In [3]:
```
import re

# Question 1 - How many File Names in Provided File?
file = open('Assignment_12.txt', 'r')
text1 = file.read()
file.close()

print(text1)
```

```
arxiv_annotate10_7_1.txt   arxiv_annotate10_7_2.txt   arxiv_annotate10_7_3.txt   arx
iv_annotate1_13_1.txt   arxiv_annotate1_13_2.txt   arxiv_annotate1_13_3.txt   arxiv_
annotate2_66_1.txt   arxiv_annotate2_66_2.txt   arxiv_annotate2_66_3.txt   arxiv_ann
otate3_80_1.txt   arxiv_annotate3_80_2.txt   arxiv_annotate3_80_3.txt   arxiv_annota
te4_168_1.txt   arxiv_annotate4_168_2.txt   arxiv_annotate4_168_3.txt   arxiv_annota
te5_240_1.txt   arxiv_annotate5_240_2.txt   arxiv_annotate5_240_3.txt   arxiv_annota
te6_52_1.txt   arxiv_annotate6_52_2.txt   arxiv_annotate6_52_3.txt   arxiv_annotate7
_268_1.txt   arxiv_annotate7_268_2.txt   arxiv_annotate7_268_3.txt   arxiv_annotate8
_81_1.txt   arxiv_annotate8_81_2.txt   arxiv_annotate8_81_3.txt   arxiv_annotate9_27
9_1.txt   arxiv_annotate9_279_2.txt   arxiv_annotate9_279_3.txt   jdm_annotate10_210
_1.txt   jdm_annotate10_210_2.txt   jdm_annotate10_210_3.txt   jdm_annotate1_103_1.t
xt   jdm_annotate1_103_2.txt   jdm_annotate1_103_3.txt   jdm_annotate2_107_1.txt   j
dm_annotate2_107_2.txt   jdm_annotate2_107_3.txt   jdm_ann^otate3_120_1.txt   jdm_an
notate3_120_2.txt   jdm_annotate3_120_3.txt   jdm_annotate4_220_1.txt   jdm_annotate
4_220_2.txt   jdm_annotate4_220_3.txt   jdm_annotate5_228_1.txt   jdm_annotate5_228_
2.txt   jdm_annotate5_228_3.txt   jdm_annotate6_32_1.txt   jdm_anno&tate6_32_2.txt
jdm_annotate6_32_3.txt   jdm_annotate7_265_1.txt   jdm_annotate7_265_2.txt   jdm_ann
otate7_265_3.txt   jdm_annotate8_177_1.txt   jdm_annotat#e8_177_2.txt   jdm_annotate
8_177_3.txt   jdm_annotate9_45_1.txt   jdm_annotate9_45_2.txt   jdm_annotate9_45_3.t
xt   plos_annotate10_1140_1.txt   plos_annotate10_1140_2.txt   plos_annotate10_1140_
3.txt   plos_annotate1_6_1.txt   plos_annotat*e1_6_2.txt   plos_annotate1_6_3.txt
plos_annotate2_336_1.txt   plos_annotate2_336_2.txt   plos_annotate2_336_3.txt   plo
s_annotate3_798_1.txt   plos_annotate3_798_2.txt   plos_annotate3_798_3.txt   plos_a
nnotate4_1052_1.txt   plos_annotate4_1052_2.txt   plos_annotate4_1052_3.txt   plos_a
nnotate5_1375_1.txt   plos_annotate5_1375_2.txt   plos_anno%tate5_1375_3.txt   plos_
annotate6_1032_1.txt   plos_annotate6_1032_2.txt   plos_annotate6_1032_3.txt   plos_
annotate7_1233_1.txt   plos_annot@ate7_1233_2.txt   plos_annotate7_1233_3.txt   plos
_annotate8_123_1.txt   plos_annotate8_123_2.txt   plos_annotate8_123_3.txt   plos_an
notate9_1187_1.txt   plos_annotate9_1187_2.txt   plos_annotate9_1187_3.txt
```

