Name : Arul Kumar ARK

Roll No. : 225229103

Lab : 9

```
                                        Building Bigram Tagger
```

Ex : 1

In [42]:
```
import nltk
```

In [43]:
```
from nltk.tokenize import sent_tokenize,word_tokenize
```

In [44]:
```
import nltk
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\1mscdsa18\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[44]:
```
True
```

In [45]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\1mscdsa18\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[45]:
```
True
```

In [46]:
```
text = word_tokenize("And now for something completely different")
nltk.pos_tag(text)
```

Out[46]:
```
[('And', 'CC'),
 ('now', 'RB'),
 ('for', 'IN'),
 ('something', 'NN'),
 ('completely', 'RB'),
 ('different', 'JJ')]
```

```
CC : coordinating conjunction
RB : dverb (occasionally, swiftly)
IN : preposition/subordinating conjunction
NN : noun, singular (cat, tree)
RB : adverb (occasionally, swiftly)
JJ : This NLTK POS Tag is an adjective (large)
```

Ex : 2

In [47]:
```
from nltk.corpus import brown
```

In [48]:
```
nltk.download('brown')
```

```
[nltk_data] Downloading package brown to
[nltk_data]     C:\Users\1mscdsa18\AppData\Roaming\nltk_data...
[nltk_data]   Package brown is already up-to-date!
```

Out[48]:
```
True
```

Step : 1

In [49]:
```
tagsen = brown.tagged_sents()
tagsen
```

Out[49]:
```
[[('The', 'AT'), ('Fulton', 'NP-TL'), ('County', 'NN-TL'), ('Grand', 'JJ-TL'), ('Jury', 'NN-TL'), ('said', 'VBD'), ('Friday',
'NR'), ('an', 'AT'), ('investigation', 'NN'), ('of', 'IN'), ("Atlanta's", 'NP$'), ('recent', 'JJ'), ('primary', 'NN'), ('electi
on', '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', 'N
N'), ('of', 'IN'), ('the', 'AT'), ('election', 'NN'), (',', ','), ('``', '``'), ('deserves', 'VBZ'), ('the', 'AT'), ('praise',
'NN'), ('and', 'CC'), ('thanks', 'NNS'), ('of', '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', 'N
N'), ('was', 'BEDZ'), ('conducted', 'VBN'), ('.', '.')], ...]
```

In [50]:
```
len(tagsen)
```

Out[50]:
```
57340
```

Step : 2

In [51]:
```
br_train = tagsen[0:50000]
br_test = tagsen[50000:]
br_test[0]
```

Out[51]:
```
[('I', 'PPSS'),
 ('was', 'BEDZ'),
 ('loaded', 'VBN'),
 ('with', 'IN'),
 ('suds', 'NNS'),
 ('when', 'WRB'),
 ('I', 'PPSS'),
 ('ran', 'VBD'),
 ('away', 'RB'),
 (',', ','),
 ('and', 'CC'),
 ('I', 'PPSS'),
 ("haven't", 'HV*'),
 ('had', 'HVN'),
 ('a', 'AT'),
 ('chance', 'NN'),
 ('to', 'TO'),
 ('wash', 'VB'),
 ('it', 'PPO'),
 ('off', 'RP'),
 ('.', '.')]
```

Step : 3

In [52]:
```
t0 = nltk.DefaultTagger('NN')
t1 = nltk.UnigramTagger(br_train, backoff=t0)
t2 = nltk.BigramTagger(br_train, backoff=t1)
```

In [53]:
```
t2.evaluate(br_test)
```

Out[53]:
```
0.9111006662708622
```

Step : 4

In [54]:
```
total_train = [len(l) for l in br_train]
sum(total_train)
```

Out[54]:
```
1039920
```

In [55]:
```
total_test = [len(l) for l in br_test]
sum(total_test)
```

Out[55]:
```
121272
```

In [56]:
```
t1.evaluate(br_test)
```

Out[56]:
```
0.8897849462365591
```

In [57]:
```
t2.evaluate(br_test)
```

Out[57]:
```
0.9111006662708622
```

