In [1]:
```
pwd
```

Out[1]:
```
'C:\\Users\\Dell'
```

In [6]:
```
cd downloads
```

```
C:\Users\Dell\downloads
```

In [4]:
```
ls
```

```
 Volume in drive C is OS
 Volume Serial Number is 04BF-48F9

 Directory of C:\Users\Dell\downloads

23-12-2019  00:08       136,316,289 #.zip
15-04-2020  12:40    <DIR>          .
15-04-2020  12:40    <DIR>          ..
08-04-2019  21:36    <DIR>          [FreeTutorials.Eu] Udemy - learn-data-structure-algorithms-with-java-interview
31-01-2019  05:22    <DIR>          [FreeTutorials.Eu] Udemy - php-for-complete-beginners-includes-msql-object-oriented
10-02-2019  19:36           174,897 __RESUME__.pdf
02-08-2019  22:50           335,977 135.png
02-08-2019  22:51            26,880 138.png
30-01-2019  17:31    <DIR>          2013-g7-ProjectFiles
17-01-2019  15:30    <DIR>          2015-g3-ProjectFiles
17-01-2019  15:30        16,817,718 2015-g3-ProjectFiles.zip
26-11-2018  12:20        28,468,910 4kvideodownloader_4.4.11_x64.zip
04-03-2019  20:19         1,062,022 556.pdf
03-04-2019  12:48            31,784 6CS119 Computer Networks (1).docx
```

In [7]:
```
cd character-extraction-master
```

```
C:\Users\Dell\downloads\character-extraction-master
```

In [8]:
```
cd character-extraction-master
```

```
C:\Users\Dell\downloads\character-extraction-master\character-extraction-master
```

In [8]:
```
ls
```

```
 Volume in drive C is OS
 Volume Serial Number is 04BF-48F9

 Directory of C:\Users\Dell\downloads\character-extraction-master\character-extraction-master

10-04-2020  11:01    <DIR>          .
10-04-2020  11:01    <DIR>          ..
10-04-2020  11:01               544 .gitignore
10-04-2020  11:01           917,048 730.txt
10-04-2020  11:01             6,924 characterExtraction.py
10-04-2020  11:01             2,129 customStopWords.txt
10-04-2020  11:01             3,305 README.md
10-04-2020  11:01            43,412 reviews.csv
10-04-2020  11:01           926,085 sentenceAnalysis.json
10-04-2020  11:01           925,938 sentenceAnalysis.txt
               8 File(s)      2,825,385 bytes
               2 Dir(s)  13,860,212,736 bytes free
```

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

from collections import defaultdict
from nltk.corpus import stopwords
from pattern.en import parse, Sentence, mood
from pattern.db import csv
from pattern.vector import Document, NB
```

In [13]:
```
import nltk
```

In [19]:
```
def readText():
    """
    Reads the text from a text file.
    """
    with open("730.txt", "rb") as f:
        text = f.read().decode('utf-8-sig')
    return text
```

In [20]:
```
def chunkSentences(text):
    """
    Parses text into parts of speech tagged with parts of speech labels.

    Used for reference: https://gist.github.com/onyxfish/322906
    """
    sentences = nltk.sent_tokenize(text)
    tokenizedSentences = [nltk.word_tokenize(sentence)
                          for sentence in sentences]
    taggedSentences = [nltk.pos_tag(sentence)
                       for sentence in tokenizedSentences]
    if nltk.__version__[0:2] == "2.":
        chunkedSentences = nltk.batch_ne_chunk(taggedSentences, binary=True)
    else:
        chunkedSentences = nltk.ne_chunk_sents(taggedSentences, binary=True)
    return chunkedSentences
```

In [11]:
```
nltk.download('averaged_perceptron_tagger')
```

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

Out[11]:
```
True
```

In [21]:
```
def extractEntityNames(tree, _entityNames=None):
    """
    Creates a local list to hold nodes of tree passed through, extracting named
    entities from the chunked sentences.

    Used for reference: https://gist.github.com/onyxfish/322906
    """
    if _entityNames is None:
        _entityNames = []
    try:
        if nltk.__version__[0:2] == "2.":
            label = tree.node
        else:
            label = tree.label()
    except AttributeError:
        pass
    else:
        if label == 'NE':
            _entityNames.append(' '.join([child[0] for child in tree]))
        else:
            for child in tree:
                extractEntityNames(child, _entityNames=_entityNames)
    return _entityNames
```

In [23]:
```
def buildDict(chunkedSentences, _entityNames=None):
    """
    Uses the global entity list, creating a new dictionary with the properties
    extended by the local list, without overwriting.

    Used for reference: https://gist.github.com/onyxfish/322906
    """
    if _entityNames is None:
        _entityNames = []

    for tree in chunkedSentences:
        extractEntityNames(tree, _entityNames=_entityNames)

    return _entityNames
```

In [14]:
```
nltk.download('maxent_ne_chunker')
```

```
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping chunkers\maxent_ne_chunker.zip.
```

Out[14]: True

In [24]:
```
def removeStopwords(entityNames, customStopWords=None):
    """
    Brings in stopwords and custom stopwords to filter mismatches out.
    """
    # Memoize custom stop words
    if customStopWords is None:
        with open("customStopWords.txt", "rb") as f:
            customStopwords = f.read().split(', ')

    for name in entityNames:
        if name in stopwords.words('english') or name in customStopwords:
            entityNames.remove(name)
```

In [16]:
```
nltk.download('words')
```

```
[nltk_data] Downloading package words to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\words.zip.
```

Out[16]: True

In [25]:
```
def getMajorCharacters(entityNames):
    """
    Adds names to the major character list if they appear frequently.
    """
    return {name for name in entityNames if entityNames.count(name) > 10}
```

In [26]:
```
def splitIntoSentences(text):
    """
    Split sentences on .?! "" and not on abbreviations of titles.
    Used for reference: http://stackoverflow.com/a/8466725
    """
    sentenceEnders = re.compile(r"""
    # Split sentences on whitespace between them.
    (?:               # Group for two positive lookbehinds.
      (?<=[.!?])      # Either an end of sentence punct,
    | (?<=[.!?]['"])  # or end of sentence punct and quote.
    )                 # End group of two positive lookbehinds.
    (?<!  Mr\.   )    # Don't end sentence on "Mr."
    (?<!  Mrs\.  )    # Don't end sentence on "Mrs."
    (?<!  Ms\.   )    # Don't end sentence on "Ms."
    (?<!  Jr\.   )    # Don't end sentence on "Jr."
    (?<!  Dr\.   )    # Don't end sentence on "Dr."
    (?<!  Prof\. )    # Don't end sentence on "Prof."
    (?<!  Sr\.   )    # Don't end sentence on "Sr."
    \s+               # Split on whitespace between sentences.
    """, re.IGNORECASE | re.VERBOSE)
    return sentenceEnders.split(text)
```

In [27]:
```
def compareLists(sentenceList, majorCharacters):
    """
    Compares the list of sentences with the character names and returns
    sentences that include names.
    """
    characterSentences = defaultdict(list)
    for sentence in sentenceList:
        for name in majorCharacters:
            if re.search(r"\b(?=\w)%s\b(?!\w)" % re.escape(name),
                         sentence,
                         re.IGNORECASE):
                characterSentences[name].append(sentence)
    return characterSentences
```

In [113]:
```
def extractMood(characterSentences):
    """
    Analyzes the sentence using grammatical mood module from pattern.
    """
    characterMoods = defaultdict(list)
    for key, value in characterSentences.items():
        for x in value:
            characterMoods[key].append(mood(Sentence(parse(str(x),
                                                           lemmata=True))))
    return characterMoods
```

In [123]:
```
for key, value in characterSentences.items():
        for x in value:
            characterMoods[key].append(mood(Sentence(parse(str(x)))))
```

In [124]:
```
characterMoods
```

Out[124]:
```
defaultdict(list,
            {'Oliver': ['indicative',
              'indicative',
              'conditional',
              'indicative',
              'indicative',
              'indicative',
              'indicative',
              'conditional',
              'conditional',
              'conditional',
              'indicative',
              'conditional',
              'indicative',
              'conditional',
              'indicative',
              'indicative',
              'indicative',
              'indicative',
              'indicative',
```

In [115]:
```
characterMoods = defaultdict(list)
for key, value in characterSentences.items():
```

In [109]:
```
def extractSentiment(characterSentences):
    """
    Trains a Naive Bayes classifier object with the reviews.csv file, analyzes
    the sentence, and returns the tone.
    """
    nb = NB()
    characterTones = defaultdict(list)
    for review, rating in csv("reviews.csv"):
        nb.train(Document(review, type=int(rating), stopwords=True))
    for key, value in characterSentences.items():
        for x in value:
            characterTones[key].append(nb.classify(str(x)))
    return characterTones
```

In [30]:
```
def writeAnalysis(sentenceAnalysis):
    """
    Writes the sentence analysis to a text file in the same directory.
    """
    with open("sentenceAnalysis.txt", "wb") as f:
        for item in sentenceAnalysis.items():
            f.write("%s:%s\n" % item)
```

In [31]:
```
def writeToJSON(sentenceAnalysis):
    """
    Writes the sentence analysis to a JSON file in the same directory.
    """
    with open("sentenceAnalysis.json", "wb") as f:
        json.dump(sentenceAnalysis, f)
```

In [32]:
```
text = readText()
```

In [33]:
```
text
```

Out[33]:
```
'The Project Gutenberg EBook of Oliver Twist, by Charles Dickens\n\nThis eBook is for the use of anyone anywhere at no cost and with\nalmost no restrictions whatsoever.  You may copy it, give it away or\nre-use it under the terms of the Project Gutenberg License included\nwith this eBook or online at www.gutenberg.net\n\n\nTitle: Oliver Twist\n\nAuthor: Charles Dickens\n\nPosting Date: October 10, 2008 [EBook #730]\nRelease Date: November, 1996\n\nLanguage: English\n\n\n*** START OF THIS PROJECT GUTENBERG EBOOK OLIVER TWIST ***\n\n\n\n\nProduced by Peggy Gaugy and Leigh Little.  HTML version by Al Haines.\n\n\n\n\n\n\n\n\nOLIVER TWIST\n\nOR\n\nTHE PARISH BOY\'S PROGRESS\n\n\nBY\n\nCHARLES DICKENS\n\n\n\n\nCONTENTS\n\n\n       I  TREATS OF THE PLACE WHERE O
LIVER TWIST WAS BORN AND OF THE\n          CIRCUMSTANCES ATTENDING HIS BIRTH\n      II  TREATS OF OLIVER TWIST\'S GROWTH, EDUCA
TION, AND BOARD\n     III  RELATES HOW OLIVER TWIST WAS VERY NEAR GETTING A PLACE WHICH\n          WOULD NOT HAVE BEEN A SINECU
RE\n      IV  OLIVER, BEING OFFERED ANOTHER PLACE, MAKES HIS FIRST ENTRY INTO\n          PUBLIC LIFE\n       V  OLIVER MINGLES
WITH NEW ASSOCIATES.  GOING TO A FUNERAL FOR THE\n          FIRST TIME, HE FORMS AN UNFAVOURABLE NOTION OF HIS MASTER\'S\n
BUSINESS\n      VI  OLIVER, BEING GOADED BY THE TAUNTS OF NOAH, ROUSES INTO ACTION,\n          AND RATHER ASTONISHES HIM\n
VII  OLIVER CONTINUES REFRACTORY\n    VIII  OLIVER WALKS TO LONDON.  HE ENCOUNTERS ON THE ROAD A STRANGE\n          SORT OF YOU
NG GENTLEMAN\n      IX  CONTAINING FURTHER PARTICULARS CONCERNING THE PLEASANT OLD\n          GENTLEMAN, AND HIS HOPEFUL PUPILS
\n       X  OLIVER BECOMES BETTER ACQUAINTED WITH THE CHARACTERS OF HIS NEW\n          ASSOCIATES; AND PURCHASES EXPERIENCE AT
A HIGH PRICE. BEING A\n          SHORT, BUT VERY IMPORTANT CHAPTER, IN THIS HISTORY\n      XI  TREATS OF MR. FANG THE POLICE MA
GISTRATE; AND FURNISHES A\n          SLIGHT SPECIMEN OF HIS MODE OF ADMINISTERING JUSTICE\n     XII  IN WHICH OLIVER IS TAKEN B
ETTER CARE OF THAN HE EVER WAS\n          BEFORE.  AND IN WHICH THE NARRATIVE REVERTS TO THE MERRY OLD\n          GENTLEMAN AND
HIS YOUTHFUL FRIENDS.\n    XIII  SOME NEW ACQUAINTANCES ARE INTRODUCED TO THE INTELLIGENT READER,\n          CONNECTED WITH WHO
M VARIOUS PLEASANT MATTERS ARE RELATED,\n          APPERTAINING TO THIS HISTORY\n     XIV  COMPRISING FURTHER PARTICULARS OF OL
```

In [34]:
```
chunkedSentences = chunkSentences(text)
```

In [35]:
```
chunkedSentences
```

Out[35]:
```
<generator object ParserI.parse_sents.<locals>.<genexpr> at 0x000001AF10F219A8>
```

In [36]:
```
entityNames = buildDict(chunkedSentences)
```

In [40]:
```
len(entityNames)
```

Out[40]:
```
4414
```

In [41]:
```
entNames = list(set(entityNames))
print(entNames)
len(entNames)
```