In [48]:
```
# Since all the names are filenames, we can technically
# seperate each entry into an array of words
regex1 = re.compile('\s+')
text1_words = regex1.split(text1)

# However, we are not really checking for files, so a
# more robust way is to just create a pattern that looks
# for '.' followed by text, since all filename extensions
# much match that form
regex2 = re.compile('.[A-Z0-9]+\s+', flags=re.IGNORECASE)
text1_files = regex2.split(text1)
```

In [49]:
```
# Question 1 Results
print('Split on Spaces Method:   ', len(text1_words))
print('Split on File Extensions: ', len(text1_files))
```

```
Split on Spaces Method:    90
Split on File Extensions:  90
```

In [57]:
```
# Question 2 - Identify Pattern of filenames and Count total
# My assumption on this is that each filename is supposed to
# follow something like this:
# a-z(2-5? times)_annotate_0-9(1-2?)_0-9(1-4?)_0-9(1).txt

# We can assume the ranges are loose, and may utilize 0-9+

file_pattern = r'[a-z]+_annotate[0-9]+_[0-9]+_[0-9].txt'
regex3 = re.compile(file_pattern, flags=re.IGNORECASE)

text1_matches = regex3.findall(text1)
print('\nFilename matches to pattern: ', len(text1_matches))
display(text1_matches)
```

```
Filename matches to pattern:  84
```

```
['arxiv_annotate10_7_1.txt',
 'arxiv_annotate10_7_2.txt',
 'arxiv_annotate10_7_3.txt',
 'arxiv_annotate1_13_1.txt',
 'arxiv_annotate1_13_2.txt',
 'arxiv_annotate1_13_3.txt',
 'arxiv_annotate2_66_1.txt',
 'arxiv_annotate2_66_2.txt',
 'arxiv_annotate2_66_3.txt',
 'arxiv_annotate3_80_1.txt',
 'arxiv_annotate3_80_2.txt',
 'arxiv_annotate3_80_3.txt',
 'arxiv_annotate4_168_1.txt',
 'arxiv_annotate4_168_2.txt',
 'arxiv_annotate4_168_3.txt',
 'arxiv_annotate5_240_1.txt',
 'arxiv_annotate5_240_2.txt',
 'arxiv_annotate5_240_3.txt',
 'arxiv_annotate6_52_1.txt',
 'arxiv_annotate6_52_2.txt',
 'arxiv_annotate6_52_3.txt',
 'arxiv_annotate7_268_1.txt',
 'arxiv_annotate7_268_2.txt',
 'arxiv_annotate7_268_3.txt',
 'arxiv_annotate8_81_1.txt',
 'arxiv_annotate8_81_2.txt',
 'arxiv_annotate8_81_3.txt',
 'arxiv_annotate9_279_1.txt',
 'arxiv_annotate9_279_2.txt',
 'arxiv_annotate9_279_3.txt',
 'jdm_annotate10_210_1.txt',
 'jdm_annotate10_210_2.txt',
 'jdm_annotate10_210_3.txt',
 'jdm_annotate1_103_1.txt',
 'jdm_annotate1_103_2.txt',
 'jdm_annotate1_103_3.txt',
 'jdm_annotate2_107_1.txt',
 'jdm_annotate2_107_2.txt',
 'jdm_annotate2_107_3.txt',
 'jdm_annotate3_120_2.txt',
 'jdm_annotate3_120_3.txt',
 'jdm_annotate4_220_1.txt',
 'jdm_annotate4_220_2.txt',
 'jdm_annotate4_220_3.txt',
 'jdm_annotate5_228_1.txt',
 'jdm_annotate5_228_2.txt',
 'jdm_annotate5_228_3.txt',
 'jdm_annotate6_32_1.txt',
 'jdm_annotate6_32_3.txt',
 'jdm_annotate7_265_1.txt',
 'jdm_annotate7_265_2.txt',
 'jdm_annotate7_265_3.txt',
 'jdm_annotate8_177_1.txt',
 'jdm_annotate8_177_3.txt',
 'jdm_annotate9_45_1.txt',
 'jdm_annotate9_45_2.txt',
```

```
 'jdm_annotate9_45_3.txt',
 'plos_annotate10_1140_1.txt',
 'plos_annotate10_1140_2.txt',
 'plos_annotate10_1140_3.txt',
 'plos_annotate1_6_1.txt',
 'plos_annotate1_6_3.txt',
 'plos_annotate2_336_1.txt',
 'plos_annotate2_336_2.txt',
 'plos_annotate2_336_3.txt',
 'plos_annotate3_798_1.txt',
 'plos_annotate3_798_2.txt',
 'plos_annotate3_798_3.txt',
 'plos_annotate4_1052_1.txt',
 'plos_annotate4_1052_2.txt',
 'plos_annotate4_1052_3.txt',
 'plos_annotate5_1375_1.txt',
 'plos_annotate5_1375_2.txt',
 'plos_annotate6_1032_1.txt',
 'plos_annotate6_1032_2.txt',
 'plos_annotate6_1032_3.txt',
 'plos_annotate7_1233_1.txt',
 'plos_annotate7_1233_3.txt',
 'plos_annotate8_123_1.txt',
 'plos_annotate8_123_2.txt',
 'plos_annotate8_123_3.txt',
 'plos_annotate9_1187_1.txt',
 'plos_annotate9_1187_2.txt',
 'plos_annotate9_1187_3.txt']
```

In [111…

```
# Question 3 - Get File Names where Pattern wasn't Matched
# First, let is split the text file based on regex from
# last question, this will return empty spaces as well
# as the non-matching names
names_unmatched = regex3.split(text1)

unmatched_files = []

# Now we can filter out the space entries only through a for loop
txt_ext = r'.txt\s+'
regex5 = re.compile(txt_ext)
for i in range(len(names_unmatched)):
    if regex5.search(names_unmatched[i]):
#         print(names_unmatched[i])
        unmatched_files.append(names_unmatched[i])

print('Not matching file names:')
display(unmatched_files)
```

```
Not matching file names:
['   jdm_ann^otate3_120_1.txt   ',
 '   jdm_anno&tate6_32_2.txt   ',
 '   jdm_annotat#e8_177_2.txt   ',
 '   plos_annotat*e1_6_2.txt   ',
 '   plos_anno%tate5_1375_3.txt   ',
 '   plos_annot@ate7_1233_2.txt   ']
```

In [37]:
```
# Question 4 - Normalize the new file and determine counts
import nltk
#nltk.download('punkt')

# First we import the file and read
file2 = open('arxiv_annotate1_13_1.txt', 'r')
text2 = file2.read()
file2.close()

# Now we can use nltk to turn the text into words
text2_words = nltk.word_tokenize(text2)
text2_words[:25]
```

Out[37]:
```
['#',
 '#',
 '#',
 'abstract',
 '#',
 '#',
 '#',
 'MISC',
 'although',
 'the',
 'internet',
 'as',
 'level',
 'topology',
 'has',
 'been',
 'extensively',
 'studied',
 'over',
 'the',
 'past',
 'few',
 'years',
 'little',
 'is']
```

In [38]:
```
# Now, we can utilize FreqDist to turn the
# words into a dictionary of unique words and
# respective counts
dist = nltk.FreqDist(text2_words)

print('Unique words found: ', len(dist))
# We can view the Head just by displaying the dictionary
display(dist)
```

```
Unique words found:  334
FreqDist({'the': 44, 'of': 34, 'as': 28, 'and': 24, 'MISC': 20, 'we': 20, 'a': 19, '
in': 19, 'to': 18, 'internet': 15, ...})
```

In [ ]: