Lab - 1

Name : Arul Kumar ARK ¶

Roll No. : 225229103

Understanding Large Text File

Exercise : 1

In [18]:
```
import nltk
```

In [19]:
```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\1mscdsa03\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[19]:
```
True
```

In [20]:
```
text="This is Andrew's text, isn't it?"
```

In [21]:
```
tokenizer = nltk.tokenize.WhitespaceTokenizer()
tokens = tokenizer.tokenize(text)
print(len(tokens))
print(tokens)
```

```
6
['This', 'is', "Andrew's", 'text,', "isn't", 'it?']
```

In [33]:
```
tokenizer = nltk.tokenize.TreebankWordTokenizer()
tokens = tokenizer.tokenize(text)
print(len(tokens))
print(tokens)
```

```
10
['This', 'is', 'Andrew', "'s", 'text', ',', 'is', "n't", 'it', '?']
```

In [34]:
```
tokenizer = nltk.tokenize.WordPunctTokenizer()
tokens = tokenizer.tokenize(text)
print(len(tokens))
print(tokens)
```

```
12
['This', 'is', 'Andrew', "'", 's', 'text', ',', 'isn', "'", 't', 'it', '?']
```

Exercise : 2

In [38]:
```
txt=open('gift-of-magi.txt')
file=txt.read()
print(file)
```

```
When Della reached home her intoxication gave way a little to prudence and re
ason. She got out her curling irons and lighted the gas and went to work repa
iring the ravages made by generosity added to love. Which is always a tremend
ous task dear friends--a mammoth task.


Within forty minutes her head was covered with tiny, close-lying curls that m
ade her look wonderfully like a truant schoolboy. She looked at her reflectio
n in the mirror long, carefully, and critically.


"If Jim doesn't kill me," she said to herself, "before he takes a second look
at me, he'll say I look like a Coney Island chorus girl. But what could I do-
-oh! what could I do with a dollar and eighty-seven cents?"


At 7 o'clock the coffee was made and the frying-pan was on the back of the st
ove hot and ready to cook the chops.
```

In [41]:
```
tokens = tokenizer.tokenize(file)
print(len(tokens))
```

```
2519
```

In [40]:
```
print(len(file))
```

```
11329
```

In [61]:
```
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\1mscdsa03\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

Out[61]:
```
True
```

In [69]:
```
print(nltk.pos_tag(tokens))
print(len(nltk.pos_tag(tokens)))
```

```
D'), ('it', 'PRP'), ('.', '.'), ('He', 'PRP'), ('looked', 'VBD'), ('thin', 'J
J'), ('and', 'CC'), ('very', 'RB'), ('serious', 'JJ'), ('.', '.'), ('Poor',
'NNP'), ('fellow', 'NN'), (',', ','), ('he', 'PRP'), ('was', 'VBD'), ('only',
'RB'), ('twenty', 'JJ'), ('-', ':'), ('two', 'CD'), ('--', ':'), ('and', 'C
C'), ('to', 'TO'), ('be', 'VB'), ('burdened', 'VBN'), ('with', 'IN'), ('a',
'DT'), ('family', 'NN'), ('!', '.'), ('He', 'PRP'), ('needed', 'VBD'), ('a',
'DT'), ('new', 'JJ'), ('overcoat', 'NN'), ('and', 'CC'), ('he', 'PRP'), ('wa
s', 'VBD'), ('without', 'IN'), ('gloves', 'NNS'), ('.', '.'), ('Jim', 'NNP'),
('stepped', 'VBD'), ('inside', 'IN'), ('the', 'DT'), ('door', 'NN'), (',',
','), ('as', 'RB'), ('immovable', 'JJ'), ('as', 'IN'), ('a', 'DT'), ('sette
r', 'NN'), ('at', 'IN'), ('the', 'DT'), ('scent', 'NN'), ('of', 'IN'), ('quai
l', 'NN'), ('.', '.'), ('His', 'PRP$'), ('eyes', 'NNS'), ('were', 'VBD'), ('f
ixed', 'VBN'), ('upon', 'IN'), ('Della', 'NNP'), (',', ','), ('and', 'CC'),
('there', 'EX'), ('was', 'VBD'), ('an', 'DT'), ('expression', 'NN'), ('in',
'IN'), ('them', 'PRP'), ('that', 'IN'), ('she', 'PRP'), ('could', 'MD'), ('no
t', 'RB'), ('read', 'VB'), (',', ','), ('and', 'CC'), ('it', 'PRP'), ('terrif
ied', 'VBD'), ('her', 'PRP'), ('.', '.'), ('It', 'PRP'), ('was', 'VBD'), ('no
t', 'RB'), ('anger', 'JJ'), (',', ','), ('nor', 'CC'), ('surprise', 'NN'),
(',', ','), ('nor', 'CC'), ('disapproval', 'NN'), (',', ','), ('nor', 'CC'),
('horror', 'NN'), ('.', '.'), ('nor', 'CC'), ('any', 'DT'), ('of', 'IN'), ('t
```