```
['Agnes Fleming', 'Redistribution', 'Exmouth Street', 'BUSINESS Oliver', 'PERFORMED', 'Midnight', 'LIII AND', 'Master', 'XXVI I
N', 'Mr. Sikes', 'CHAPTER XXV', 'REFUND', 'CHAPTER XIII', 'STRANGE INTERVIEW', 'Kiss', 'Such', 'Move', 'Bedwin', 'Bleak', 'AND B
OARD', 'Jack Dawkins', 'Child', 'Commons', 'Deep', 'Bah', 'BROWNLOW AT', 'LIKE MISFORTUNES', 'Think', 'Chester', 'Sundays', 'Bec
ky', 'Joe', 'Nobody', 'Little Oliver', 'Remain', 'Hark', 'Merry Old', 'Wot', 'Somebody', 'PLEASANT', 'CHAPTER IX', 'Lower Hallif
ord', 'WILLIAM', 'Mr. Brownlow', 'WHICH', 'OLIVER', 'Mr. Grimwig', 'Sowerberry', 'CHAPTER XLV', 'Esquire', 'Trip', 'Artfull', 'X
XXVI', 'Mr. Crackit', 'Harry Maylie', 'Lor', 'Master Charles Bates', 'Faguey', 'Betsy', 'LONDON', 'DESTINY', 'Mr. Fang', 'Farewe
ll', 'Tom Chitling', 'Brownlow', 'Mr. Chitling', 'Inquisitive', 'Lie', '_Your_', 'Believe', 'AND THE', 'Flight', 'XXXIX', 'Londo
n', 'Angels', 'Geneva', 'Shoreditch', 'Death', 'Conkey Chickweed', 'CHAPTER III', 'Grimwig', 'XXXVII IN', 'Rome', 'Look', 'Stran
d', 'Confide', 'Yer', 'INCLUDING', 'CONNECTED', 'GUINEAS', 'SINECURE IV OLIVER', 'Covered', 'Monks', 'Donations', 'Close', 'Bibl
e', 'Mr. Harry', 'Crimson', 'BEEN', 'Oliver', 'Mr. Dawkin', 'NEW ASSOCIATES', 'Little Oliver Twist', 'Dear', 'Highgate', 'Chiswe
ll Street', 'East', 'SUDDEN', 'CHAPTER XV', 'Folly Ditch', 'Bill', 'Rotherhithe', 'Horror', 'Poor Noah', 'Hand', 'Highgate Hil
l', 'Edwin Leeford', 'SHOWS', 'Nature', 'Staunch', 'Stick', 'Jew', 'Proud', 'Mr. Charles Bates', 'Fancy', 'SCENE', 'XXIX', 'Gree
n', 'Southwark', 'Hammersmith', 'Mr. Sowerberry', 'SEND', 'Mr. Monks', 'Pentonville', 'Saint', 'INSEPARABLE', 'NOW ARRIVES', 'A
K', 'Bethnal Green', 'Thames', 'HAPPY', 'Ecod', 'Faster', 'Agnes', 'LIMITED', 'Come', 'Professor Michael', 'Harry', 'Jump', 'Mr.
Blathers', 'Spyers', 'Mr. John Dawkins', 'BOARD', 'XXXII OF THE', 'Has', 'Public Domain', 'Bet', 'Miss Rose', 'Giles', 'Smithfie
ld', 'Foolery', 'Noah', 'Crackit', 'Artful Dodger', 'Conkey', 'Mississippi', 'Christmas', 'Saint John', 'Aunt', 'Sadler', 'Run',
'SELDOM', 'Payment', 'XLIII', 'INDEMNITY', 'Speak', 'CHAPTER X', 'Nancy', 'Rose Fleming', 'Men', 'Cheap', 'PUBLIC', 'Damme', 'Cl
erkenwell', 'Jack Ketch', 'Goodness', 'Turk', 'FURNISHES', 'Nearer', 'Stay', 'Susan', 'INJURE', 'London Bridge', 'Mill Pond', 'H
ARRY', 'Sally', 'REWARD', 'METROPOLIS', 'EXHIBITING', 'Sikes', 'PURSUIT AND', 'Everything', 'Project', 'Common Garden', 'Hunge
r', 'Hendon', 'Spiders', 'Hatfield', 'CHAPTER XI', 'OLIVER AND', 'GREAT', 'Devil', 'Kensington', 'Brittle', 'Eh', 'Everybody',
'Barney', 'FLIGHT OF', 'Illness', 'Saint Magnus', 'TO', 'Lauk', 'Chertsey Bridge', 'Depend', 'Mr. Grannett', 'Duff', 'THE ROAD',
'Mr. Browlow', 'Miss Nancy', 'Brittles', 'PGLAF', 'Christian', 'Thank', 'XXXIV', 'LENGTH MEET', 'Mr. Losberne', 'LIMITED WARRANT
Y', 'ERRAND Oliver', 'White', 'Corney', 'THE HOUSE', 'Chiswick', 'France', 'Tom', 'THE POSSIBILITY OF', 'Mr. Bumble', 'Lord', 'H
ome', 'PARTICULARS', 'Rose', 'STRANGE', 'ESCAPE Near', 'Mr. Fagin', 'Ned', 'Rose Maylie', 'Mr. Sike', 'Mrs Mann', 'Bumble', 'Mut
ton Hill', 'CHAPTER VII', 'ANY PURPOSE', 'Isleworth', 'Saffron Hill', 'Bank', 'Intent', 'Mill Lane', 'DIRECT', 'MONKS AND MR.',
'Biss Dadsy', 'Mind', 'Irish', 'Mr. Noah Claypole', 'THE CIRCUMSTANCES', 'CHECK', 'Fire', 'Hyde Park', 'Wait', 'Holborn', 'Mr. W
illiam Sikes', 'GOING TO', 'XXVIII', 'Jamaica', 'Arriving', 'STRICT LIABILITY', 'Bister Fagid', 'AND', 'VIII', 'Parliament', 'XI
X IN', 'Unhand', 'THE JEW', 'Mr. Toby Crackit', 'Barbican', 'Hosier Lane', 'Darkness', 'MANY THINGS', 'Mr. Dawkins', 'Boys', 'MA
KES', 'AFTER', 'DAMAGES', 'PARAGRAPH F3', 'SEQUEL', 'PARISH BOY', 'CHAPTER XX', 'International', 'Mr. Bill Sikes', 'Newgate', 'S
uperfine', 'ADVENTURE', 'Road', 'Nance', 'DISCLAIMER OF', 'Scarce', 'Peggy Gaugy', 'Master Bates', 'Great North Road', 'Mussulma
n', 'READER IS', 'Artful', 'CHAPTER LI', 'XXIII', 'Heaven', 'Oliver White', 'Halliford', 'Dearest Rose', 'Home Affairs', 'Pharis
ee', 'XLVII', 'PURCHASES', 'MERRY', 'Copyright', 'Murder', 'Supper', 'Sit', 'Thingummy', 'MAY', 'Doctor Losberne', 'Phil Barke
r', 'Spitalfields', 'Poor Oliver', 'North End', 'Martin', 'Mr.', 'Toby', 'LADY', 'Long Lane', 'Blathers', 'Master Charley Bate
s', 'CHECK Spring', 'England', 'Break', 'THE UNPOLITENESS', 'CHAPTER IV OLIVER', 'Tom White', 'No', 'CHAPTER LII FAGIN', 'THE IN
MATES', 'BRINGS', 'River Company', 'Charles Dickens', 'Miss Maylie', 'Bartlemy', 'Chertsey', 'XXXV', 'Tommy Chitling', 'Dick',
'Ingenious', 'Sunbury Church', 'Hush', 'Bridewell', 'Horrid', 'Dodger', 'XXVII ATONES', 'Battle Bridge', 'CHAPTER', 'Shepperto
n', 'READER', 'High Street', 'XVIII', 'Doctors', 'Pooh', 'Mr. Chickweed', 'Foundation', 'Beadle', 'FOR', 'MR.', 'CHAPTER XXX',
'Newgate Calendar', 'Mercy', 'Three Cripples', 'United States', 'Twist', 'CHAPTER II', 'Walk', 'LONDON TO', 'New World', 'Char',
'Snow Hill', 'Craven Street', 'EDUCATION', 'TRADEMARK OWNER', 'Burn', 'Prayer Book', 'Quiet', 'CHAPTER V', 'Charly', 'Roman', 'P
aris', 'AND IN', 'Wales', 'BREACH OF', 'Help', 'Fagin', 'THE FULL', 'Grosvenor Square', 'Mr. Corney', 'Yonder', 'Use', 'Crown St
reet', 'Mr. Lively', 'Bolter', 'Mr. Kags', 'Email', 'Dignity', 'Lead Mills', 'THE FOUNDATION', 'God', 'Clever', 'Great', 'Camden
Town', 'AND MR. MONKS', 'Miss', 'Noah Claypole', 'West Indies', 'XLIX', 'THE BURGLARY', 'Edward Leeford', 'CONSEQUENTIAL', 'Ann
```

```
y', 'NO', 'Kags', 'SHE FAILS', 'NOT', 'Mr. Maylie', 'Mother', 'Royalty', 'Good', 'English', 'Kew Bridge', 'Thirty', 'APPERTAININ
G TO', 'EXPRESS OR IMPLIED', 'WITH', 'Please', 'eBooks', 'GROWTH', 'NOAH', 'Bill Sikes', 'HTML', 'Horses', 'Power', 'INDIRECT',
'City', 'NARRATIVE', 'Perfect', 'Unworthy', 'PUNITIVE OR', 'SOME', 'Hampstead Heath', 'Bethnal Green Road', 'XII IN', 'Tommy',
'BEING', 'Stop', 'Jack', 'Salt Lake City', 'XXXI', 'FLIGHT OF SIKES Of', 'Islington', 'BEADLE', 'Rich', 'Bow Street', 'SLIGHT',
'Island', 'Assured', 'Martha', 'Clerkinwell Sessions', 'TREATS', 'Make', 'Barnet', 'House', 'CHAPTER XLI', 'EXPERIENCES', 'Charl
ey Bates', 'Young', '_That_', 'ERRAND XV', 'Chitling', 'MOST', 'Neither', 'Madness', 'St.', 'Maylie', 'Nothing', 'Jem Spyers',
'XXXVIII', 'Which', 'John Dawkins', 'XXXIII', 'CHAPTER XLII', 'Pray', 'Hampstead', 'BECOMES', 'THE', 'Joy', 'Mealy', 'Tell', 'Ge
t', 'Project Gutenberg', 'Angel', 'Robbery', 'Sunbury', 'Mr. Morris Bolter', 'FATAL', 'Dianas', 'Patience', 'Home Department',
'Caen Wood', 'PROJECT', 'Coppice Row', 'Bench', 'REMARKABLE', 'Clerkinwell', 'Mr. Bolter', 'Compliance', 'Ben', 'Quick', 'Full P
roject', 'HAPPINESS OF', 'Hither', 'Brentford', 'Hallo', 'Truth', 'Finsbury', 'Hard', 'Body', 'Say', 'Poor', 'REDEEM', 'End', 'O
ho', 'CHAPTER VI OLIVER', 'Anythink', 'St. Paul', 'XLV', 'Family Pet', 'Michael Hart', 'Oliver Twist', 'French', 'NEW', 'Jacob',
'Six', 'Covent Garden', 'Damn', 'Mark', 'NOT BE', 'Ah', 'Mr. Duff', 'Morris Bolter', 'Surrey', 'Mr. Giles', 'BROWNLOW', 'Mann',
'St. John', 'Internal Revenue Service', 'Good Samaritan', 'DETERMINED', 'Mister Noah Claypole', 'Ratcliffe', 'Fraud', 'TAUNTS OF
NOAH', 'Edmonton', 'Acquaintance', 'Cast', 'Whether Noah Claypole', 'Write', 'Weak', 'BUT', 'Dodgers', 'Chickweed', 'INCIDENTA
L', 'Mr. Limbkins', 'Al Haines', 'Mr. Claypole', 'Blanched', 'Leigh Little', 'Fang', 'Unwin', 'Heath', 'West', 'CONTRACT', 'STAY
AT MR.', 'Whitechapel', 'ARTFUL', 'WHEN', 'U.S.', 'Old Bailey', 'PLACE', '_I_', 'Charlotte', 'Cripples', 'IRS', 'Crazy', 'Toby C
rackit', 'Bad', 'Ay', 'XXIV', 'Charley', 'Birmingham', 'Fine', 'Kingston', 'AT', 'ESCAPE LI', 'CHAPTER XVI', 'Poor Dick', 'Regar
dless', 'Neptune', 'Little Saffron Hill', 'Middlesex', 'Mr. Brittles', 'Field Lane', 'Mr. Gamfield', 'XVII OLIVER', 'Mr. Slout',
'Jove', 'Gray', 'Whittington', 'FOLLOW', 'Mister Noah', 'Miss Betsy', 'Gamfield', 'ARE DONE', 'Holborn Hill', 'Jem', 'ROUSES',
'CHAPTER XIV']
```

Out[41]: 647

In [70]:
```
from nltk.corpus import stopwords
stoplist = stopwords.words('english')
```

In [71]:
```
stoplist
```

Out[71]:
```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
```

In [50]:
```
with open('customStopWords.txt') as stopfile:
    stopwords = stopfile.read()
    list1 = stopwords.split(',')
    print(list1)
```

```
['Street', ' Road', ' Bridge', ' Town', ' Park', ' Hill', ' Lane', ' CHAPTER', ' Chapter', ' CHAPTER I', ' CHAPTER II', ' CHAPTE
R III', ' CHAPTER IV', ' CHAPTER V', ' CHAPTER VI', ' CHAPTER VII', ' CHAPTER VIII', ' CHAPTER IX', ' CHAPTER X', ' CHAPTER XI',
' CHAPTER XII', ' CHAPTER XIII', ' CHAPTER XIV', ' CHAPTER XV', ' CHAPTER XVI', ' CHAPTER XVII', ' CHAPTER XVIII', ' CHAPTER XI
X', ' CHAPTER XX', ' CHAPTER XXI', ' CHAPTER XXII', ' CHAPTER XXIII', ' CHAPTER XXIV', ' CHAPTER XXV', ' CHAPTER XXVI', ' CHAPTE
R XXVII', ' CHAPTER XXVIII', ' CHAPTER XXIX', ' CHAPTER XXX', ' CHAPTER XXXI', ' CHAPTER XXXII', ' CHAPTER XXXIII', ' CHAPTER XX
XIV', ' CHAPTER XXXV', ' CHAPTER XXXVI', ' CHAPTER XXXVII', ' CHAPTER XXXVIII', ' CHAPTER XXXIX', ' CHAPTER XL', ' CHAPTER XLI',
' CHAPTER XLII', ' CHAPTER XLIII', ' CHAPTER XLIV', ' CHAPTER XLV', ' CHAPTER XLVI', ' CHAPTER XLVII', ' CHAPTER XLVIII', ' CHAP
TER XLIX', ' CHAPTER L', ' CHAPTER LI', ' CHAPTER LII', ' CHAPTER LIII', ' CHAPTER LIV', ' CHAPTER LV', ' Chapter I', ' Chapter
II', ' Chapter III', ' Chapter IV', ' Chapter V', ' Chapter VI', ' Chapter VII', ' Chapter VIII', ' Chapter IX', '  Chapter X',
' Chapter XI', ' Chapter XII', ' Chapter XIII', ' Chapter XIV', ' Chapter XV', ' Chapter XVI', ' Chapter XVII', ' Chapter XVII
I', ' Chapter XIX', ' Chapter XX', ' Chapter XXI', ' Chapter XXII', ' Chapter XXIII', ' Chapter XXIV', ' Chapter XXV', ' Chapter
XXVI', ' Chapter XXVII', ' Chapter XXVIII', ' Chapter XXIX', ' Chapter XXX', ' Chapter XXXI', ' Chapter XXXII', ' Chapter XXXII
I', ' Chapter XXXIV', ' Chapter XXXV', ' Chapter XXXVI', ' Chapter XXXVII', ' Chapter XXXVIII', ' Chapter XXXIX', ' Chapter XL',
' Chapter XLI', ' Chapter XLII', ' Chapter XLIII', ' Chapter XLIV', ' Chapter XLV', ' Chapter XLVI', ' Chapter XLVII', ' Chapter
XLVIII', ' Chapter XLIX', ' Chapter L', ' Chapter LI', ' Chapter LII', ' Chapter LIII', ' Chapter LIV', ' Chapter LV', ' God', '
Lord', ' Heaven', ' Heavens', ' Come', ' Thank God', ' Poor', ' Tell', ' Well', ' Very', ' VERY', ' Meantime', ' No', ' Yes', '
Aye', ' Nor', ' Hark', ' Thou', ' Stand', ' Thank', ' French', ' German', ' English', ' Italian', ' Polish', ' Austrian', ' Russ
ian', ' Pray', ' Certainly', ' Listen', ' Adieu', ' Mediterranean', ' Arabian', ' Hotel', ' Island', ' Ah', ' Oh', ' Please', '
Republic', ' Take', ' WHICH', ' Which', ' Certain', ' Alas', ' Ney', ' Woe', ' Life', ' Good', ' Death', ' Was', ' Latin', ' Not
hing', ' Boulevard', ' Slang', ' Light', ' Greek', ' Place', ' Make', ' Such', ' Holy Sacrament', ' Night', ' Day', ' BOOK', ' B
ook', ' Great', ' Are', ' Guard', ' Wait', ' Him', ' Her', ' Look', ' Everything', ' Toward', ' Thy', ' Everyone', ' Every', ' P
roject', ' Project Gutenberg', '']
```

In [62]:
```
for name in entityNames:
        if name in list1:
            entityNames.remove(name)
```

In [63]:
```
for name in entNames:
        if name in list1:
            entNames.remove(name)
```

In [64]:
```
entNames
```

Out[64]:
```
['Agnes Fleming',
 'Redistribution',
 'Exmouth Street',
 'BUSINESS Oliver',
 'PERFORMED',
 'Midnight',
 'LIII AND',
 'Master',
 'XXVI IN',
 'Mr. Sikes',
 'CHAPTER XXV',
 'REFUND',
 'CHAPTER XIII',
 'STRANGE INTERVIEW',
 'Kiss',
 'Such',
 'Move',
 'Bedwin',
 'Bleak',
```

In [65]:
```
len(entNames)
```

Out[65]:
```
647
```

In [76]:
```
for name in entityNames:
        if name in stoplist:
            print(name)
            entityNames.remove(name)
```

In [77]:
```
len(entityNames)
```

Out[77]:
```
4414
```

In [78]:
```
for name in entNames:
        if name in stoplist:
            print(name)
            entNames.remove(name)
```

In [79]:
```
 majorCharacters = getMajorCharacters(entityNames)
```

In [84]:
```
majorCharacters
```

Out[84]:
```
{'AND',
 'Artful',
 'Barney',
 'Bill',
 'Blathers',
 'Brittles',
 'Bumble',
 'CHAPTER',
 'Charley',
 'Charley Bates',
 'Charlotte',
 'Chertsey',
 'Come',
 'Dodger',
 'Duff',
 'Fagin',
 'Giles',
 'God',
 'Harry',
 'Harry Maylie',
 'Heaven',
 'Jew',
 'London',
 'Master Bates',
 'Monks',
 'Mr.',
 'Mr. Bolter',
 'Mr. Brownlow',
 'Mr. Bumble',
 'Mr. Chitling',
 'Mr. Claypole',
 'Mr. Dawkins',
 'Mr. Fagin',
 'Mr. Fang',
 'Mr. Gamfield',
 'Mr. Giles',
 'Mr. Grimwig',
 'Mr. Limbkins',
 'Mr. Losberne',
 'Mr. Sikes',
 'Mr. Sowerberry',
 'Nancy',
 'Noah',
 'Noah Claypole',
```

```
 'Oliver',
 'Oliver Twist',
 'Project',
 'Rose',
 'Sikes',
 'Sowerberry',
 'THE',
 'Toby',
 'Toby Crackit',
 'Tom',
 'WHICH'}
```

In [83]:
```
print(majorCharacters)
len(majorCharacters)
type(majorCharacters)
len(majorCharacters)
```

```
{'Rose', 'Charlotte', 'Mr. Brownlow', 'WHICH', 'Sowerberry', 'Mr. Grimwig', 'Barney', 'Monks', 'Blathers', 'AND', 'Noah Claypol
e', 'Toby Crackit', 'Mr. Giles', 'Come', 'Mr. Fagin', 'Charley', 'Duff', 'Harry Maylie', 'Harry', 'Master Bates', 'Brittles', 'O
liver', 'Bumble', 'Artful', 'Mr. Fang', 'Mr. Bolter', 'Heaven', 'Mr. Sikes', 'Chertsey', 'Giles', 'Charley Bates', 'Noah', 'Mr.
Chitling', 'Mr. Gamfield', 'Bill', 'Fagin', 'Mr. Dawkins', 'Sikes', 'Mr. Losberne', 'Mr. Limbkins', 'Dodger', 'Mr. Claypole', 'J
ew', 'Project', 'CHAPTER', 'London', 'Mr. Sowerberry', 'Tom', 'God', 'Mr. Bumble', 'THE', 'Mr.', 'Nancy', 'Oliver Twist', 'Tob
y'}
```

Out[83]: 55

In [85]:
```
additional_stopwords = """AND THE WHICH Mr."""

# Split the the additional stopwords string on each word and then add
# those words to the NLTK stopwords list
stoplist += additional_stopwords.split()
```

In [96]:
```
type(majorCharacters)
s=list(majorCharacters)
```

In [99]:
```
majCharacters=set(s)
```

In [98]:
```
for name in s:
    if name in stoplist:
        s.remove(name)
```

In [100]:
```
majCharacters
```

Out[100]:
```
{'Artful',
 'Barney',
 'Bill',
 'Blathers',
 'Brittles',
 'Bumble',
 'CHAPTER',
 'Charley',
 'Charley Bates',
 'Charlotte',
 'Chertsey',
 'Come',
 'Dodger',
 'Duff',
 'Fagin',
 'Giles',
 'God',
 'Harry',
 'Harry Maylie',
 'Heaven',
 'Jew',
 'London',
 'Master Bates',
 'Monks',
 'Mr. Bolter',
 'Mr. Brownlow',
 'Mr. Bumble',
 'Mr. Chitling',
 'Mr. Claypole',
 'Mr. Dawkins',
 'Mr. Fagin',
 'Mr. Fang',
 'Mr. Gamfield',
 'Mr. Giles',
 'Mr. Grimwig',
 'Mr. Limbkins',
 'Mr. Losberne',
 'Mr. Sikes',
 'Mr. Sowerberry',
 'Nancy',
 'Noah',
 'Noah Claypole',
 'Oliver',
```

```
 'Oliver Twist',
 'Project',
 'Rose',
 'Sikes',
 'Sowerberry',
 'Toby',
 'Toby Crackit',
 'Tom'}
```

In [101]:
```
sentenceList = splitIntoSentences(text)
```

In [103]:
```
characterSentences = compareLists(sentenceList, majCharacters)
```

In [110]:
```
 characterTones = extractSentiment(characterSentences)
```

In [125]:
```
sentenceAnalysis = defaultdict(list,
                               [(k, [characterSentences[k],
                                     characterTones[k],
                                     characterMoods[k]])
                                for k in characterSentences])
```

In [126]:
```
sentenceAnalysis
```

Out[126]:
```
defaultdict(list,
            {'Oliver': [['The Project Gutenberg EBook of Oliver Twist, by Charles Dickens\n\nThis eBook is for the use of anyon
e anywhere at no cost and with\nalmost no restrictions whatsoever.',
              'You may copy it, give it away or\nre-use it under the terms of the Project Gutenberg License included\nwith thi
s eBook or online at www.gutenberg.net\n\n\nTitle: Oliver Twist\n\nAuthor: Charles Dickens\n\nPosting Date: October 10, 2008 [E
Book #730]\nRelease Date: November, 1996\n\nLanguage: English\n\n\n*** START OF THIS PROJECT GUTENBERG EBOOK OLIVER TWIST ***\n
\n\n\n\nProduced by Peggy Gaugy and Leigh Little.',
              "OLIVER TWIST\n\nOR\n\nTHE PARISH BOY'S PROGRESS\n\n\nBY\n\nCHARLES DICKENS\n\n\n\n\nCONTENTS\n\n       I  TREAT
S OF THE PLACE WHERE OLIVER TWIST WAS BORN AND OF THE\n          CIRCUMSTANCES ATTENDING HIS BIRTH\n      II  TREATS OF OLIVER
TWIST'S GROWTH, EDUCATION, AND BOARD\n     III  RELATES HOW OLIVER TWIST WAS VERY NEAR GETTING A PLACE WHICH\n          WOULD N
OT HAVE BEEN A SINECURE\n      IV  OLIVER, BEING OFFERED ANOTHER PLACE, MAKES HIS FIRST ENTRY INTO\n          PUBLIC LIFE\n
V  OLIVER MINGLES WITH NEW ASSOCIATES.",
              "GOING TO A FUNERAL FOR THE\n          FIRST TIME, HE FORMS AN UNFAVOURABLE NOTION OF HIS MASTER'S\n          BU
SINESS\n      VI  OLIVER, BEING GOADED BY THE TAUNTS OF NOAH, ROUSES INTO ACTION,\n          AND RATHER ASTONISHES HIM\n     VI
I  OLIVER CONTINUES REFRACTORY\n    VIII  OLIVER WALKS TO LONDON.",
              'HE ENCOUNTERS ON THE ROAD A STRANGE\n          SORT OF YOUNG GENTLEMAN\n      IX  CONTAINING FURTHER PARTICULAR
S CONCERNING THE PLEASANT OLD\n          GENTLEMAN, AND HIS HOPEFUL PUPILS\n       X  OLIVER BECOMES BETTER ACQUAINTED WITH THE
CHARACTERS OF HIS NEW\n          ASSOCIATES; AND PURCHASES EXPERIENCE AT A HIGH PRICE.',
              'BEING A\n          SHORT, BUT VERY IMPORTANT CHAPTER, IN THIS HISTORY\n      XI  TREATS OF MR. FANG THE POLICE
```

In [129]:
```
with open("sentenceAnalysis1.txt", "w") as f:
        for item in sentenceAnalysis.items():
            f.write("%s:%s\n" % item)
```

In [133]:
```
with open("sentenceAnalysis.json", "w") as f:
        json.dump(sentenceAnalysis, f)
```

In [141]:
```
from bookworm import *
```

In [140]:
```
!pip install python-louvain
```