```
In [58]: br_train[0]
```

```
Out[58]: [('The', 'AT'),
          ('Fulton', 'NP-TL'),
          ('County', 'NN-TL'),
          ('Grand', 'JJ-TL'),
          ('Jury', 'NN-TL'),
          ('said', 'VBD'),
          ('Friday', 'NR'),
          ('an', 'AT'),
          ('investigation', 'NN'),
          ('of', 'IN'),
          ("Atlanta's", 'NP$'),
          ('recent', 'JJ'),
          ('primary', 'NN'),
          ('election', 'NN'),
          ('produced', 'VBD'),
          ('``', '``'),
          ('no', 'AT'),
          ('evidence', 'NN'),
          ("''", "''"),
          ('that', 'CS'),
          ('any', 'DTI'),
          ('irregularities', 'NNS'),
          ('took', 'VBD'),
          ('place', 'NN'),
          ('.', '.')]
```

```
In [59]: br_train[1277]
```

```
Out[59]: [('``', '``'),
          ('I', 'PPSS'),
          ('told', 'VBD'),
          ('him', 'PPO'),
          ('who', 'WPS'),
          ('I', 'PPSS'),
          ('was', 'BEDZ'),
          ('and', 'CC'),
          ('he', 'PPS'),
          ('was', 'BEDZ'),
          ('quite', 'QL'),
          ('cold', 'JJ'),
          ('.', '.')]
```

```
In [60]: br_train[1277] [11]
```

```
Out[60]: ('cold', 'JJ')
```

```
In [61]: br_train_flat = [(word, tag) for sent in br_train for (word, tag) in sent]
In [62]: br_train_flat[:40]
```

```
Out[62]: [('The', 'AT'),
          ('Fulton', 'NP-TL'),
          ('County', 'NN-TL'),
          ('Grand', 'JJ-TL'),
          ('Jury', 'NN-TL'),
          ('said', 'VBD'),
          ('Friday', 'NR'),
          ('an', 'AT'),
          ('investigation', 'NN'),
          ('of', 'IN'),
          ("Atlanta's", 'NP$'),
          ('recent', 'JJ'),
          ('primary', 'NN'),
          ('election', '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')]
```

```
In [63]: br_train_flat[13]
```

```
Out[63]: ('election', 'NN')
```

```
In [64]: fd = nltk.FreqDist(br_train_flat)
         cfd = nltk.ConditionalFreqDist(br_train_flat)
```

```
In [65]: cfd['cold'].most_common()
```

```
Out[65]: [('JJ', 110), ('NN', 8), ('RB', 2)]
```

In [66]:

```
br_train_2grams = list(nltk.ngrams(br_train_flat, 2))
br_train_cold = [a[1] for (a,b) in br_train_2grams if b[0] == 'cold']
fdist = nltk.FreqDist(br_train_cold)
[tag for (tag, _) in fdist.most_common()]
```

Out[66]:

```
['AT',
 'IN',
 'CC',
 'QL',
 'BEDZ',
 'JJ',
 ',',
 'DT',
 'PP$',
 'RP',
 '``',
 'NN',
 'VBN',
 'VBD',
 'CS',
 'BEZ',
 'DOZ',
 'RB',
 'PPSS',
 'BE',
 'VB',
 'VBZ',
 'NP$',
 'BEDZ*',
 '--',
 'DTI',
 'WRB',
 'BED']
```

In [67]:

```
br_pre = [(w2+"/"+t2, t1) for ((w1,t1),(w2,t2)) in br_train_2grams]
br_pre_cfd = nltk.ConditionalFreqDist(br_pre)
br_pre
```

```
 ('primary/NN', 'JJ'),
 ('election/NN', 'NN'),
 ('produced/VBD', 'NN'),
 ('``/``', 'VBD'),
 ('no/AT', '``'),
 ('evidence/NN', 'AT'),
 ("''/''", 'NN'),
 ('that/CS', "''"),
 ('any/DTI', 'CS'),
 ('irregularities/NNS', 'DTI'),
 ('took/VBD', 'NNS'),
 ('place/NN', 'VBD'),
 ('./.', 'NN'),
 ('The/AT', '.'),
 ('jury/NN', 'AT'),
 ('further/RBR', 'NN'),
 ('said/VBD', 'RBR'),
 ('in/IN', 'VBD'),
 ('term-end/NN', 'IN'),
 ('presentments/NNS', 'NN'),
 ('that/CS', 'NNS'),
```

In [68]:

```
br_pre_cfd['cold/NN'].most_common()
```

Out[68]:

```
[('AT', 4), ('JJ', 2), (',', 1), ('DT', 1)]
```

```
In [69]: br_pre_cfd['cold/JJ'].most_common()
```

```
Out[69]: [('AT', 38),
          ('IN', 14),
          ('CC', 8),
          ('QL', 7),
          ('BEDZ', 7),
          ('JJ', 4),
          ('DT', 3),
          (',', 3),
          ('PP$', 3),
          ('``', 2),
          ('NN', 2),
          ('VBN', 2),
          ('VBD', 2),
          ('CS', 1),
          ('BEZ', 1),
          ('DOZ', 1),
          ('RB', 1),
          ('PPSS', 1),
          ('BE', 1),
          ('VB', 1),
          ('VBZ', 1),
          ('NP$', 1),
          ('BEDZ*', 1),
          ('--', 1),
          ('RP', 1),
          ('DTI', 1),
          ('WRB', 1),
          ('BED', 1)]
```

```
In [70]: bigram_tagger = nltk.BigramTagger(br_train)
```

```
In [71]: text1 = word_tokenize('I was very cold.')
         bigram_tagger.tag(text1)
```

```
Out[71]: [('I', 'PPSS'), ('was', 'BEDZ'), ('very', 'QL'), ('cold', 'JJ'), ('.', '.')]
```

```
In [72]: text2 = word_tokenize('I had a cold.')
         bigram_tagger.tag(text2)
```

```
Out[72]: [('I', 'PPSS'), ('had', 'HVD'), ('a', 'AT'), ('cold', 'JJ'), ('.', '.')]
```

```
In [73]: text3 = word_tokenize('I had a severe cold.')
         bigram_tagger.tag(text3)
```

```
Out[73]: [('I', 'PPSS'),
          ('had', 'HVD'),
          ('a', 'AT'),
          ('severe', 'JJ'),
          ('cold', 'JJ'),
          ('.', '.')]
```

```
In [74]: text4 = word_tokenize('January was a cold month.')
         bigram_tagger.tag(text4)
```

```
Out[74]: [('January', None),
          ('was', None),
          ('a', None),
          ('cold', None),
          ('month', None),
          ('.', None)]
```

```
In [75]: text5 = word_tokenize('I failed to do so.')
         bigram_tagger.tag(text5)
```

```
Out[75]: [('I', 'PPSS'),
          ('failed', 'VBD'),
          ('to', 'TO'),
          ('do', 'DO'),
          ('so', 'RB'),
          ('.', '.')]
```

In [76]:
```
text6 = word_tokenize('I was happy,but so was my enemy.')
bigram_tagger.tag(text6)
```

Out[76]:
```
[('I', 'PPSS'),
 ('was', 'BEDZ'),
 ('happy', 'JJ'),
 (',', ','),
 ('but', 'CC'),
 ('so', 'RB'),
 ('was', 'BEDZ'),
 ('my', 'PP$'),
 ('enemy', 'NN'),
 ('.', '.')]
```

In [77]:
```
text7 = word_tokenize('So, how was the exam?')
bigram_tagger.tag(text7)
```

Out[77]:
```
[('So', 'RB'),
 (',', ','),
 ('how', 'WRB'),
 ('was', 'BEDZ'),
 ('the', 'AT'),
 ('exam', None),
 ('?', None)]
```

In [78]:
```
text8 = word_tokenize('The students came in early so they can get good seats.')
bigram_tagger.tag(text8)
```

Out[78]:
```
[('The', 'AT'),
 ('students', 'NNS'),
 ('came', 'VBD'),
 ('in', 'IN'),
 ('early', 'JJ'),
 ('so', 'CS'),
 ('they', 'PPSS'),
 ('can', 'MD'),
 ('get', 'VB'),
 ('good', 'JJ'),
 ('seats', 'NNS'),
 ('.', '.')]
```

In [79]:
```
text9 = word_tokenize('She failed the exam, so she must take it again.')
bigram_tagger.tag(text9)
```

Out[79]:
```
[('She', 'PPS'),
 ('failed', 'VBD'),
 ('the', 'AT'),
 ('exam', None),
 (',', None),
 ('so', None),
 ('she', None),
 ('must', None),
 ('take', None),
 ('it', None),
 ('again', None),
 ('.', None)]
```

In [80]:
```
text10 = word_tokenize('That was so incredible.')
bigram_tagger.tag(text10)
```

Out[80]:
```
[('That', 'DT'),
 ('was', 'BEDZ'),
 ('so', 'QL'),
 ('incredible', 'JJ'),
 ('.', '.')]
```

In [81]:
```
text11 = word_tokenize('Wow, so incredible.')
bigram_tagger.tag(text11)
```

Out[81]:
```
[('Wow', None), (',', None), ('so', None), ('incredible', None), ('.', None)]
```

In [ ]: