.45454545454545453
('of', 'this')  :  0.05650319829424307
('this', 'transportation')  :  0.02197802197802198

sentence probablity :  2.0882988790363286e-21
```



```
################################
Part #2 : Add One Smoothing
################################

bigram counts :  19

bigram probablilites :
('Richard', 'W.')  :  0.0005331437711036076
('W.', 'Lock')  :  0.0003558085749866572
('Lock', ',')  :  0.0003566969859104691
(',', 'retired')  :  0.000655307994757536
('retired', 'vice')  :  0.0003561887800534283
('vice', 'president')  :  0.002119392440833627
('president', 'and')  :  0.008075842696629214
('and', 'treasurer')  :  0.0004774789113480821
('treasurer', 'of')  :  0.0003565697985380638
('of', 'Owens-Illinois')  :  0.0003056234718826406
('Owens-Illinois', 'Inc.')  :  0.0003566969859104691
('Inc.', ',')  :  0.005479936362029344
(',', 'was')  :  0.006815203145478375
('was', 'named')  :  0.007438159487977858
('named', 'a')  :  0.0019424333392194949
('a', 'director')  :  0.004040077569489335
('director', 'of')  :  0.0037168141592920354
('of', 'this')  :  0.008251833740831296
('this', 'transportation')  :  0.000526592943654555

sentence probablity :  1.7665340127647296e-55
```



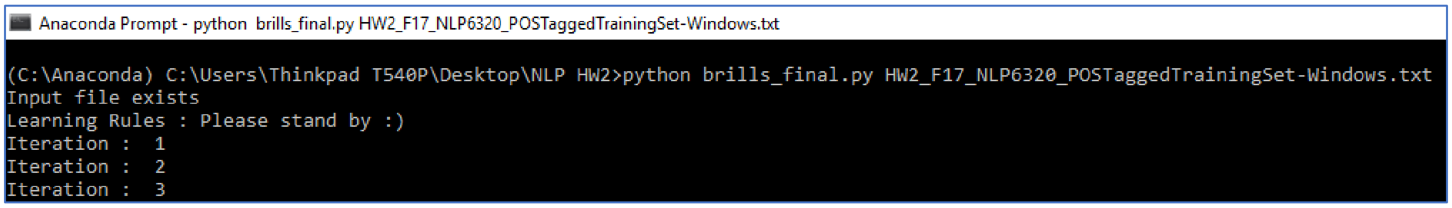
```
################################
Part #3 : Good Turing Smoothing
################################

bigram counts :  19

bigram probablilites :
('Richard', 'W.')  :  3.4247229326513216e-05
('W.', 'Lock')  :  7.41687979539642e-06
('Lock', ',')  :  7.41687979539642e-06
(',', 'retired')  :  8.918670076726343e-05
('retired', 'vice')  :  7.41687979539642e-06
('vice', 'president')  :  0.00038363171355498723
('president', 'and')  :  3.410059676044331e-05
('and', 'treasurer')  :  3.4247229326513216e-05
('treasurer', 'of')  :  7.41687979539642e-06
('of', 'Owens-Illinois')  :  7.41687979539642e-06
('Owens-Illinois', 'Inc.')  :  7.41687979539642e-06
('Inc.', ',')  :  0.5160443307757886
(',', 'was')  :  3.410059676044331e-05
('was', 'named')  :  6.820119352088662e-05
('named', 'a')  :  0.0002857971014492754
('a', 'director')  :  0.00017050298380221653
('director', 'of')  :  0.00020460358056265986
('of', 'this')  :  0.5160443307757886
('this', 'transportation')  :  3.4247229326513216e-05

sentence probablity :  4.817385020692828e-77
```

4. Output files generated:
   - adj160230_bigram_probablities.txt (gives bigram probabilities without smoothing)
   - adj160230_bigram_probablities_addOne.txt (gives bigram probabilities with add one smoothing)
   - adj160230_bigram_probablities_goodTurning.txt (gives bigram probabilities with good)
   - adj160230_outputPart_1 (contains bigram counts , sentence counts)

READ ME NLP HW2 adj160230
**Question #2:**

**1. Command:** python brills_final.py <input-file>

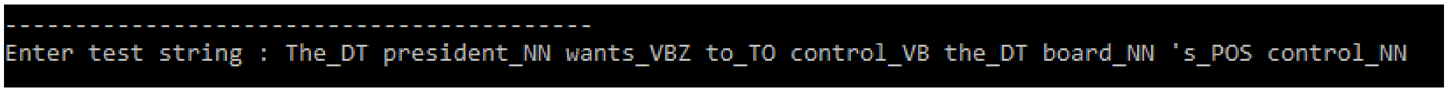
***python brills_final.py HW2_F17_NLP6320_POSTaggedTrainingSet-Windows.txt***



```
Anaconda Prompt - python  brills_final.py HW2_F17_NLP6320_POSTaggedTrainingSet-Windows.txt

(C:\Anaconda) C:\Users\Thinkpad T540P\Desktop\NLP HW2>python brills_final.py HW2_F17_NLP6320_POSTaggedTrainingSet-Windows.txt
Input file exists
Learning Rules : Please stand by :)
Iteration :  1
Iteration :  2
Iteration :  3
```

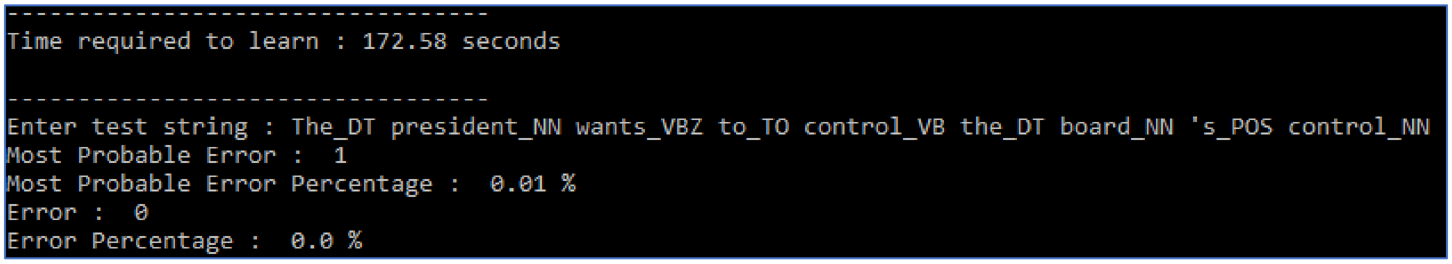
2. Enter input test string (POS tagged senetence)

Example: “*The*_DT *president*_NN *wants*_VBZ *to*_TO *control*_VB *the*_DT *board*_NN *'s*_POS *control*_NN”



```
--------------------------------------------
Enter test string : The_DT president_NN wants_VBZ to_TO control_VB the_DT board_NN 's_POS control_NN
```

3. output:



```
----------------------------------
Time required to learn : 172.58 seconds

----------------------------------
Enter test string : The_DT president_NN wants_VBZ to_TO control_VB the_DT board_NN 's_POS control_NN
Most Probable Error :  1
Most Probable Error Percentage :  0.01 %
Error :  0
Error Percentage :  0.0 %
```

4. output files:

- adj160230_brills_output.txt