```
Collecting python-louvain
  Downloading https://files.pythonhosted.org/packages/31/d4/d244fa4ca96af747b9204ce500e8919ce72eab60de2129b22a45434f1598/python-louvain-0.14.tar.gz (https://files.pythonhosted.org/packages/31/d4/d244fa4ca96af747b9204ce500e8919ce72eab60de2129b22a45434f1598/python-louvain-0.14.tar.gz)
Requirement already satisfied: networkx in c:\users\dell\anaconda3\lib\site-packages (from python-louvain) (2.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from python-louvain) (1.16.4)
Requirement already satisfied: decorator>=4.1.0 in c:\users\dell\anaconda3\lib\site-packages (from networkx->python-louvain) (4.3.0)
Building wheels for collected packages: python-louvain
  Building wheel for python-louvain (setup.py): started
  Building wheel for python-louvain (setup.py): finished with status 'done'
  Stored in directory: C:\Users\Dell\AppData\Local\pip\Cache\wheels\e7\8d\24\6b3a464bb23e96ecba3f68868e85721534fd8158a9cd7b426b
Successfully built python-louvain
Installing collected packages: python-louvain
Successfully installed python-louvain-0.14

WARNING: You are using pip version 19.1.1, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

In [137]:
```
!pip install spacy
```

```
Collecting spacy
  Downloading https://files.pythonhosted.org/packages/39/c9/6c6bbb563588cd9b5d155ebda200796911588fb46a85e5f4d21b7b2ab759/spacy-2.2.4-cp37-cp37m-win_amd64.whl (https://files.pythonhosted.org/packages/39/c9/6c6bbb563588cd9b5d155ebda200796911588fb46a85e5f4d21b7b2ab759/spacy-2.2.4-cp37-cp37m-win_amd64.whl) (9.9MB)
Collecting thinc==7.4.0 (from spacy)
  Downloading https://files.pythonhosted.org/packages/7b/5e/d7e297587f75f4932b57b47163fb92f7b479939327ca83badc0938194415/thinc-7.4.0-cp37-cp37m-win_amd64.whl (https://files.pythonhosted.org/packages/7b/5e/d7e297587f75f4932b57b47163fb92f7b479939327ca83badc0938194415/thinc-7.4.0-cp37-cp37m-win_amd64.whl) (2.1MB)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (2.19.1)
Collecting srsly<1.1.0,>=1.0.2 (from spacy)
  Downloading https://files.pythonhosted.org/packages/fd/31/edaf3cdb7fcb644a51c7a568bc62a8c8d7462e61fa79b090f4d12d73d39e/srsly-1.0.2-cp37-cp37m-win_amd64.whl (https://files.pythonhosted.org/packages/fd/31/edaf3cdb7fcb644a51c7a568bc62a8c8d7462e61fa79b090f4d12d73d39e/srsly-1.0.2-cp37-cp37m-win_amd64.whl) (179kB)
Collecting blis<0.5.0,>=0.4.0 (from spacy)
  Downloading https://files.pythonhosted.org/packages/d5/7e/1981d5389b75543f950026de40a9d346e2aec7e860b2800e54e65bd46c06/blis-0.4.1-cp37-cp37m-win_amd64.whl (https://files.pythonhosted.org/packages/d5/7e/1981d5389b75543f950026de40a9d346e2aec7e860b2800e54e65bd46c06/blis-0.4.1-cp37-cp37m-win_amd64.whl) (5.0MB)
Collecting murmurhash<1.1.0,>=0.28.0 (from spacy)
  Downloading https://files.pythonhosted.org/packages/4f/7b/d77bc9bb101e113884b2d70a118e7ec8dcc9846a35a0e10d47ca37acdcbf/murmurhash-1.0.2-cp37-cp37m-win_amd64.whl (https://files.pythonhosted.org/packages/4f/7b/d77bc9bb101e113884b2d70a118e7ec8dcc9846a35a0e10d47ca37acdcbf/murmurhash-1.0.2-cp37-cp37m-win_amd64.whl)
Collecting tqdm<5.0.0,>=4.38.0 (from spacy)
  Downloading https://files.pythonhosted.org/packages/4a/1c/6359be64e8301b84160f6f6f7936bbfaaa5e9a4eab6cbc681db07600b949/tqdm-4.45.0-py2.py3-none-any.whl (https://files.pythonhosted.org/packages/4a/1c/6359be64e8301b84160f6f6f7936bbfaaa5e9a4eab6cbc681db07600b949/tqdm-4.45.0-py2.py3-none-any.whl) (60kB)
Collecting catalogue<1.1.0,>=0.0.7 (from spacy)
  Downloading https://files.pythonhosted.org/packages/6c/f9/9a5658e2f56932e41eb264941f9a2cb7f3ce41a80cb36b2af6ab78e2f8af/catalogue-1.0.0-py2.py3-none-any.whl (https://files.pythonhosted.org/packages/6c/f9/9a5658e2f56932e41eb264941f9a2cb7f3ce41a80cb36b2af6ab78e2f8af/catalogue-1.0.0-py2.py3-none-any.whl)
Requirement already satisfied: numpy>=1.15.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (1.16.4)
Requirement already satisfied: setuptools in c:\users\dell\anaconda3\lib\site-packages (from spacy) (46.1.3)
Collecting cymem<2.1.0,>=2.0.2 (from spacy)
  Downloading https://files.pythonhosted.org/packages/84/d1/35eab0c8cc9fd9432becaf3e90144762b3201a45079e62c47a8ae8739763/cymem-2.0.3-cp37-cp37m-win_amd64.whl (https://files.pythonhosted.org/packages/84/d1/35eab0c8cc9fd9432becaf3e90144762b3201a45079e62c47a8ae8739763/cymem-2.0.3-cp37-cp37m-win_amd64.whl)
Collecting wasabi<1.1.0,>=0.4.0 (from spacy)
  Downloading https://files.pythonhosted.org/packages/21/e1/e4e7b754e6be3a79c400eb766fb34924a6d278c43bb828f94233e0124a21/wasabi-0.6.0-py3-none-any.whl (https://files.pythonhosted.org/packages/21/e1/e4e7b754e6be3a79c400eb766fb34924a6d278c43bb828f94233e0124a21/wasabi-0.6.0-py3-none-any.whl)
Collecting plac<1.2.0,>=0.9.6 (from spacy)
  Downloading https://files.pythonhosted.org/packages/86/85/40b8f66c2dd8f4fd9f09d59b22720cffecf1331e788b8a0cab5bafb353d1/plac-1.1.3-py2.py3-none-any.whl (https://files.pythonhosted.org/packages/86/85/40b8f66c2dd8f4fd9f09d59b22720cffecf1331e788b8a0cab5bafb353d1/plac-1.1.3-py2.py3-none-any.whl)
Collecting preshed<3.1.0,>=3.0.2 (from spacy)
```

```
  Downloading https://files.pythonhosted.org/packages/3c/5a/0d1b575ed40989d74fab25723083837c220246b25f3582917135cb32453f/preshed-3.0.2-cp37-cp37m-win_amd64.whl (https://files.pythonhosted.org/packages/3c/5a/0d1b575ed40989d74fab25723083837c220246b25f3582917135cb32453f/preshed-3.0.2-cp37-cp37m-win_amd64.whl) (105kB)
Requirement already satisfied: urllib3<1.24,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (1.23)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2019.3.9)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (3.0.4)
Requirement already satisfied: idna<2.8,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2.7)
Collecting importlib-metadata>=0.20; python_version < "3.8" (from catalogue<1.1.0,>=0.0.7->spacy)
  Downloading https://files.pythonhosted.org/packages/ad/e4/891bfcaf868ccabc619942f27940c77a8a4b45fd8367098955bb7e152fb1/importlib_metadata-1.6.0-py2.py3-none-any.whl (https://files.pythonhosted.org/packages/ad/e4/891bfcaf868ccabc619942f27940c77a8a4b45fd8367098955bb7e152fb1/importlib_metadata-1.6.0-py2.py3-none-any.whl)
Collecting zipp>=0.5 (from importlib-metadata>=0.20; python_version < "3.8"->catalogue<1.1.0,>=0.0.7->spacy)
  Downloading https://files.pythonhosted.org/packages/b2/34/bfcb43cc0ba81f527bc4f40ef41ba2ff4080e047acb0586b56b3d017ace4/zipp-3.1.0-py3-none-any.whl (https://files.pythonhosted.org/packages/b2/34/bfcb43cc0ba81f527bc4f40ef41ba2ff4080e047acb0586b56b3d017ace4/zipp-3.1.0-py3-none-any.whl)
Installing collected packages: tqdm, cymem, murmurhash, preshed, blis, zipp, importlib-metadata, catalogue, srsly, wasabi, plac, thinc, spacy
  Found existing installation: tqdm 4.26.0
    Uninstalling tqdm-4.26.0:
      Successfully uninstalled tqdm-4.26.0
Successfully installed blis-0.4.1 catalogue-1.0.0 cymem-2.0.3 importlib-metadata-1.6.0 murmurhash-1.0.2 plac-1.1.3 preshed-3.0.2 spacy-2.2.4 srsly-1.0.2 thinc-7.4.0 tqdm-4.45.0 wasabi-0.6.0 zipp-3.1.0


WARNING: You are using pip version 19.1.1, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

In [226]:
```
book = load_book('730.txt', lower=True)
```

In [227]:
```
book
```

```
tion. \'it\'s very\nlikely it _will_ be troublesome. give it a little gruel if it is.\'  he\nput on his hat, and, pausing by th
e bed-side on his way to the door,\nadded, \'she was a good-looking girl, too; where did she come from?\'\n\n\'she was brought
here last night,\' replied the old woman, \'by the\noverseer\'s order.  she was found lying in the street.  she had walked\nsom

e distance, for her shoes were worn to pieces; but where she came\nfrom, or where she was going to, nobody knows.\'\n\nthe surg
eon leaned over the body, and raised the left hand.  \'the old\nstory,\' he said, shaking his head: \'no wedding-ring, i see. a
h!\ngood-night!\'\n\nthe medical gentleman walked away to dinner; and the nurse, having once\nmore applied herself to the green
bottle, sat down on a low chair\nbefore the fire, and proceeded to dress the infant.\n\nwhat an excellent example of the power
of dress, young oliver twist\nwas!  wrapped in the blanket which had hitherto formed his only\ncovering, he might have been the
child of a nobleman or a beggar; it\nwould have been hard for the haughtiest stranger to have assigned him\nhis proper station
in society.  but now that he was enveloped in the\nold calico robes which had grown yellow in the same service, he was\nbadged
and ticketed, and fell into his place at once--a parish\nchild--the orphan of a workhouse--the humble, half-starved drudge--to
\nbe cuffed and buffeted through the world--despised by all, and pitied\nby none.\n\noliver cried lustily. if he could have kno
wn that he was an orphan,\nleft to the tender mercies of church-wardens and overseers, perhaps he\nwould have cried the loude
r.\n\n\n\nchapter ii\n\ntreats of oliver twist\'s growth, education, and board\n\nfor the next eight or ten months, oliver was
the victim of a systematic\ncourse of treachery and deception.  he was brought up by hand.  the\nhungry and destitute situation
of the infant orphan was duly reported\nby the workhouse authorities to the parish authorities.  the parish\nauthorities inquir
ed with dignity of the workhouse authorities, whether\nthere was no female then domiciled in \'the house\' who was in a\nsituat
ion to impart to oliver twist, the consolation and nourishment of\nwhich he stood in need.  the workhouse authorities replied w
ith\nhumility, that there was not. upon this, the parish authorities\nmagnanimously and humanely resolved, that oliver should b
```

In [178]:
```
characters = s
```

In [179]:
```
characters
```

Out[179]:
```
['Rose',
 'Charlotte',
 'Mr. Brownlow',
 'Sowerberry',
 'Mr. Grimwig',
 'Barney',
 'Monks',
 'Blathers',
 'Noah Claypole',
 'Toby Crackit',
 'Mr. Giles',
 'Come',
 'Mr. Fagin',
 'Charley',
 'Duff',
 'Harry Maylie',
 'Harry',
 'Master Bates',
 'Brittles',
 'Oliver',
 'Bumble',
 'Artful',
 'Mr. Fang',
 'Mr. Bolter',
 'Heaven',
 'Mr. Sikes',
 'Chertsey',
 'Giles',
 'Charley Bates',
 'Noah',
 'Mr. Chitling',
 'Mr. Gamfield',
 'Bill',
 'Fagin',
 'Mr. Dawkins',
 'Sikes',
 'Mr. Losberne',
 'Mr. Limbkins',
 'Dodger',
 'Mr. Claypole',
 'Jew',
 'Project',
 'CHAPTER',
 'London',
```

```
 'Mr. Sowerberry',
 'Tom',
 'God',
 'Mr. Bumble',
 'Nancy',
 'Oliver Twist',
 'Toby']
```

In [228]:
```
sequences = get_sentence_sequences(book)
```

In [325]:
```
type(sequences)
```

Out[325]:
```
list
```

In [181]:
```
df = find_connections(sequences, characters)
```

In [183]:
```
cooccurence = calculate_cooccurence(df)
```

In [184]:
```
cooccurence
```

Out[184]:

	Rose	Charlotte	Mr. Brownlow	Sowerberry	Mr. Grimwig	Barney	Monks	Blathers	Noah Claypole	Toby Crackit	...	Project	CHAPTER	London	So
Rose	0	18935825	19519705	17232928	16383511	10655760	6533545	15722747	24928627	19616627	...	11537792	0	10298755	
Charlotte	18935825	0	46501207	40943077	39127717	25413386	15434856	37600600	59519403	46929173	...	27517350	0	24448961	
Mr. Brownlow	19519705	46501207	0	42295754	40699032	26192035	16052259	38479652	61484720	48564637	...	28282837	0	25480444	
Sowerberry	17232928	40943077	42295754	0	35581247	23152539	13968562	33971196	53773182	42423893	...	24998449	0	22139431	
Mr. Grimwig	16383511	39127717	40699032	35581247	0	22078958	13453743	32469383	51651900	41007023	...	23777014	0	21275280	
Barney	10655760	25413386	26192035	23152539	22078958	0	8707117	21113301	33470443	26389951	...	15443731	0	13809863	
Monks	6533545	15434856	16052259	13968562	13453743	8707117	0	12789792	20402937	16120379	...	9390515	0	8563766	
Blathers	15722747	37600600	38479652	33971196	32469383	21113301	12789792	0	49331162	38850779	...	22763857	0	20174114	
Noah Claypole	24928627	59519403	61484720	53773182	51651900	33470443	20402937	49331162	0	61926074	...	36043223	0	32339601	
Toby Crackit	19616627	46929173	48564637	42423893	41007023	26389951	16120379	38850779	61926074	0	...	28488130	0	25523268	
Mr. Giles	17224800	40975476	42393137	37221780	35896469	23098306	14060959	34063155	54082369	42720290	...	24884789	0	22177044	
Come	6944817	16445686	16966775	14988119	14231860	9256946	5645194	13613551	21653870	17032602	...	10030706	0	8958486	
Mr. Fagin	15454293	36954263	38359486	33454453	32512633	20900342	12740394	30665166	48862839	38704495	...	22405319	0	20180932	
Charley	11713935	28018924	28747712	25433173	24228656	15805625	9509156	23294001	36879595	28973517	...	16967130	0	15052250	
Duff	2183051	5185692	5368208	4708433	4509900	2916720	1789652	4290347	6845906	5407511	...	3160880	0	2837294	
Harry Maylie	22781235	54447818	56293624	49450836	47593305	30788429	18619914	45222611	71938207	56796443	...	33019428	0	29469016	
Harry	6540838	15668554	16223971	14363315	13704186	8914185	5343436	13020253	20637359	16331605	...	9529772	0	8464438	
Master Bates	29547037	70525862	72415591	63758991	61147930	39677809	24091471	58578581	92710008	73184584	...	42758541	0	37996802	
Brittles	16192897	38727524	39658755	34968878	33567608	21690446	13194845	32126398	50678987	40090018	...	23499612	0	20804216	
Oliver	9221997	21965272	22642409	20034617	19192237	12400810	7487950	18259333	28891666	22798126	...	13363829	0	11842320	
Bumble	7086087	16851070	17310863	15340373	14585494	9495535	5735396	13998116	22190562	17428130	...	10237871	0	9078367	
Artful	7218949	17342000	17811437	15674748	15008242	9699036	5892294	14348694	22709279	17956529	...	10525875	0	9328647	
Mr. Fang	13380657	31996530	33214160	28989914	28068625	18109702	11021533	26548025	42322870	33472753	...	19398293	0	17471631	

	Rose	Charlotte	Mr. Brownlow	Sowerberry	Mr. Grimwig	Barney	Monks	Blathers	Noah Claypole	Toby Crackit	...	Project	CHAPTER	London	So
Mr. Bolter	20185149	48179749	49791179	43769319	41938446	27080188	16465796	39909022	63416296	50087223	...	29297269	0	26084575	
Heaven	12418914	29548869	30319988	26862919	25551338	16750236	10096137	24564235	38876688	30569227	...	17952433	0	16003990	
Mr. Sikes	16270613	38675982	40028703	35161412	33917605	21818775	13293984	32149408	51023905	40362004	...	23507544	0	20954029	
Chertsey	16578390	39492016	40390667	35864270	34079089	22204907	13424780	32815012	51712922	40747850	...	23991743	0	21181442	
Giles	8939974	21204129	21803924	19228199	18487601	11943189	7276171	17663977	27921372	22003261	...	12868669	0	11438023	
Charley Bates	28702088	68567021	70461542	62034449	59450827	38605269	23381296	56959917	90294658	71138566	...	41533879	0	36940695	
Noah	6665341	15963872	16436276	14346095	13775546	8945752	5476176	13215219	21072731	16566158	...	9642493	0	8690032	
Mr. Chitling	21064749	50471747	52151934	45519368	44234439	28356123	17320256	41840934	66485715	52660913	...	30572596	0	27413852	
Mr. Gamfield	20572025	49095907	50753943	44539912	42936234	27710247	16806439	40772785	64848804	51172959	...	29779839	0	26625064	
Bill	4448476	10667953	11030200	9583997	9371342	5973912	3651087	8865419	14134735	11122360	...	6432242	0	5766693	
Fagin	7169467	17182916	17770273	15460872	15103765	9745225	5955606	14265988	22701842	17987466	...	10389199	0	9441911	
Mr. Dawkins	17603796	41982019	43576863	38016734	36893913	23723011	14502796	34878952	55510789	43945292	...	25450141	0	22892558	
Sikes	7985787	18904635	19439490	17167831	16508737	10663658	6509196	15750230	24862908	19644975	...	11491424	0	10215008	
Mr. Losberne	24559847	58321165	60332655	53176920	50844413	32943462	20040317	48412848	76843318	60665309	...	35514513	0	31702655	
Mr. Limbkins	17882145	42580862	44236195	38605565	37588780	24045439	14737362	35363263	56266154	44654456	...	25857189	0	23253861	
Dodger	10039672	23846335	24627403	21781651	20693818	13459395	8175944	19763423	31363711	24706965	...	14537070	0	13002367	
Mr. Claypole	20297936	48410199	50103775	43936910	42161159	27285892	16580106	40150623	64026478	50411150	...	29375880	0	26266517	
Jew	4544726	10784896	11063994	9852277	9307372	6086320	3658696	8968578	14153877	11107702	...	6562264	0	5788997	
Project	11537792	27517350	28282837	24998449	23777014	15443731	9390515	22763857	36043223	28488130	...	0	0	14886612	
CHAPTER	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
London	10298755	24448961	25480444	22139431	21275280	13809863	8563766	20174114	32339601	25523268	...	14886612	0	0	
Mr. Sowerberry	25517754	60714424	62884967	55592388	52990115	34307656	20753350	50370374	79934179	63140922	...	37014569	0	32878452	
Tom	3114676	7371804	7699112	6691545	6423363	4128205	2571301	6064730	9762196	7701653	...	4499400	0	4094446	
God	3741532	8869828	9232472	8044740	7683843	4979030	3084574	7313116	11740864	9237495	...	5400523	0	4941344	
Mr. Bumble	15370913	36622417	37900076	33333954	31994362	20650652	12520184	30397294	48351559	38145159	...	22253991	0	19817388	

	Rose	Charlotte	Mr. Brownlow	Sowerberry	Mr. Grimwig	Barney	Monks	Blathers	Noah Claypole	Toby Crackit	...	Project	CHAPTER	London	So
Nancy	5788988	13860565	14325727	12508854	12068701	7893175	4797322	11501957	18352936	14500977	...	8401586	0	7621281	
Oliver Twist	22825179	54456626	56232955	49296215	47656790	30612527	18660381	45210192	71727937	56750915	...	33042977	0	29446741	
Toby	3395340	8045258	8398658	7329365	6982425	4520167	2800071	6620056	10670666	8424399	...	4906840	0	4460439	

51 rows × 51 columns

In [182]:
```
df
```

Out[182]:

	Rose	Charlotte	Mr. Brownlow	Sowerberry	Mr. Grimwig	Barney	Monks	Blathers	Noah Claypole	Toby Crackit	...	Project	CHAPTER	London	Mr. Sowerberry
"and oh, kind heaven!"	3	7	8	5	5	7	5	6	13	8	...	3	0	10	8
"brittles," i says, when i had woke him, "don't be frightened!"'	11	25	26	25	28	14	9	22	31	29	...	14	0	13	37
"come!"	2	2	2	3	1	1	1	1	3	2	...	3	0	2	3
"d--me!"	1	1	0	2	1	1	0	1	1	0	...	1	0	1	2
"don't go to him," i called out of the window, "he's an assassin!	14	26	34	14	24	11	15	21	42	31	...	14	0	23	26
"have you?"	2	4	3	4	1	3	1	3	8	4	...	2	0	2	5

In [153]:
```
cooccurence = cooccurence.to_sparse()
```

In [156]:

```
def print_five_closest(character):
    print('-'*len(str(character))
          + '\n' + str(character) + '\n'
          + '-'*len(str(character)))

    top_five = (cooccurence[str(character)]
                .sort_values(ascending=False)
                .index.values
                [:5])

    for name in top_five:
        print(name)
```

In [157]:
```
from random import randint

for i in range(5):
    print_five_closest(characters[randint(0, len(characters))])
    print()
```

```
---------
Mr. Fagin
---------
Master Bates
Charley Bates
Mr. Sowerberry
Noah Claypole
Mr. Losberne

-----
Harry
-----
Master Bates
Charley Bates
Mr. Sowerberry
Noah Claypole
Mr. Losberne

------------
Oliver Twist
------------
Master Bates
Charley Bates
Mr. Sowerberry
Noah Claypole
Mr. Losberne

------
Bumble
------
Master Bates
Charley Bates
Mr. Sowerberry
Noah Claypole
Mr. Losberne

-----
Fagin
-----
Master Bates
```

```
Charley Bates
Mr. Sowerberry
Noah Claypole
Mr. Losberne
```

In [158]:
```
print_five_closest('Mr. Losberne')
```

```
------------
Mr. Losberne
------------
Master Bates
Charley Bates
Mr. Sowerberry
Noah Claypole
Harry Maylie
```

In [186]:
```
from bookworm import *
```

In [191]:
```
!pip install spacy
```

```
Requirement already satisfied: spacy in c:\users\dell\anaconda3\lib\site-packages (2.2.4)
Requirement already satisfied: thinc==7.4.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (7.4.0)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (1.1.3)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (3.0.2)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (1.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (2.19.1)
Requirement already satisfied: blis<0.5.0,>=0.4.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (0.4.1)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (1.0.0)
Requirement already satisfied: setuptools in c:\users\dell\anaconda3\lib\site-packages (from spacy) (46.1.3)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (4.45.0)
Requirement already satisfied: numpy>=1.15.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (1.16.4)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (0.6.0)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (2.0.3)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy) (1.0.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->sp
acy) (2019.3.9)
Requirement already satisfied: urllib3<1.24,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0-
>spacy) (1.23)
Requirement already satisfied: idna<2.8,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy)
(2.7)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0-
>spacy) (3.0.4)
Requirement already satisfied: importlib-metadata>=0.20; python_version < "3.8" in c:\users\dell\anaconda3\lib\site-packages (fr
om catalogue<1.1.0,>=0.0.7->spacy) (1.6.0)
Requirement already satisfied: zipp>=0.5 in c:\users\dell\anaconda3\lib\site-packages (from importlib-metadata>=0.20; python_ver
sion < "3.8"->catalogue<1.1.0,>=0.0.7->spacy) (3.1.0)

WARNING: You are using pip version 19.1.1, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

In [196]:
```
import networkx as nx

import spacy
import nltk
import string
```

In [257]:
```
len(book.split())
```

Out[257]:
```
161009
```

In [286]:
```
chunkedSentences = chunkSentences(text)
```

In [287]:
```
entityNames2 = buildDict(chunkedSentences)
```

In [291]:
```
len(entityNames2)
len(words3)
```

Out[291]: 5736

In [290]:
```
words3 = [remove_punctuation(p) for p in entityNames2]
```

In [304]:
```
words3
```

```
'Albus Dumbledore',
'Twelve',
'Dumbledore',
'Professor',
'Harry Potter',
'Philosophers Stone',
'Professor McGonagall',
'Dursleys',
'Kent',
'Dedalus Diggle',
'Dumbledore',
'Professor',
'Harry Potter',
'Philosophers Stone',
'Muggle',
'Dumbledore',
'Muggles',
'Dumbledore',
'Dumbledore',
'Muggle',
```

In [265]:
```
remove_punctuation = lambda s: s.translate(str.maketrans('', '', string.punctuation+'’'))
words = [remove_punctuation(p) for p in book.split()]

#unique_words = list(set(words))
#len(unique_words)
len(words)
```

Out[265]: 161009

In [212]:
```
!python -m spacy download en
```

```
Requirement already satisfied: en_core_web_sm==2.2.5 from https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.2.5/en_core_web_sm-2.2.5.tar.gz#egg=en_core_web_sm==2.2.5 (https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.2.5/en_core_web_sm-2.2.5.tar.gz#egg=en_core_web_sm==2.2.5) in c:\users\dell\anaconda3\lib\site-packages (2.2.5)
Requirement already satisfied: spacy>=2.2.2 in c:\users\dell\anaconda3\lib\site-packages (from en_core_web_sm==2.2.5) (2.2.4)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.2)
Requirement already satisfied: setuptools in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (46.1.3)
Requirement already satisfied: thinc==7.4.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (7.4.0)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (2.19.1)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (0.6.0)
Requirement already satisfied: blis<0.5.0,>=0.4.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (0.4.1)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (3.0.2)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.0)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (2.0.3)
Requirement already satisfied: numpy>=1.15.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.16.4)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.2)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (4.45.0)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in c:\users\dell\anaconda3\lib\site-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.1.3)
Requirement already satisfied: urllib3<1.24,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.5) (1.23)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.5) (2019.3.9)
Requirement already satisfied: idna<2.8,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.5) (2.7)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\dell\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.5) (3.0.4)
Requirement already satisfied: importlib-metadata>=0.20; python_version < "3.8" in c:\users\dell\anaconda3\lib\site-packages (from catalogue<1.1.0,>=0.0.7->spacy>=2.2.2->en_core_web_sm==2.2.5) (1.6.0)
Requirement already satisfied: zipp>=0.5 in c:\users\dell\anaconda3\lib\site-packages (from importlib-metadata>=0.20; python_version < "3.8"->catalogue<1.1.0,>=0.0.7->spacy>=2.2.2->en_core_web_sm==2.2.5) (3.1.0)
[+] Download and installation successful
You can now load the model via spacy.load('en_core_web_sm')
```

```
[x] Couldn't link model to 'en'
Creating a symlink in spacy/data failed. Make sure you have the required
permissions and try re-running the command as admin, or use a virtualenv. You
can still import the model as a module and call its load() method, or create the
symlink manually.
C:\Users\Dell\Anaconda3\lib\site-packages\en_core_web_sm -->
C:\Users\Dell\Anaconda3\lib\site-packages\spacy\data\en
[!] Download successful but linking failed
Creating a shortcut link for 'en' didn't work (maybe you don't have admin
permissions?), but you can still load the model via its full package name: nlp =
spacy.load('en_core_web_sm')


WARNING: You are using pip version 19.1.1, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
You do not have sufficient privilege to perform this operation.
```

In [231]:
```
import spacy
import en_core_web_sm
nlp = en_core_web_sm.load()
```

In [293]:
```
char=[word.text for word in nlp(' '.join(words3)) if word.pos_ == 'PROPN']
len(char)
```

Out[293]: 6966

In [266]:
```
candidates = [word.text for word in nlp(' '.join(words)) if word.pos_ == 'PROPN']
len(candidates)
```

Out[266]: 6349

In [267]:
```
candidates = [c for c in candidates if len(c) > 2]
len(candidates)
```

Out[267]: 5201

In [294]:
```
char = [c for c in char if len(c) > 2]
len(char)
```

Out[294]: 6890

In [262]:
```
candidates
```

```
 'mrs',
 'mann',
 'bumble',
 'mrs',
 'mann',
 'mrs',
 'mann',
 'mrs',
 'mann',
 'bumble',
 'bumble',
 'mrs',
 'mann',
 'bumble',
 'bumble',
 'mrs',
 'mann',
 'bumble',
 'mrs',
 'mann'
```

In [320]:
```
text
```

Out[320]:
```
'/ \n\n\n\n\nTHE BOY WHO LIVED \n\nMr. and Mrs. Dursley, of number four, Privet Drive, \nwere proud to say that they were perfe
ctly normal, \nthank you very much. They were the last people you'd \nexpect to be involved in anything strange or \nmysteriou
s, because they just didn't hold with such \nnonsense. \n\nMr. Dursley was the director of a firm called \nGrunnings, which mad
e drills. He was a big, beefy \nman with hardly any neck, although he did have a \nvery large mustache. Mrs. Dursley was thin a
nd \nblonde and had nearly twice the usual amount of \nneck, which came in very useful as she spent so \nmuch of her time crani
ng over garden fences, spying \non the neighbors. The Dursley s had a small son \ncalled Dudley and in their opinion there was
no finer \nboy anywhere. \n\nThe Dursleys had everything they wanted, but they \nalso had a secret, and their greatest fear was
that \nsomebody would discover it. They didn't think they \ncould bear it if anyone found out about the Potters. \nMrs. Potter
was Mrs. Dursley's sister, but they hadn't \n\nPage | 2 Harry Potter and the Philosophers Stone - J.K. Rowling \n\n\n\nmet for
several years; in fact, Mrs. Dursley pretended \nshe didn't have a sister, because her sister and her \ngood-for-nothing husban
d were as unDursleyish as it \nwas possible to be. The Dursleys shuddered to think \nwhat the neighbors would say if the Potter
s arrived in \nthe street. The Dursleys knew that the Potters had a \nsmall son, too, but they had never even seen him. \n\nThi
s boy was another good reason for keeping the \nPotters away; they didn't want Dudley mixing with a \nchild like that. \n\nWhen
Mr. and Mrs. Dursley woke up on the dull, gray \nTuesday our story starts, there was nothing about the \ncloudy sky outside to
suggest that strange and \nmysterious things would soon be happening all over \nthe country. Mr. Dursley hummed as he picked ou
t \nhis most boring tie for work, and Mrs. Dursley \ngossiped away happily as she wrestled a screaming \nDudley into his high c
hair. \n\nNone of them noticed a large, tawny owl flutter past \nthe window. \n\nAt half past eight, Mr. Dursley picked up his
\nbriefcase, pecked Mrs. Dursley on the cheek, and \ntried to kiss Dudley good-bye but missed, because \nDudley was now having
a tantrum and throwing his \ncereal at the walls. "Little tyke," chortled Mr. Dursley \nas he left the house. He got into his c
```

In [215]:
```
candidates = [c for c in candidates if c.istitle()]
len(candidates)
```

Out[215]: 0

In [297]:
```
char = [c for c in char if c.istitle()]
len(char)
```

Out[297]: 6790

In [268]:
```
candidates = [c for c in candidates if not (c[-1] == 's' and c[:-1] in candidates)]
len(candidates)
```

Out[268]: 5054

In [298]:
```
char = [c for c in char if not (c[-1] == 's' and c[:-1] in char)]
len(char)
```

Out[298]: 6694

In [269]:
```
majorCharacters2 = getMajorCharacters(candidates)
```

In [299]:
```
majorCharacters3 = getMajorCharacters(char)
```

In [357]:
```
def getMajorCharacters(entityNames):
    """
    Adds names to the major character list if they appear frequently.
    """
    return {name for name in entityNames if entityNames.count(name) > 100}
```

In [358]:
```
majorCharacters4 = getMajorCharacters(words3)
```

In [316]:
```
for name in words3:
    if(len(name)>5):
        print(words3.count(name))
        print(name)
```

```
Yorkshire
1
Dundee
1
Bonfire Night
2
Britain
368
Harry Potter
347
Philosophers Stone
27
Mr Dursley
27
Mr Dursley
127
Dudley
1
Howard
27
```

In [359]:
```
majorCharacters4
```

Out[359]:
```
{'Dudley',
 'Dumbledore',
 'Hagrid',
 'Harry',
 'Harry Potter',
 'Hermione',
 'Malfoy',
 'Neville',
 'Page',
 'Philosophers Stone',
 'Ron',
 'Snape'}
```

In [360]:
```
len(majorCharacters4)
```

Out[360]:
```
12
```

In [273]:
```
stopwords = nltk.corpus.stopwords.words('english')

candidates = list(set([c.title() for c in [c.lower() for c in majorCharacters2]]) - set(stopwords))

len(candidates)
```

Out[273]: 56

In [302]:
```
stopwords = nltk.corpus.stopwords.words('english')

char = list(set([c.title() for c in [c.lower() for c in majorCharacters3]]) - set(stopwords))

len(char)
```

Out[302]: 69

In [350]:
```
len(candidates)
```

Out[350]: 56

In [351]:
```
char[:10]
```

Out[351]:
```
['Norbert',
 'Crabbe',
 'Sorcerer',
 'Hogwarts',
 'Page',
 'Ollivander',
 'Nicolas',
 'Slytherin',
 'Gryffindor',
 'Muggle']
```

In [275]:
```
candidates1 = list(set(candidates))
```

In [276]:
```
characters = [tuple([character + ' ']) for character in set(candidates)]
```

In [281]:
```
characters=candidates1
```

In [251]:
```
 majorCharacters=s
```

In [245]:
```
len(candidates1)
```

Out[245]: 1861

In [326]:
```
sequences = get_sentence_sequences(text)
```

In [328]:
```
type(sequences)
```

Out[328]: list

In [361]:
```
df = find_connections(sequences, majorCharacters4 )
cooccurence = calculate_cooccurence(df)
```

In [362]:
```
df
```

Cloak.												
“You’re going out again,” he said.	11	29	18	8	7	11	5	6	7	5	5	13
“You’re in luck, Weasley, Potter’s obviously spotted \nsome money on the ground!” said Malfoy.	43	94	58	13	28	13	14	23	16	13	23	38
“You’re losing it, too,” said Ron.	11	35	19	3	7	7	8	7	4	3	4	13
“You’re lucky you weren’t killed.	19	26	20	4	15	4	3	7	5	6	13	14
“You’re saying it wrong,” Harry heard Hermione snap.	20	45	42	10	11	18	7	9	13	20	8	27
“You’re the only one who knows how to get past \nFluffy, aren’t you, Hagrid?” said Harry anxiously.	33	89	68	13	19	20	14	25	17	23	19	34
“You’re too nosy to live, Potter.	16	41	30	4	10	3	7	8	4	5	6	16
“You’ve got to eat some breakfast.” \n\n“I don’t want anything.” \n\n“Just a bit of toast,” wheedled	38	92	70	18	24	18	13	19	21	13	16	39

In [353]:
```
df
```

Out[353]:

	Page	Gryffindor	Snape	Harry	Philosophers Stone	Quidditch	Neville	Wood	Malfoy	Hermione	Dursleys	Dumbledore	Harry Potter	Hagrid	Ron	D
"... and until Hagrid told me, I didn't know anything \nabout being a wizard or about my parents or \nVoldemort — " \n\nRon gasped.	18	46	25	24	100	40	22	26	23	44	23	45	83	34	19	
"... d-don't know why you wanted t-t-to meet here of \nall p-places, Severus ..." \n\n"Oh, I thought we'd keep this private," said Snape, his \nvoice	21	33	31	15	131	45	43	19	20	49	36	50	84	21	11	

In [283]:
```
df
```

the poor man, "but we're sure to have him; for between ten and\neleven o'clock at night he passed again."	37	35	98	50	32	75	25	72	14	30	...	83	94	50	15	14	32
"in\nshort, fagin," he says, "jew as you are, you never laid such snares as\ni'll contrive for my young brother, oliver.'"	25	18	47	36	21	60	22	48	15	19	...	49	44	33	9	9	25
"my heart!"	1	1	6	4	1	6	0	6	1	1	...	2	4	3	1	0	2
"oh no, you won't," says our overseer.'	12	5	16	11	6	24	6	13	3	5	...	20	13	13	6	6	4
"oh, my dear																

In [284]:
```
interaction_df = get_interaction_df(cooccurence, threshold=1)
interaction_df.sample(5)
```

Out[284]:

	source	target	value
1075	Noah	Sunday	6123972
125	Charlotte	Ill	5710220
1476	Bolter	Waistcoat	25686965
1279	Maam	Matron	7922829
120	Charlotte	Gutenbergtm	47166812

In [363]:
```
interaction_df = get_interaction_df(cooccurence, threshold=1)
interaction_df.sample(5)
```

Out[363]:

	source	target	value
3	Dumbledore	Neville	8201305
39	Neville	Ron	2197620
53	Ron	Harry	1338222
31	Page	Hagrid	2469964
47	Hagrid	Snape	3029599

In [354]:
```
interaction_df = get_interaction_df(cooccurence, threshold=1)
interaction_df.sample(5)
```

Out[354]:

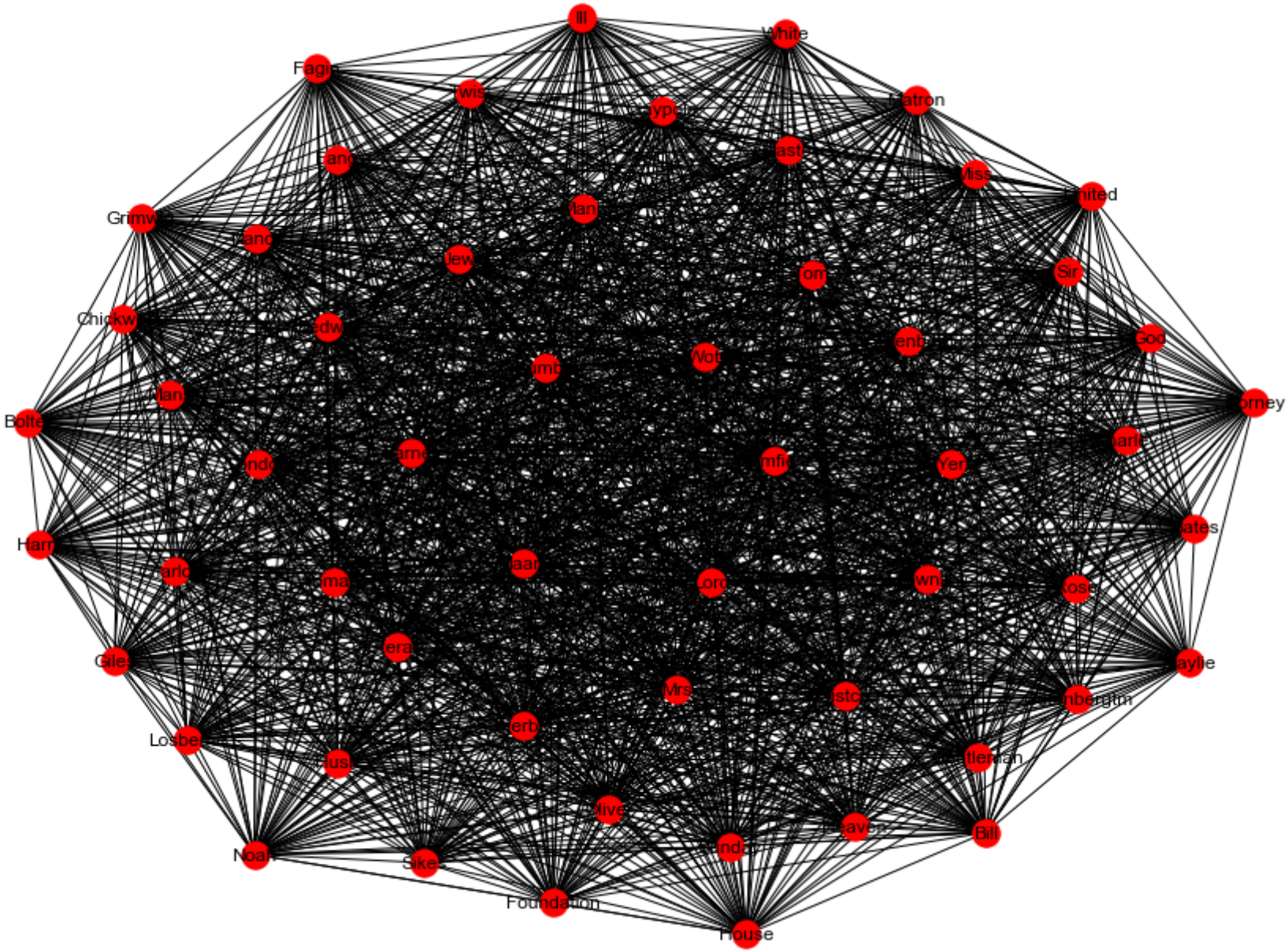
	source	target	value
18	Page	Uncle Vernon	7659460
45	Snape	Dumbledore	5783693
180	Dudley	Aunt Petunia	8572110
92	Quidditch	Hagrid	4334471
99	Neville	Wood	3067386

In [321]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (12,9)
```

In [364]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (24,18)
```

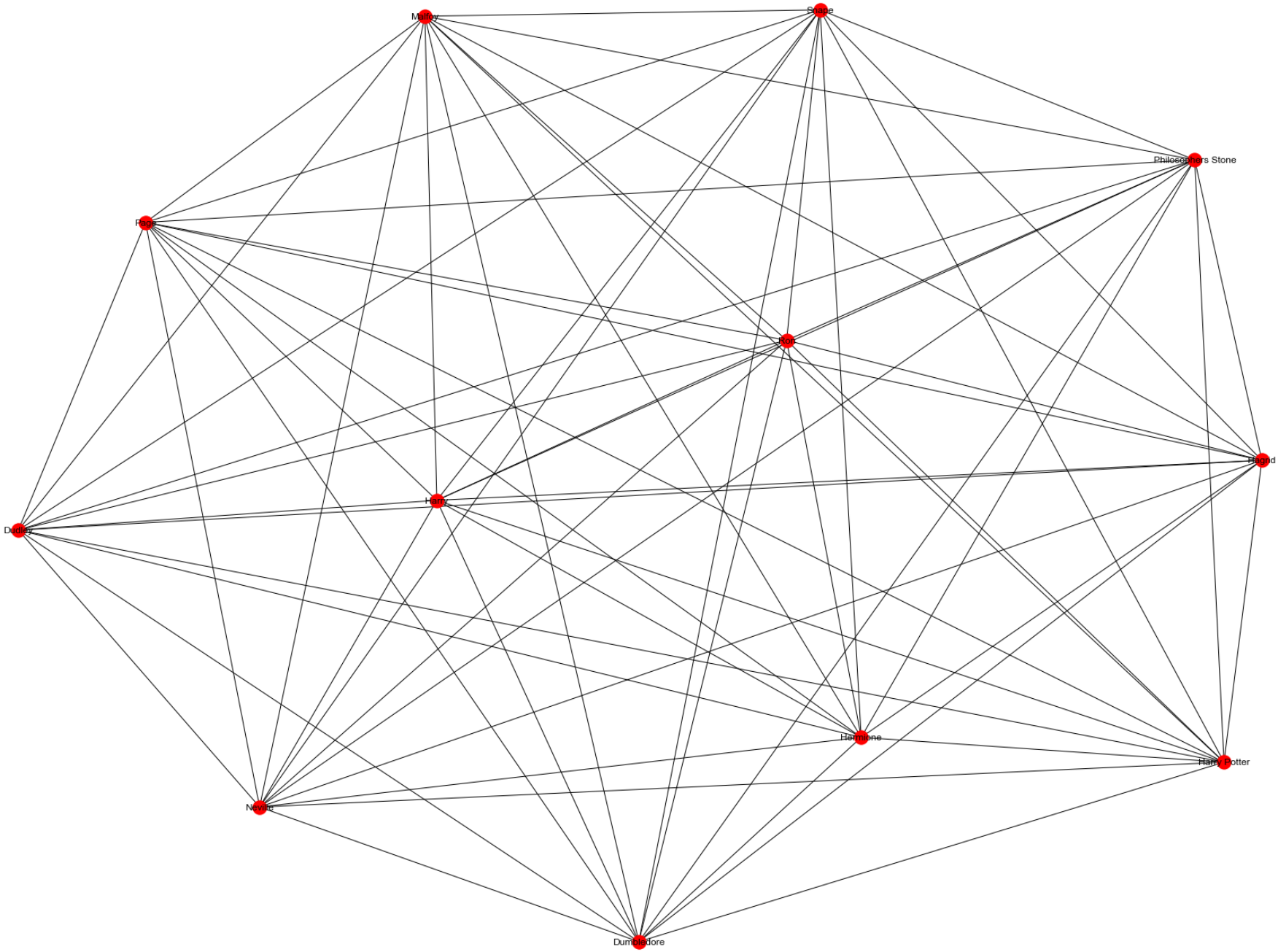
In [323]:
```
G = nx.from_pandas_edgelist(interaction_df,
                            source='source',
                            target='target')

nx.draw(G, with_labels=True)
```



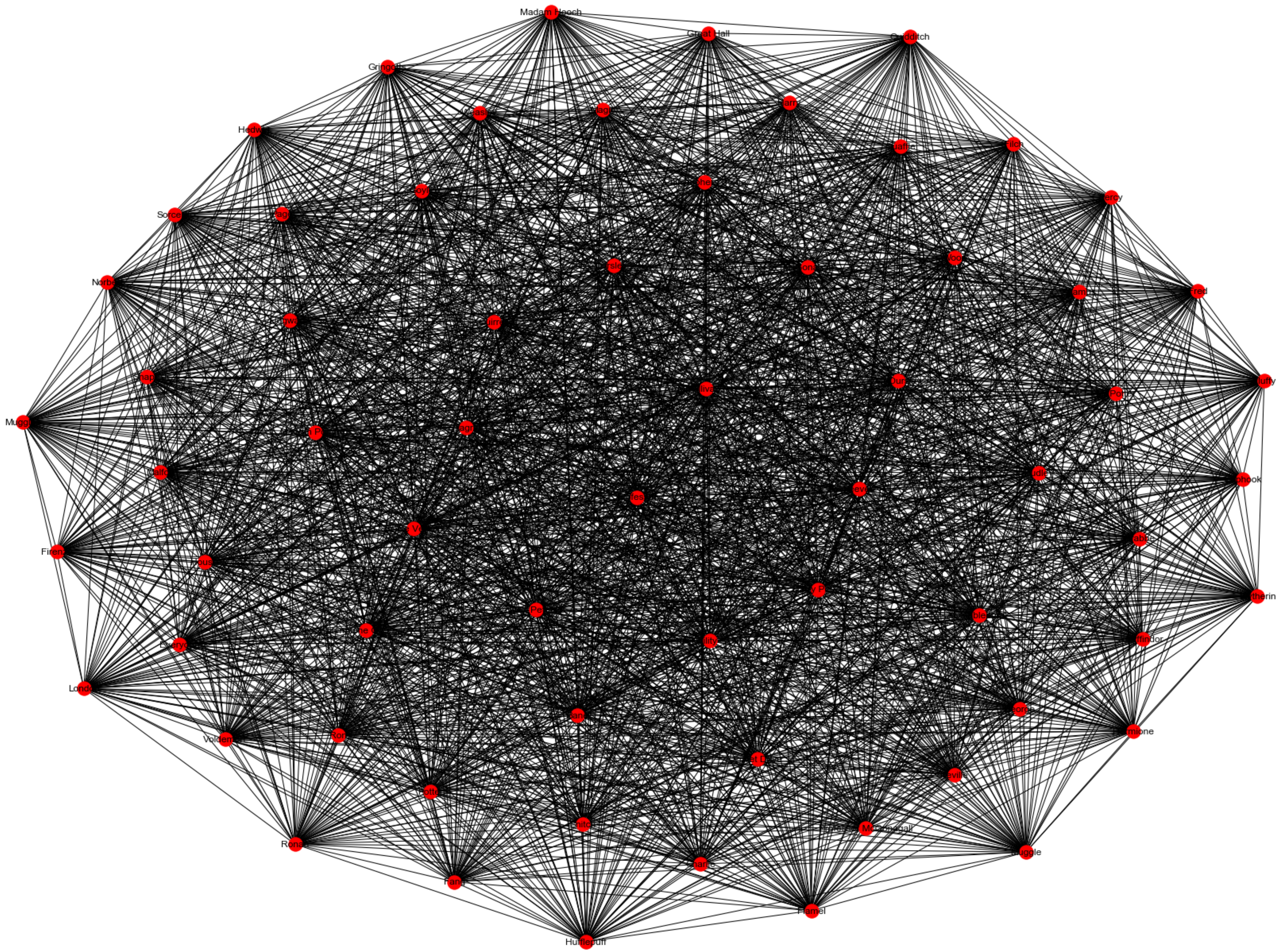
In [365]:
```
G = nx.from_pandas_edgelist(interaction_df,
                            source='source',
                            target='target')

nx.draw(G, with_labels=True)
```


Marley
Snape
Philosophers Stone
Rage
Ron
Hagrid
Harry
Dudley
Hermione
Harry Potter
Neville
Dumbledore

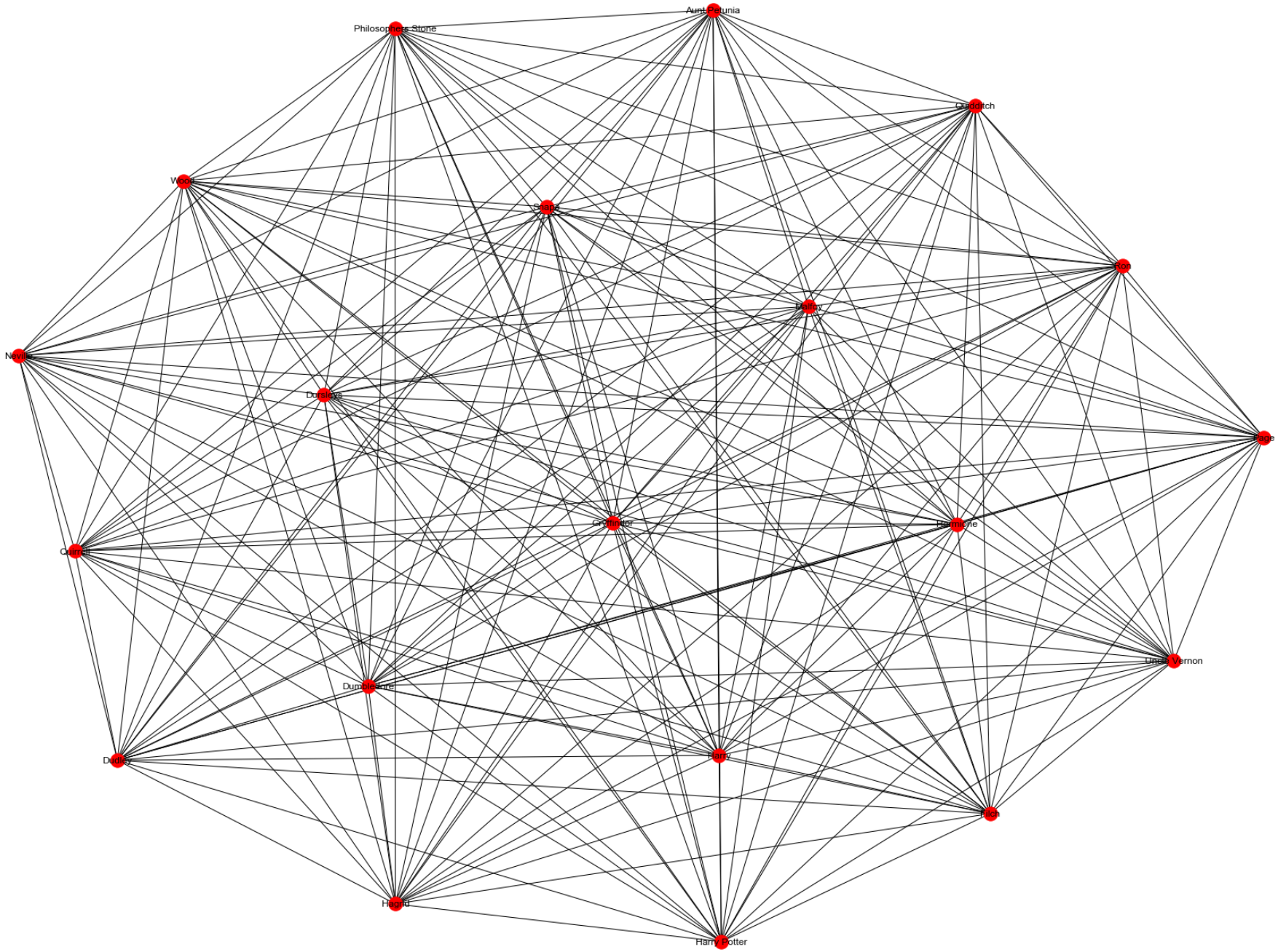
In [333]:
```
G = nx.from_pandas_edgelist(interaction_df,
                            source='source',
                            target='target')

nx.draw(G, with_labels=True)
```


Madam Hooch
Great Hall
Hufflepuff
Flamel

In [356]:
```
G = nx.from_pandas_edgelist(interaction_df,
                            source='source',
                            target='target')

nx.draw(G, with_labels=True)
```



In [335]:
```
import pandas as pd
```

In [336]:
```
pd.Series(nx.pagerank(G)).sort_values(ascending=False)[:10]
```

Out[336]:
```
Gringotts              0.015385
Crabbe                 0.015385
Sorcerer               0.015385
Professor McGonagall   0.015385
Hogwarts               0.015385
Great Hall             0.015385
Page                   0.015385
Slytherin              0.015385
Gryffindor             0.015385
Muggle                 0.015385
dtype: float64
```

In [366]:
```
pd.Series(nx.pagerank(G)).sort_values(ascending=False)[:10]
```

Out[366]:
```
Hermione       0.083333
Dudley         0.083333
Harry          0.083333
Snape          0.083333
Malfoy         0.083333
Ron            0.083333
Hagrid         0.083333
Neville        0.083333
Page           0.083333
Harry Potter   0.083333
dtype: float64
```

In [ ]:
```
    '''
    Automatically extracts lists of plausible character names from a book

    Parameters
    ----------
    book : string (required)
        book in string form (with original upper/lowercasing intact)

    Returns
    -------
    characters : list
        list of plasible character names
    '''
    nlp = spacy.load('en')
    stopwords = nltk.corpus.stopwords.words('english')

    words = [remove_punctuation(w) for w in book.split()]
    unique_words = list(set(words))

    characters = [word.text for word in nlp(' '.join(unique_words)) if word.pos_ == 'PROPN']
    characters = [c for c in characters if len(c) > 2]
    characters = [c for c in characters if c.istitle()]
    characters = [c for c in characters if not (c[-1] == 's' and c[:-1] in characters)]
    characters = list(set([c.title() for c in [c.lower() for c in characters]]) - set(stopwords))

    return [tuple([c + ' ']) for c in set(characters)]
```

In [ ]:
```
book = load_book('data/raw/fellowship_of_the_ring.txt', lower=False)
```

In [ ]:
```
extract_character_names(book)[-10:]
```

In [ ]:
```
%%timeit
extract_character_names(book)
```

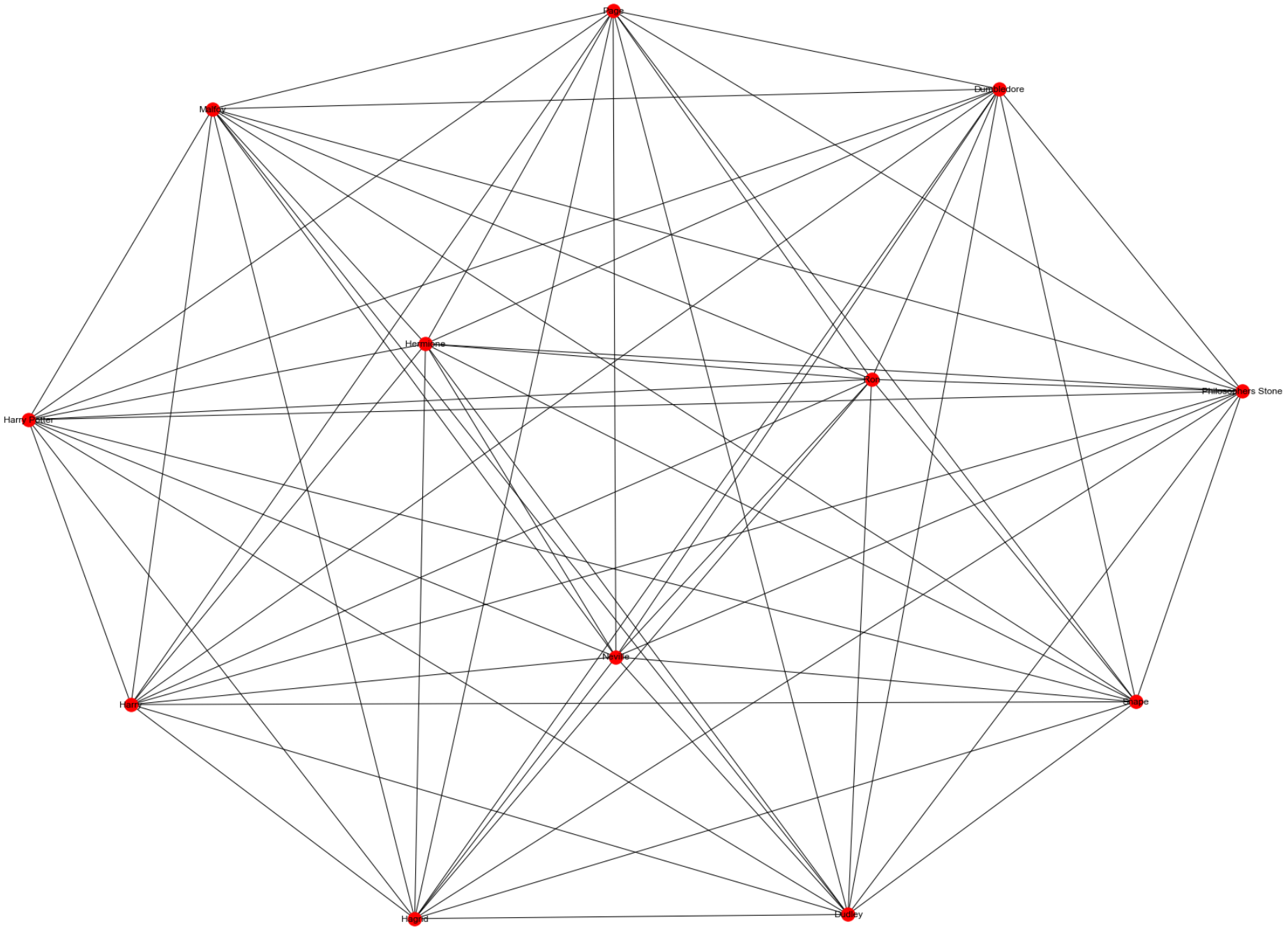
In [367]:
```
import networkx as nx
interaction_df = get_interaction_df(cooccurence, threshold=2)
interaction_df.sample(5)
```

Out[367]:

	source	target	value
55	Ron	Hermione	3016073
25	Harry Potter	Malfoy	8634198
33	Page	Malfoy	2165126
57	Malfoy	Harry	2296048
5	Dumbledore	Ron	3013139

In [369]:
```
G = nx.from_pandas_edgelist(interaction_df,
                            source='source',
                            target='target')
```

In [370]:
```
nx.draw_spring(G, with_labels=True)
```



In [371]:
```
pd.Series(nx.pagerank(G)).sort_values(ascending=False)[:5]
```

Out[371]:
```
Hermione    0.083333
Dudley      0.083333
Harry       0.083333
Snape       0.083333
Malfoy      0.083333
dtype: float64
```

In [372]:
```
a, b = nx.hits(G)
pd.Series(a).sort_values(ascending=False)[:5]
```

Out[372]:
```
Hermione    0.083333
Dudley      0.083333
Harry       0.083333
Snape       0.083333
Malfoy      0.083333
dtype: float64
```

In [373]:
```
list(nx.enumerate_all_cliques(G))[-1]
```

Out[373]:
```
['Dumbledore',
 'Philosophers Stone',
 'Harry Potter',
 'Page',
 'Neville',
 'Hagrid',
 'Ron',
 'Malfoy',
 'Snape',
 'Harry',
 'Dudley',
 'Hermione']
```

In [379]:
```
comms = nx.communicability(G)
print(comms["Malfoy"]["Dudley"])
print(comms["Ron"]["Hermione"])
```

```
4989.481152979712
4989.481152979712
```

In [1]:
```
import nltk
nltk.download('vader_lexicon')
```

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

Out[1]:
```
True
```

In [3]:
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
```

```
C:\Users\Dell\Anaconda3\lib\site-packages\nltk\twitter\__init__.py:20: UserWarning: The twython library has not been installed.
Some functionality from the twitter package will not be available.
  warnings.warn("The twython library has not been installed. "
```

In [ ]:
```
!pip uninstall spacy
```

In [ ]:
```
!pip install spacy
```

In [15]:
```
from bookworm import *

%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (12,9)

import pandas as pd
import numpy as np
import networkx as nx
```

In [2]:
```
!pip install bookworm
```

```
Collecting bookworm
  Downloading https://files.pythonhosted.org/packages/fb/49/f320f2204929d40107a11d47739604df4f333a2ca419d9d6ade12b983045/bookworm-0.2.1.tar.gz (https://files.pythonhosted.org/packages/fb/49/f320f2204929d40107a11d47739604df4f333a2ca419d9d6ade12b983045/bookworm-0.2.1.tar.gz)
Requirement already satisfied: requests in c:\users\dell\anaconda3\lib\site-packages (from bookworm) (2.19.1)
Requirement already satisfied: urllib3<1.24,>=1.21.1 in c:\users\dell\anaconda3\lib\site-packages (from requests->bookworm) (1.23)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\anaconda3\lib\site-packages (from requests->bookworm) (2019.3.9)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\dell\anaconda3\lib\site-packages (from requests->bookworm) (3.0.4)
Requirement already satisfied: idna<2.8,>=2.5 in c:\users\dell\anaconda3\lib\site-packages (from requests->bookworm) (2.7)
Building wheels for collected packages: bookworm
  Building wheel for bookworm (setup.py): started
  Building wheel for bookworm (setup.py): finished with status 'done'
  Stored in directory: C:\Users\Dell\AppData\Local\pip\Cache\wheels\ff\b2\9f\3dfdbb3b0161b1d03c7911a1ea1f14ded242fc1a1f9595c507
Successfully built bookworm
Installing collected packages: bookworm
Successfully installed bookworm-0.2.1

WARNING: You are using pip version 19.1.1, however version 20.0.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

In [4]:
```
import community
```

In [5]:
```
pwd
```

Out[5]:
```
'C:\\Users\\Dell'
```

In [11]:
```
import pandas as pd
```

In [17]:
```
pwd
```

Out[17]:
```
'C:\\Users\\Dell\\downloads\\character-extraction-master\\character-extraction-master'
```

In [ ]:
```

```