In [68]:
```
from collections import Counter
top_tokens = Counter(tokens)
print(top_tokens.most_common(20))
print(len(top_tokens))
```

```
[('.', 139), ('the', 109), (',', 95), ('and', 75), ('a', 65), ('of', 51), ('t
o', 41), ("'", 38), ('"', 34), ('it', 29), ('was', 27), ('Jim', 26), ('she', 2
5), ('in', 24), ('her', 24), ('had', 21), ('that', 20), ('Della', 20), ('for',
20), ('at', 19)]
835
```

In [91]:
```
for chrt in tokens:
    if len(chrt)>10:
        len_10=chrt
        print([(len_10)],end=",")
print()
print(len(len_10))
```

```
['predominating'],['description'],['appertaining'],['contracting'],['longitudin
al'],['brilliantly'],['possessions'],['grandfather'],['proclaiming'],['meretric
ious'],['ornamentation'],['description'],['intoxication'],['wonderfully'],['dis
approval'],['laboriously'],['inconsequential'],['mathematician'],['illuminate
d'],['necessitating'],['wonderfully'],['duplication'],
11
```

In [95]:
```
wrd=[word for word in tokens if len(word) >10]
freq=nltk.FreqDist(wrd)
for wrds,cnt in freq.items():
    if len(wrds)>10 and cnt >1:
        print(wrds,cnt)
```

```
description 2
wonderfully 2
```

Exercise 3

In [102]:
```
fname = "austen-emma.txt"
f = open(fname, 'r')
etxt= f.read()
print(etxt)
f.close()
```

```
The event had every promise of happiness for her friend.  Mr. Weston
was a man of unexceptionable character, easy fortune, suitable age,
and pleasant manners; and there was some satisfaction in considering
with what self-denying, generous friendship she had always wished
and promoted the match; but it was a black morning's work for her.
The want of Miss Taylor would be felt every hour of every day.
She recalled her past kindness--the kindness, the affection of sixteen
years--how she had taught and how she had played with her from five
years old--how she had devoted all her powers to attach and amuse
her in health--and how nursed her through the various illnesses
of childhood.  A large debt of gratitude was owing here; but the
intercourse of the last seven years, the equal footing and perfect
unreserve which had soon followed Isabella's marriage, on their
being left to each other, was yet a dearer, tenderer recollection.
She had been a friend and companion such as few possessed: intelligent,
well-informed, useful, gentle, knowing all the ways of the family,
interested in all its concerns, and peculiarly interested in herself,
in every pleasure, every scheme of hers--one to whom she could speak
```

In [103]:
```
f = open(fname, 'r')
etxt = f.read()
f.close()
etxt[-200:]
```

Out[103]:
```
' deficiencies, the wishes,\nthe hopes, the confidence, the predictions of the
small band\nof true friends who witnessed the ceremony, were fully answered\nin
the perfect happiness of the union.\n\n\n\n\nFINIS'
```

In [104]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\1mscdsa03\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

Out[104]:
```
True
```

In [105]:
```
etoks = nltk.word_tokenize(etxt.lower())
etoks [-20:]
```

Out[105]:
```
['of',
 'true',
 'friends',
 'who',
 'witnessed',
 'the',
 'ceremony',
 ',',
 'were',
 'fully',
 'answered',
 'in',
 'the',
 'perfect',
 'happiness',
 'of',
 'the',
 'union',
 '.',
 'finis']
```

In [107]:
```
etypes =sorted(set(etoks))
print(etypes[-10:])
print(len(etypes))
```

```
['yours', 'yourself', 'yourself.', 'youth', 'youthful', 'zeal', 'zigzags', '»',
'¿', 'ï']
8003
```

In [108]:
```
efreq = nltk.FreqDist(etoks)
efreq['the']
```

Out[108]:
```
5198
```

Question 1: Words with prefix and suffix

In [109]:
```
words=[word for word in etoks if word.startswith("un") & word.endswith("able")]
print(words)
```

```
['unexceptionable', 'unsuitable', 'unreasonable', 'unreasonable', 'uncomfortabl
e', 'unfavourable', 'unexceptionable', 'unexceptionable', 'uncomfortable', 'unp
ersuadable', 'unavoidable', 'unreasonable', 'uncomfortable', 'unsuitable', 'unm
anageable', 'unexceptionable', 'unreasonable', 'unobjectionable', 'unpersuadabl
e', 'unsuitable', 'unreasonable', 'uncomfortable', 'unexceptionable', 'unpardon
able', 'unmanageable', 'unanswerable', 'unfavourable', 'unpersuadable', 'unacco
untable', 'undesirable', 'unable', 'unable', 'unpardonable', 'unexceptionable',
'unreasonable', 'unreasonable', 'uncomfortable', 'unreasonable', 'unpardonabl
e', 'unaccountable', 'unexceptionable', 'unreasonable', 'unaccountable']
```

Question 2: Length

In [110]:
```
tokenizer = nltk.tokenize.WordPunctTokenizer()
toke = tokenizer.tokenize (etxt)
words=[word for word in toke if len(word)>15]
print(words)
```

```
['companionableness', 'misunderstanding', 'incomprehensible', 'undistinguishin
g', 'unceremoniousness', 'Disingenuousness', 'disagreeableness', 'misunderstand
ings', 'misunderstandings', 'misunderstandings', 'misunderstandings', 'disinter
estedness', 'unseasonableness']
```

Question 3: Average word length

In [111]:
```
avg=sum(len (word) for word in toke)/len (toke)
print(avg )
```

```
3.7552643066497597
```

Question 4: Word frequency

In [113]:
```
fdiemm = FreqDist (toke)

for wrd,cont in fdiemm.items():
    if cont > 200:
        print(wrd,cont)
```

```
Jane 301
I 3178
Woodhouse 313
, 11454
and 4672
with 1187
a 3004
to 5183
some 248
of 4279
the 4844
; 2199
had 1606
- 574
one 413
in 2118
very 1151
little 354
or 490
her 2381
```

Question 4:Eamma Words not in fdiemm

In [142]:
```
if "Emma" in fdiemm:
    print("yes")
```

```
yes
```

STEP 3: bigrams in Emma

In [114]:
```
e2grams = list(nltk.bigrams (toke))
e2gramfd = nltk.FreqDist(e2grams)
print(e2gramfd )
```

```
<FreqDist with 66580 samples and 192427 outcomes>
```

Question 6: Bigrams

In [115]:
```
last_ten = FreqDist(dict(e2gramfd.most_common()[-10:]))
last_ten
```

Out[115]:
```
FreqDist({('.', 'FINIS'): 1,
          ('answered', 'in'): 1,
          ('fully', 'answered'): 1,
          ('the', 'ceremony'): 1,
          ('the', 'perfect'): 1,
          ('the', 'union'): 1,
          ('union', '.'): 1,
          ('were', 'fully'): 1,
          ('who', 'witnessed'): 1,
          ('witnessed', 'the'): 1})
```

In [118]:
```
tokenizer = nltk.tokenize. WhitespaceTokenizer()
tokens =tokenizer.tokenize(etxt)
print(tokens)
```

```
'years', 'had', 'Miss', 'Taylor', 'been', 'in', 'Mr.', 'Woodhouse's', 'famil
y,', 'less', 'as', 'a', 'governess', 'than', 'a', 'friend,', 'very', 'fond',
'of', 'both', 'daughters,', 'but', 'particularly', 'of', 'Emma.', 'Between',
'_them_', 'it', 'was', 'more', 'the', 'intimacy', 'of', 'sisters.', 'Even',
'before', 'Miss', 'Taylor', 'had', 'ceased', 'to', 'hold', 'the', 'nominal',
'office', 'of', 'governess,', 'the', 'mildness', 'of', 'her', 'temper', 'ha
d', 'hardly', 'allowed', 'her', 'to', 'impose', 'any', 'restraint;', 'and',
'the', 'shadow', 'of', 'authority', 'being', 'now', 'long', 'passed', 'awa
y,', 'they', 'had', 'been', 'living', 'together', 'as', 'friend', 'and', 'fri
end', 'very', 'mutually', 'attached,', 'and', 'Emma', 'doing', 'just', 'wha
t', 'she', 'liked;', 'highly', 'esteeming', 'Miss', "Taylor's", 'judgment,',
'but', 'directed', 'chiefly', 'by', 'her', 'own.', 'The', 'real', 'evils,',
'indeed,', 'of', "Emma's", 'situation', 'were', 'the', 'power', 'of', 'havin
g', 'rather', 'too', 'much', 'her', 'own', 'way,', 'and', 'a', 'disposition',
'to', 'think', 'a', 'little', 'too', 'well', 'of', 'herself;', 'these', 'wer
e', 'the', 'disadvantages', 'which', 'threatened', 'alloy', 'to', 'her', 'man
y', 'enjoyments.', 'The', 'danger,', 'however,', 'was', 'at', 'present', 's
o', 'unperceived,', 'that', 'they', 'did', 'not', 'by', 'any', 'means', 'ran
k', 'as', 'misfortunes', 'with', 'her.', 'Sorrow', 'came--a', 'gentle', 'sorr
ow--but', 'not', 'at', 'all', 'in', 'the', 'shape', 'of', 'any', 'disagreeabl
```

In [121]:
```
e2grams = list(nltk.bigrams (tokens))
e2gramfd = nltk.FreqDist(e2grams)
print(e2gramfd)
```

```
<FreqDist with 82983 samples and 158166 outcomes>
```

In [122]:
```
e2gramfd.most_common (20)
```

Out[122]:
```
[(('to', 'be'), 562),
 (('of', 'the'), 556),
 (('in', 'the'), 431),
 (('I', 'am'), 302),
 (('had', 'been'), 299),
 (('could', 'not'), 270),
 (('it', 'was'), 253),
 (('she', 'had'), 242),
 (('to', 'the'), 236),
 (('have', 'been'), 233),
 (('of', 'her'), 230),
 (('I', 'have'), 214),
 (('and', 'the'), 208),
 (('would', 'be'), 208),
 (('she', 'was'), 206),
 (('do', 'not'), 196),
 (('of', 'his'), 182),
 (('that', 'she'), 178),
 (('to', 'have'), 176),
 (('such', 'a'), 176)]
```

Question 8: Bigram frequency count

In [123]:
```
for w, c in e2gramfd.items():
    if w == ('so', 'happy'):
        print(w,c)
```

```
('so', 'happy') 3
```

Question 9: Word following 'so'

In [124]:
```
import re
```

In [127]:
```
words = re.findall(r'so+ \w+', open('austen-emma.txt').read())
so= Counter(zip(words))
print(so)
```

```
Counter({('so much',): 95, ('so very',): 76, ('so well',): 30, ('so many',): 2
7, ('so long',): 27, ('so little',): 20, ('so far',): 17, ('so I',): 14, ('so k
ind',): 13, ('so good',): 12, ('so often',): 10, ('so soon',): 9, ('so grea
t',): 8, ('so to',): 7, ('so fond',): 7, ('so she',): 7, ('so it',): 6, ('so an
xious',): 6, ('so as',): 6, ('so you',): 6, ('so truly',): 6, ('so completel
y',): 5, ('so obliging',): 5, ('so extremely',): 5, ('so entirely',): 4, ('so h
appy',): 4, ('so interesting',): 4, ('so fast',): 4, ('so near',): 4, ('so plea
sed',): 4, ('so few',): 4, ('so that',): 4, ('so strong',): 4, ('so liberal',):
4, ('so miserable',): 4, ('so happily',): 3, ('so proper',): 3, ('so pleasantl
y',): 3, ('so superior',): 3, ('so warmly',): 3, ('so bad',): 3, ('so odd',):
3, ('so ill',): 3, ('so delighted',): 3, ('so particularly',): 3, ('so easil
y',): 3, ('so on',): 3, ('so attentive',): 3, ('so fortunate',): 3, ('so gla
d',): 3, ('so shocked',): 3, ('so at',): 3, ('so obliged',): 2, ('so perfectl
y',): 2, ('so dear',): 2, ('so busy',): 2, ('so did',): 2, ('so forth',): 2,
('so totally',): 2, ('so remarkably',): 2, ('so plainly',): 2, ('so charmin
g',): 2, ('so surprized',): 2, ('so early',): 2, ('so too',): 2, ('so easy',):
2, ('so decidedly',): 2, ('so absolutely',): 2, ('so particular',): 2, ('so dec
eived',): 2, ('so palpably',): 2, ('so clever',): 2, ('so short',): 2, ('so col
d',): 2, ('so high',): 2, ('so happened',): 2, ('so full',): 2, ('so thoroughl
y',): 2, ('so equal',): 2, ('so off',): 2, ('so naturally',): 2, ('so afrai
d',): 2, ('so deep',): 2, ('so kindly',): 2, ('so pale',): 2, ('so noble',): 2,
('so lovely',): 2, ('so mad',): 2, ('so nearly',): 2, ('so sorry',): 2, ('so ch
eerful',): 2, ('so unfeeling',): 2, ('so ready',): 2, ('so unperceived',): 1,
('so mild',): 1, ('so constantly',): 1, ('so comfortably',): 1, ('so avowed',):
1, ('so deservedly',): 1, ('so convenient',): 1, ('so just',): 1, ('so apparen
t',): 1, ('so sorrowful',): 1, ('so spent',): 1, ('so artlessly',): 1, ('so pla
in',): 1, ('so firmly',): 1, ('so genteel',): 1, ('so _then_',): 1, ('so brilli
ant',): 1, ('so seldom',): 1, ('so nervous',): 1, ('so indeed',): 1, ('so pac
k',): 1, ('so doubtful',): 1, ('so with',): 1, ('so contemptible',): 1, ('so sl
ightingly',): 1, ('so by',): 1, ('so loudly',): 1, ('so materially',): 1, ('so
hard',): 1, ('so delightful',): 1, ('so pointed',): 1, ('so equalled',): 1, ('s
o evidently',): 1, ('so immediately',): 1, ('so sought',): 1, ('so excellen
t',): 1, ('so prettily',): 1, ('so extreme',): 1, ('so wonder',): 1, ('so alway
s',): 1, ('so silly',): 1, ('so satisfied',): 1, ('so smiling',): 1, ('so prosi
ng',): 1, ('so undistinguishing',): 1, ('so apt',): 1, ('so dreadful',): 1, ('s
o respected',): 1, ('so tenderly',): 1, ('so grieved',): 1, ('so shocking',):
1, ('so conceited',): 1, ('so before',): 1, ('so prevalent',): 1, ('so heav
y',): 1, ('so swiftly',): 1, ('so spoken',): 1, ('so or',): 1, ('so overcharge
d',): 1, ('so pleasant',): 1, ('so fenced',): 1, ('so hospitable',): 1, ('so in
terested',): 1, ('so sanguine',): 1, ('so sure',): 1, ('so careless',): 1, ('so
rapidly',): 1, ('so frequent',): 1, ('so sensible',): 1, ('so misled',): 1, ('s
o blind',): 1, ('so complaisant',): 1, ('so misinterpreted',): 1, ('so activ
e',): 1, ('so pointedly',): 1, ('so striking',): 1, ('so sudden',): 1, ('so ind
ustriously',): 1, ('so partial',): 1, ('so natural',): 1, ('so inevitable',):
1, ('so lately',): 1, ('so beautifully',): 1, ('so distinct',): 1, ('so conside
rate',): 1, ('so light',): 1, ('so intimate',): 1, ('so magnified',): 1, ('so c
autious',): 1, ('so confined',): 1, ('so wish',): 1, ('so he',): 1, ('so glorio
us',): 1, ('so quick',): 1, ('so sweetly',): 1, ('so inseparably',): 1, ('so de
serving',): 1, ('so disappointed',): 1, ('so ended',): 1, ('so sluggish',): 1,
('so amiable',): 1, ('so quiet',): 1, ('so idolized',): 1, ('so cried',): 1,
('so acceptable',): 1, ('so properly',): 1, ('so reasonable',): 1, ('so delight
fully',): 1, ('so rich',): 1, ('so warm',): 1, ('so large',): 1, ('so handsomel
y',): 1, ('so abundant',): 1, ('so outree',): 1, ('so thoughtful',): 1, ('so mu
```

```
st',): 1, ('so effectually',): 1, ('so beautiful',): 1, ('so Patty',): 1, ('so
honoured',): 1, ('so close',): 1, ('so imprudent',): 1, ('so limited',): 1, ('s
o from',): 1, ('so amusing',): 1, ('so indifferent',): 1, ('so indignant',): 1,
('so said',): 1, ('so right',): 1, ('so wretched',): 1, ('so now',): 1, ('so oc
cupied',): 1, ('so unhappy',): 1, ('so highly',): 1, ('so generally',): 1, ('so
exactly',): 1, ('so double',): 1, ('so secluded',): 1, ('so regular',): 1, ('so
determined',): 1, ('so motherly',): 1, ('so the',): 1, ('so glibly',): 1, ('so
calculated',): 1, ('so thrown',): 1, ('so exclusively',): 1, ('so disgustingl
y',): 1, ('so needlessly',): 1, ('so does',): 1, ('so resolutely',): 1, ('so wo
uld',): 1, ('so infinitely',): 1, ('so fluently',): 1, ('so they',): 1, ('so im
patient',): 1, ('so briskly',): 1, ('so vigorously',): 1, ('so young',): 1, ('s
o hardened',): 1, ('so gratified',): 1, ('so received',): 1, ('so then',): 1,
('so and',): 1, ('so gratefully',): 1, ('so found',): 1, ('so placed',): 1, ('s
o lain',): 1, ('so his',): 1, ('so arranged',): 1, ('so moving',): 1, ('so walk
ing',): 1, ('so when',): 1, ('so favourable',): 1, ('so late',): 1, ('so silen
t',): 1, ('so dull',): 1, ('so irksome',): 1, ('so agitated',): 1, ('so bruta
l',): 1, ('so cruel',): 1, ('so depressed',): 1, ('so no',): 1, ('so justly',):
1, ('so astonished',): 1, ('so will',): 1, ('so simple',): 1, ('so dignifie
d',): 1, ('so suddenly',): 1, ('so a',): 1, ('so herself',): 1, ('so peremptori
ly',): 1, ('so uneasy',): 1, ('so wonderful',): 1, ('so _very_',): 1, ('so expr
essly',): 1, ('so angry',): 1, ('so anxiously',): 1, ('so strange',): 1, ('so s
toutly',): 1, ('so mistake',): 1, ('so mistaken',): 1, ('so dreadfully',): 1,
('so voluntarily',): 1, ('so satisfactory',): 1, ('so disinterested',): 1, ('so
foolishly',): 1, ('so ingeniously',): 1, ('so entreated',): 1, ('so like',): 1,
('so cordially',): 1, ('so essential',): 1, ('so designedly',): 1, ('so hast
y',): 1, ('so richly',): 1, ('so grateful',): 1, ('so tenaciously',): 1, ('so f
eeling',): 1, ('so engaging',): 1, ('so engaged',): 1, ('so hot',): 1, ('so use
ful',): 1, ('so attached',): 1, ('so peculiarly',): 1, ('so singularly',): 1,
('so taken',): 1, ('so recently',): 1, ('so fresh',): 1, ('so hateful',): 1,
('so heartily',): 1, ('so steady',): 1, ('so complete',): 1, ('so in',): 1, ('s
o suffered',): 1})
```

