a. Create and use your own corpora.

Code and Output:

```
import nltk
from nltk.corpus import PlaintextCorpusReader
corpus_root = 'D:\MSc.IT'
wordlists = PlaintextCorpusReader(corpus_root, '.*')
wordlists.fileids()
```

```
['ca01', 'ca02'  ['test14826', '  ['test14832', '  ['MScIT-NEP
File  Edit  View
(1).pptx', 'sem_4/DL/Overflow_Underflow_DeepLearning (1).pptx', 'sem_4/DL/PDF-of-Deep-
Learning.pdf', 'sem_4/DL/Poor_Conditioning_DeepLearning (1).pptx', 'sem_
4/NLP/FSA_FST.pdf', 'sem_4/NLP/Handbook Of Natural Language Processing, Second Edition
Chapman & Hall Crc Machine Learning & Pattern Recognition 2010.pdf', 'sem_
4/NLP/NLP_CE-2/4916 - 4924_Covington algorithm Explaination and usage.pptx', 'sem_
4/NLP/NLP_content_final_Syllabus Mapping.pdf', 'sem_4/NLP/NLP_practical/NLP.zip',
'sem_4/NLP/NLP_practical/NLP_prac_1.docx', 'sem_4/NLP/NLP_practical/NLP_prac_1.pdf',
'sem_4/NLP/NLP_practical/NLP_prac_2.docx', 'sem_4/NLP/NLP_practical/NLP_prac_2.pdf',
'sem_4/NLP/NLP_practical/NLP_prac_3.docx', 'sem_4/NLP/NLP_practical/NLP_prac_3.pdf',
'sem_4/NLP/NLP_practical/NLP_prac_codes/exam.mp3', 'sem_
4/NLP/NLP_practical/NLP_prac_codes/female.wav', 'sem_
4/NLP/NLP_practical/NLP_prac_codes/funny.txt', 'sem_
4/NLP/NLP_practical/NLP_prac_codes/nlp_code_p1_b.py', 'sem_
4/NLP/NLP_practical/NLP_prac_codes/nlp_code_p1_c.py', 'sem_
4/NLP/NLP_practical/NLP_prac_codes/nlp_code_p2.py', 'sem_
4/NLP/NLP_practical/NLP_prac_codes/nlp_code_p3.py', 'sem_4/NLP/NLP_practical/~$P_prac_
3.docx', 'sem_4/NLP/NLP_practical/~WRL2455.tmp', 'sem_
4/NLP/Speech_and_language_processing.pdf', 'sem_4/NLP/Unit 1- Inroduction To NLP.pdf',
'sem_4/_research/model_flowchart.zip', 'sem_4/_research/results/water-2_results.docx',
'sem_4/_research/results/water-2_results.pdf', 'sem_4/
_research/results/water_boost.docx', 'sem_4/_research/results/water_boost.pdf', 'sem_
4/_research/synopsis/4913-4916_WQA_synopsis.pdf', 'sem_4/_research/synopsis/water_2
_synopsis.docx', 'sem_4/_research/unused/model_flowchart.zip', 'sem_4/
_research/unused/stack_review_1.pdf', 'sem_4/_research/vote_model/model_flowchart
(1).png', 'sem_4/_research/water2_paper.docx', 'sem_4/_research/water2_paper.pdf',
'sem_4/_research/water_vote_paper.docx', 'sem_4/_research/water_vote_paper.pdf',
'tracker-fitbit-luxe-377882341-t3ajw.avif', 'write_research_paper.docx']
Ln 1, Col 332255  332,254 characters  100%  Windows (CRLF)  UTF-8
```

```
wordlists.words("funny.txt")
```

```
['I', 'tried', 'to', 'clean', 'my', 'room', ',', 'but', ...]
```

b. Study of tagged corpora with methods like tagged_sents, tagged_words.

Code and Output:

import nltk

nltk.corpus.brown.tagged_words()

```
[('The', 'AT'), ('Fulton', 'NP-TL'), ...]
```

nltk.corpus.brown.tagged_sents()

```
[[('The', 'AT'), ('Fulton', 'NP-TL'), ('County', 'NN-TL'), ('G
rand', 'JJ-TL'), ('Jury', 'NN-TL'), ('said', 'VBD'), ('Friday'
, 'NR'), ('an', 'AT'), ('investigation', 'NN'), ('of', 'IN'),
("Atlanta's", 'NP$'), ('recent', 'JJ'), ('primary', 'NN'), ('e
lection', 'NN'), ('produced', 'VBD'), ('``', '``'), ('no', 'AT
'), ('evidence', 'NN'), ("''", "''"), ('that', 'CS'), ('any',
'DTI'), ('irregularities', 'NNS'), ('took', 'VBD'), ('place',
'NN'), ('.', '.')], [('The', 'AT'), ('jury', 'NN'), ('further'
, 'RBR'), ('said', 'VBD'), ('in', 'IN'), ('term-end', 'NN'), (
'presentments', 'NNS'), ('that', 'CS'), ('the', 'AT'), ('City'
, 'NN-TL'), ('Executive', 'JJ-TL'), ('Committee', 'NN-TL'), ('
,', ','), ('which', 'WDT'), ('had', 'HVD'), ('over-all', 'JJ')
, ('charge', 'NN'), ('of', 'IN'), ('the', 'AT'), ('election',
'NN'), (',', ','), ('``', '``'), ('deserves', 'VBZ'), ('the',
'AT'), ('praise', 'NN'), ('and', 'CC'), ('thanks', 'NNS'), ('o
f', 'IN'), ('the', 'AT'), ('City', 'NN-TL'), ('of', 'IN-TL'),
('Atlanta', 'NP-TL'), ("''", "''"), ('for', 'IN'), ('the', 'AT
'), ('manner', 'NN'), ('in', 'IN'), ('which', 'WDT'), ('the',
'AT'), ('election', 'NN'), ('was', 'BEDZ'), ('conducted', 'VBN
'), ('.', '.')], ...]
```

nltk.corpus.conll2000.tagged_words()

```
[('Confidence', 'NN'), ('in', 'IN'), ('the', 'DT'), ...]
```

nltk.corpus.conll2000.tagged_sents()

```
[[('Confidence', 'NN'), ('in', 'IN'), ('the', 'DT'), ('pound',
'NN'), ('is', 'VBZ'), ('widely', 'RB'), ('expected', 'VBN'), (
'to', 'TO'), ('take', 'VB'), ('another', 'DT'), ('sharp', 'JJ'
), ('dive', 'NN'), ('if', 'IN'), ('trade', 'NN'), ('figures',
'NNS'), ('for', 'IN'), ('September', 'NNP'), (',', ','), ('due
', 'JJ'), ('for', 'IN'), ('release', 'NN'), ('tomorrow', 'NN')
, (',', ','), ('fail', 'VB'), ('to', 'TO'), ('show', 'VB'), ('
a', 'DT'), ('substantial', 'JJ'), ('improvement', 'NN'), ('fro
m', 'IN'), ('July', 'NNP'), ('and', 'CC'), ('August', 'NNP'),
("'s", 'POS'), ('near-record', 'JJ'), ('deficits', 'NNS'), ('.
', '.')], [('Chancellor', 'NNP'), ('of', 'IN'), ('the', 'DT'),
('Exchequer', 'NNP'), ('Nigel', 'NNP'), ('Lawson', 'NNP'), ("'
s", 'POS'), ('restated', 'VBN'), ('commitment', 'NN'), ('to',
'TO'), ('a', 'DT'), ('firm', 'NN'), ('monetary', 'JJ'), ('poli
cy', 'NN'), ('has', 'VBZ'), ('helped', 'VBN'), ('to', 'TO'), (
'prevent', 'VB'), ('a', 'DT'), ('freefall', 'NN'), ('in', 'IN'
), ('sterling', 'NN'), ('over', 'IN'), ('the', 'DT'), ('past',
'JJ'), ('week', 'NN'), ('.', '.')], ...]
```

nltk.corpus.treebank.tagged_words()

```
[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ...]
```

nltk.corpus.treebank.tagged_sents()

```
[[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ('61', 'CD
'), ('years', 'NNS'), ('old', 'JJ'), (',', ','), ('will', 'MD'
), ('join', 'VB'), ('the', 'DT'), ('board', 'NN'), ('as', 'IN'
), ('a', 'DT'), ('nonexecutive', 'JJ'), ('director', 'NN'), ('
Nov.', 'NNP'), ('29', 'CD'), ('.', '.')], [('Mr.', 'NNP'), ('V
inken', 'NNP'), ('is', 'VBZ'), ('chairman', 'NN'), ('of', 'IN'
), ('Elsevier', 'NNP'), ('N.V.', 'NNP'), (',', ','), ('the', '
DT'), ('Dutch', 'NNP'), ('publishing', 'VBG'), ('group', 'NN')
, ('.', '.')], ...]
```

c. Write a program to find the most frequent noun tags.

Code and Output:

wsj = nltk.corpus.treebank.tagged_words()

word_tag_fd = nltk.FreqDist(wsj)

[word + "/" + tag for (word, tag) in word_tag_fd if tag.startswith('N')]

```
['crook/NN', 'cold/NN', 'revelations/NNS', 'writing/NN', 'scams/NNS', 'ingenuity/NN',
'Auditors/NNS', 'crookery/NN', 'garden-variety/NN', 'lifes/NNS', 'Mercedes/NNPS',
'clothes/NNS', 'Irving/NNP', 'Lobsenz/NNP', 'pediatrician/NN', 'Rusty/NNP',
'gambler/NN', 'blackjack/NN', 'tidbits/NNS', 'root/NN', 'auspices/NNS',
'Programs/NNS', 'thieves/NNS', 'scandals/NNS', 'HUD/NNP', 'characteristics/NNS',
'insider/NN', 'bloc/NN', 'Nomenklatura/NN', 'Stern/NNP', 'Urban/NNP', 'Everything/NN',
'Christian/NNP', 'Andersson/NNP', 'receivers/NNS', 'Subcontractors/NNS', 'swift/NN',
'shipyards/NNS', 'Customers/NNS', 'Lines/NNPS', 'Helsinki/NNP', 'Norfolk/NNP',
'Purina/NNP', 'seafood/NN', 'phase-out/NN', 'Hostess/NNP', 'bakery/NN', 'cadmium/NN',
'carbon/NN', 'zinc/NN', 'ingredients/NNS', 'cereal/NN', 'Continental/NNP',
'Baking/NNP', 'bread/NN', 'Eveready/NNP', 'Companies/NNS', 'percent/NN',
'Ravenswood/NNP', 'binge/NN', 'bottlenecks/NNS', 'pipeline/NN', 'trains/NNS',
'River/NN', 'Railroad/NN', 'windfall/NN', 'strain/NN', 'Dunton/NNP', 'Plains/NNS',
'tows/NNS', 'reservoirs/NNS', 'Barge/NN', 'Corn/NN', 'Lyle/NNP', 'Railroad/NNP',
'Waterloo/NNP', 'one-fifth/NN', 'Biedermann/NNP', 'Allendale/NNP', 'train/NN',
'port/NN', 'Ports/NNS', 'Lakes/NNPS', 'Coast/NNP', 'ENERGY/NN', 'refinery/NN',
'Heating/NN', 'sell-off/NN', 'PRECIOUS/NNP', 'METALS/NNPS', 'silver/NN',
"O'Neill/NNP", 'Elders/NNPS', 'equities/NNS', 'warehouses/NNS', 'miners/NNS',
'Bronces/NNP', 'Soldado/NNP', 'Minera/NNP', 'Disputada/NNP', 'Reasons/NNS',
'walkout/NN', 'mood/NN', 'readings/NNS', 'NCR/NNP', 'networking/NN', 'hub/NN',
'Novell/NNP', 'NetWare/NNP', 'chunks/NNS', 'Marcus/NNP', 'appliance/NN', 'Lorain/NNP',
'joint/NN', 'Kobe/NNP', 'Bethlehem/NNP', 'Inland/NNP', 'projections/NNS',
'Bradford/NNP', 'pipe/NN', 'Marathon/NNP', 'red/NN', 'TXO/NNP', 'Proceeds/NNS',
'Barrett/NNP', 'Leon/NNP', 'Level/NNP', 'McFarlan/NNP', 'DiLoreto/NNP',
'container/NN', 'Delmont/NNP', 'constitutional-law/NN', 'Professors/NNP',
'Laurence/NNP', 'ends/NNS', 'spectrum/NN', 'liberal/NN', 'Civil/NNP', 'railcar/NN',
'platforms/NNS', 'Trailer/NNP', 'Train/NNP']
```

Ln 1, Col 99299 | 99,298 characters | 100% | Windows (CRLF) | UTF-8