Question 10: Trigrams

In [130]:
```
e3grams = list(nltk.trigrams(tokens))
e3gramfd = nltk.FreqDist(e3grams)
print(e3gramfd)
```

```
<FreqDist with 137443 samples and 158165 outcomes>
```

In [131]:
```
last_ten = FreqDist(dict(e3gramfd.most_common()[-10:]))
last_ten
```

Out[131]:
```
FreqDist({('answered', 'in', 'the'): 1,
          ('ceremony,', 'were', 'fully'): 1,
          ('fully', 'answered', 'in'): 1,
          ('in', 'the', 'perfect'): 1,
          ('of', 'the', 'union.'): 1,
          ('perfect', 'happiness', 'of'): 1,
          ('the', 'ceremony,', 'were'): 1,
          ('the', 'perfect', 'happiness'): 1,
          ('the', 'union.', 'FINIS'): 1,
          ('were', 'fully', 'answered'): 1})
```

Question 11: Trigram top frequency

In [133]:
```
print(e3gramfd.most_common(10))
```

```
[(('I', 'do', 'not'), 94), (('I', 'am', 'sure'), 75), (('would', 'have', 'bee
n'), 55), (('a', 'great', 'deal'), 55), (('she', 'could', 'not'), 49), (('coul
d', 'not', 'be'), 45), (('she', 'had', 'been'), 44), (('it', 'would', 'be'), 4
3), (('do', 'not', 'know'), 43), (('Mr.', 'and', 'Mrs.'), 37)]
```

Question 12: trigram frequency count

In [148]:
```
word = re.findall(r'so happy to', open('austen-emma.txt').read())
print(word)
print(len(word))
```

```
['so happy to']
1
```

In [ ]:

In [ ]: