HW1

September 26, 2017

Pan Chen
IST-664
HW1
Due 9/26/2017
Prof. Nancy McCracken

Choosing the data: For this assignment, I would like to analyze and compare the first books of two contemporary literature masterpiece Series: "Twilight" and "Fifty Shades of Gray". The two highly successful franchise are actually related, as the Fifty Shades trilogy was developed from a Twilight fan fiction series originally titled Master of the Universe and published episodically on fan-fiction websites under the pen name "Snowqueen’s Icedragon". I want to see **what do the themes and/or subject matter of Fifty Shades and Twilight have in common and what do they differ?**

Due to the very high popularities of these two books, it is easy to find them on txt format on the internet. Here I just downloaded the txt files of both books with some Google searches.

There are some limitations to my data collecting process though: the most significant one is, these documents were collected came from unreliable sources without paying(internet search engine), instead of official sources like Amazon or Kindle, so I was pretty sure I violated some copyright law.

These documents from unoffical/unreliable sources might contain typos, imcompleteness, altering in the txt file, which might generate unreliable results. Even if there were something like typos in the document, it is hard for me to detect as I do not know if the author incorrectly misspelled the word or not. Although some of my family members did own these two books, I found it impractical to type every single word of the book into txt file for ths assignment, so I chose to get the txt files for free on the internet.

Also, when I tried to import Twilight.txt I got from the internet, the console said "’utf-8’ codec can’t decode byte 0xa1 in position 1116: invalid start byte". So I guess the txt file is not in utf-8 format. I resolved this by converting the txt file to utf-8 format with the text editior of my operating system.

Here I will demonstrate the process of import two documents in my Python kernel in the cell below.

```
In [17]: #Import txt files
         #1. Import 50 Shades
         f = open('Fifty Shades Of Grey.txt')
         fiftytext = f.read()
         f.close()
```

```
#2. Import Twilight
f = open('Twilight.txt')
twilighttext = f.read()
f.close()

#3. Validate if they are imported successfully
print(twilighttext[0:100])
print(fiftytext[0:100])
#Hooray, they are successfully imported
```

```
TWILIGHT
By:Stephenie Meyer
==========================================================
Contents

50 Shades of Grey

I scowl with frustration at myself in the mirror. Damn my hair  it just won't be
```

Examine the text in the documents that you chose and decide how to process the words a) Briefly state why you chose the processing options that you did

1) Firstly, I tokenized all th words from the document
2) Then I made all these tokens in lower cases..because I don't want "Knowing" at the beginning of a sentence and "knowing" in the middle of a sentence be treated like different words.
3) I made an alpha filter to filter out all the tokens that contains punctuation mark(s) only, because I was analyzing the subjects and themes of the two books, not the punctuation marks.
4) I also filtered out the stopwords tokens by using the default stopwords filter from 'nltk' package, because these stop words did not contribute to the analysis of subject, as they are usually contraction words that contained little substance unlike nouns

b) Are there any problems with the word or bigram lists that you found? Could you get a better list of bigrams? How are the top 50 bigrams by frequency different from the top 50 bigrams scored by Mutual Information?

1) There is a problem with lower-casing all of the tokens: "Grey" as in the name "Christian Grey" was treated the same as "grey" as in the color grey, and "Christian" as in the name "Chirstian Grey" was treated as the same as "christian" as someone who believes in christianity.
2) My initial word frequency list contained frequent tokens like "'ll","'d","'s", which would contribute little to my analysis, so I decide to add them to the stopword list and filter out these tokens. And then run the word frequency list again as well as bigrams with my new filtered tokens.

c) If you modify the stop word list, or expand the methods of filtering, describe that here.

- As I mentioned in the question above, there were some tokens that were distracting for the analysis. They were "’s","n’t","’m","’ll","’d","’ll","’re","’ve","could","would", so I filtered them out.

d)You may choose to also run top trigram lists, and include them in the analysis in part 3

- I don’t really think it would be necessary for the analysis of this assignment, but I still ran one just for fun....because these two books are very entertaining.

```
In [18]: #Processing of the two documents
         #0.import the necessary packages
         import nltk
         import re

         #For fifty shades of grey
         #1. Tokenize fiftytext
         fiftytokens = nltk.word_tokenize(fiftytext)
         twilighttokens = nltk.word_tokenize(twilighttext)

         #2. Use all lower cases
         fiftywords = [w.lower( ) for w in fiftytokens]

         #3. eliminate all the punctuations from those lowercased-tokens cuz they annoying
         def alpha_filter(w):
           # pattern to match word of non-alphabetical characters
             pattern = re.compile('^[^a-z]+$')
             if (pattern.match(w)):
                 return True
             else:
                 return False
         alphafiftywords = [w for w in fiftywords if not alpha_filter(w)]

         #print(alphafiftywords[:100])

         #4. eliminate the stopwords
         stopwords = nltk.corpus.stopwords.words('english')
         stoppedfiftywords = [w for w in alphafiftywords if not w in stopwords]


         #Run these steps again for the processiong of Twilight
         #1. Tokenize twilight text
         twilighttokens = nltk.word_tokenize(twilighttext)

         #2. Use all lower cases
         twilightwords = [w.lower( ) for w in twilighttokens]

         #3. eliminate all the punctuations from those lowercased-tokens cuz they annoying
```

```
        alphatwilightwords = [w for w in twilightwords if not alpha_filter(w)]

        #print(alphafiftywords[:100])

        #4. eliminate the stopwords
        stoppedtwilightwords = [w for w in alphatwilightwords if not w in stopwords]


        #I think I've done processing of both of them? Let me check it

        print('stoppedfiftywords:', stoppedfiftywords[0:100],"\n\nstoppedtwilightwords:", sto

        #Looking good!
```

```
stoppedfiftywords: ['shades', 'grey', 'scowl', 'frustration', 'mirror', 'damn', 'hair', 'wo', '

stoppedtwilightwords: ['twilight', 'stephenie', 'meyer', 'contents', 'preface', 'first', 'sigh
```

```
In [19]: #1. list the top 50 words by frequency (normalized by the length of the document)
         #For fifty shades
         fiftydist = nltk.FreqDist(stoppedfiftywords)
         fiftyitems = fiftydist.most_common(50)
         print ("top 50 words by normalized frequency of Fifty Shades:\n")
         for item in fiftyitems:
             print (item[0], '\t',item[1]/len(alphafiftywords))

         #For Twilight
         twilightdist = nltk.FreqDist(stoppedtwilightwords)
         twilightitems = twilightdist.most_common(50)
         print ("\ntop 50 words by normalized frequency of Fifty Shades:\n")
         for item in twilightitems:
             print (item[0], '\t',item[1]/len(alphatwilightwords))
```

```
top 50 words by normalized frequency of Fifty Shades:

's          0.013553417170683693
n't          0.007563493194151244
christian          0.005523681245386139
'm          0.005038011733775401
eyes          0.0034579669226684627
like          0.003185991996166449
want          0.003030577752451012
grey          0.002959346224081437
back          0.002609664175721705
know          0.002577286208280989
anastasia          0.0025190058668877005
oh          0.002512530273399557
```

```
kate          0.0022276041599212566
're          0.0021175190706228227
head          0.0021175190706228227
'll          0.002065714322717677
think          0.002039811948765104
one          0.0019815316073718155
've          0.0018390685506326655
steele          0.0018325929571445224
hand          0.0017743126157512335
feel          0.0016253739655239404
going          0.0015929959980832244
says          0.001586520404595081
hands          0.0015541424371543651
see          0.0015347156566899356
time          0.0014958620957610765
miss          0.0014958620957610765
ca          0.001405203786927072
hair          0.0013469234455337831
ana          0.00134044785204564
voice          0.00134044785204564
would          0.00134044785204564
take          0.001308069884604924
look          0.0012951186976286376
good          0.0012951186976286376
yes          0.0012951186976286376
mouth          0.0012821675106523512
go          0.001275691917164208
need          0.001269216323676065
well          0.0012627407301879217
way          0.001236838356235349
face          0.001165606827865774
get          0.0011397044539132011
still          0.0011267532669369147
'd          0.0011138020799606283
smile          0.0011138020799606283
around          0.0010749485190317692
bed          0.0010360949581029102
may          0.0010360949581029102

top 50 words by normalized frequency of Fifty Shades:

n't          0.01180594435383754
's          0.007781375007099795
could          0.005111852194445121
eyes          0.004024569346737746
edward          0.0035458403317024092
would          0.0034646998206794705
said          0.0031320237254854233
```

```
'd          0.0030589972655647784
back          0.0030021989078487217
like          0.0029535146012349585
voice          0.0027425492725753186
asked          0.0026777636863756968
face          0.0025153558417110091
'm          0.0024342153306881526
one          0.0023368467174606266
see          0.0022881624108468633
looked          0.0021583375932101662
still          0.0020853111332895173
know          0.0020285127755734606
away          0.001898687957936759
're          0.0018337755491184082
alice          0.0017607490891977638
'll          0.00175263503809547
charlie          0.0017120647825840008
time          0.0016958366803794413
bella          0.0016877226292771191
going          0.0016877226292771191
around          0.0016877226292771191
think          0.0016877226292771191
get          0.0015741259138450053
head          0.001444301096208304
hand          0.001444301096208304
way          0.001395616789594541
go          0.001395616789594541
much          0.0013793886873899532
door          0.0013469324829807779
room          0.0013225903296738963
us          0.0013225903296738963
look          0.0012982481763670149
something          0.001282020074162427
right          0.0012333357675486664
well          0.0012252217164463702
thought          0.0012089936142417824
never          0.0011521952565257255
sure          0.0011521952565257255
want          0.0011440812054234317
seemed          0.0011359671543211138
even          0.0011278531032188441
though          0.0011116250010142564
first          0.001087282847707375
```

I don't like some of the words in the word frequency list because 1) some of them do not seem to contribute to the analysis of the books' subjects and styles (could, would), and some of them look like they are not cleaned up thoroughly('s,n't,'ll), so I manually added them to the stopword list and run the word frequency again

```
In [20]: #Manually add some stopwords to the stopwords list
         stopwords.extend(("'s","n't","'m","'ll","'d","'ll","'re","'ve","could","would"))

         #4. eliminate the stopwords
         stoppedfiftywords = [w for w in alphafiftywords if not w in stopwords]
         stoppedtwilightwords = [w for w in alphatwilightwords if not w in stopwords]
```

```
In [21]: #Run the word frequency a second time
         #1. list the top 50 words by frequency (normalized by the length of the document)
         #For fifty shades
         fiftydist = nltk.FreqDist(stoppedfiftywords)
         fiftyitems = fiftydist.most_common(50)
         print ("New top 50 words by normalized frequency of Fifty Shades:\n")
         for item in fiftyitems:
             print (item[0], '\t',item[1]/len(alphafiftywords))

         #For Twilight
         twilightdist = nltk.FreqDist(stoppedtwilightwords)
         twilightitems = twilightdist.most_common(50)
         print ("\nNew top 50 words by normalized frequency of Fifty Shades:\n")
         for item in twilightitems:
             print (item[0], '\t',item[1]/len(alphatwilightwords))
```

```
New top 50 words by normalized frequency of Fifty Shades:

christian          0.005523681245386139
eyes          0.0034579669226684627
like          0.003185991996166449
want          0.003030577752451012
grey          0.002959346224081437
back          0.002609664175721705
know          0.002577286208280989
anastasia          0.0025190058668877005
oh          0.002512530273399557
kate          0.0022276041599212566
head          0.0021175190706228227
think          0.002039811948765104
one          0.0019815316073718155
steele          0.0018325929571445224
hand          0.0017743126157512335
feel          0.0016253739655239404
going          0.0015929959980832244
says          0.001586520404595081
hands          0.0015541424371543651
see          0.0015347156566899356
time          0.0014958620957610765
```

```
miss          0.0014958620957610765
ca          0.001405203786927072
hair          0.0013469234455337831
ana          0.00134044785204564
voice          0.00134044785204564
take          0.001308069884604924
look          0.0012951186976286376
good          0.0012951186976286376
yes          0.0012951186976286376
mouth          0.0012821675106523512
go          0.001275691917164208
need          0.001269216323676065
well          0.0012627407301879217
way          0.001236838356235349
face          0.001165606827865774
get          0.0011397044539132011
still          0.0011267532669369147
smile          0.0011138020799606283
around          0.0010749485190317692
bed          0.0010360949581029102
may          0.0010360949581029102
really          0.001029619364614767
holy          0.0010166681776384806
thought          0.0010101925841503373
room          0.000997241397174051
say          0.0009713390232214782
date          0.0009454366492689055
looks          0.0009195342753163327
asks          0.0009065830883400464

New top 50 words by normalized frequency of Fifty Shades:

eyes          0.004024569346737746
edward          0.0035458403317024092
said          0.0031320237254854233
back          0.0030021989078487217
like          0.0029535146012349585
voice          0.0027425492725753186
asked          0.002677636863756968
face          0.002515355841711091
one          0.0023368467174606266
see          0.0022881624108468633
looked          0.002158337593210162
still          0.0020853111332895173
know          0.0020285127755734606
away          0.001898687957936759
alice          0.0017607490891977638
charlie          0.0017120647825840008
```

```
time          0.001695836680379413
bella          0.0016877226292771191
going          0.0016877226292771191
around          0.0016877226292771191
think          0.0016877226292771191
get          0.0015741259138450053
head          0.001444301096208304
hand          0.001444301096208304
way          0.001395616789594541
go          0.001395616789594541
much          0.0013793886873899532
door          0.0013469324829807779
room          0.0013225903296738963
us          0.0013225903296738963
look          0.0012982481763670149
something          0.001282020074162427
right          0.001233335767548664
well          0.0012252217164463702
thought          0.0012089936142417824
never          0.0011521952565257255
sure          0.0011521952565257255
want          0.0011440812054234317
seemed          0.001135967154321138
even          0.0011278531032188441
though          0.0011116250010142564
first          0.001087282847707375
tried          0.001087282847707375
carlisle          0.0010710547455027872
really          0.0010629406944004933
mike          0.0010629406944004933
smiled          0.0010467125921959056
turned          0.0010467125921959056
made          0.0010385985410936117
long          0.001030484489991318
```

```
In [22]: #2. list the top 50 bigrams by frequencies
         #For fifty shades
         # setup for bigrams and bigram measures
         from nltk.collocations import *
         bigram_measures = nltk.collocations.BigramAssocMeasures()
         # create the bigram finder and score the bigrams by frequency
         finder = BigramCollocationFinder.from_words(stoppedfiftywords)
         scored = finder.score_ngrams(bigram_measures.raw_freq)
         #scores are bigram frequencies normalized by dividing by the total number of bigrams
         #scores are sorted in decreasing frequency
         print ("Top 50 bigrams by frequencies of Fifty Shades:")
         for bscore in scored[:50]:
```

```
    print (bscore)
print("\n")

#For Twilight
finder = BigramCollocationFinder.from_words(stoppedtwilightwords)
scored = finder.score_ngrams(bigram_measures.raw_freq)
#scores are bigram frequencies normalized by dividing by the total number of bigrams
#scores are sorted in decreasing frequency
print ("Top 50 bigrams by frequencies of Twilight:")
for bscore in scored[:50]:
    print (bscore)
print("\n")
```

```
Top 50 bigrams by frequencies of Fifty Shades:
(('christian', 'grey'), 0.002897505926716668)
(('anastasia', 'steele'), 0.001954776725679666)
(('miss', 'steele'), 0.0016359124370936213)
(('date', 'may'), 0.0013725028073921061)
(('enterprises', 'holdings'), 0.0009843201952003994)
(('grey', 'enterprises'), 0.0009843201952003994)
(('mr.', 'grey'), 0.0009704565304792669)
(('gray', 'eyes'), 0.0009427292010370022)
(('grey', 'subject'), 0.0009150018715947374)
(('christian', 'greyceo'), 0.0008872745421524726)
(('greyceo', 'grey'), 0.0008456835479890754)
(('steele', 'subject'), 0.0008179562185468106)
(('inner', 'goddess'), 0.0008040925538256783)
(('holdings', 'inc.'), 0.0007347742302200163)
(('may', 'anastasia'), 0.0007347742302200163)
(('miss', 'steele.'), 0.0005961375830086925)
(('holy', 'shit'), 0.0005545465888452954)
(('shakes', 'head'), 0.0004990919299607658)
(('may', 'christian'), 0.0004575009357973687)
(('shake', 'head'), 0.0004436372710762363)
(('first', 'time'), 0.0004297736063551039)
(('holy', 'crap'), 0.0004297736063551039)
(('one', 'side'), 0.0004297736063551039)
(('voice', 'soft'), 0.0004297736063551039)
(('roll', 'eyes'), 0.00041590994163397154)
(('want', 'know'), 0.00041590994163397154)
(('close', 'eyes'), 0.00040204627691283915)
(('inc.', 'anastasia'), 0.00040204627691283915)
(('closes', 'eyes'), 0.00038818261219170677)
(('date', 'june'), 0.0003743189474705744)
(('let', 'go'), 0.0003743189474705744)
(('est', 'christian'), 0.000360455282749442)
(('head', 'back'), 0.0003465916180283096)
(('head', 'one'), 0.0003465916180283096)
```

```
(('deep', 'breath'), 0.00033272795330717723)
(('feel', 'like'), 0.00033272795330717723)
(('dominant', 'shall'), 0.00031886428858604485)
(('holds', 'hand'), 0.00031886428858604485)
(('oh', 'no'), 0.00031886428858604485)
(('katherine', 'kavanagh'), 0.00030500062386491246)
(('last', 'night'), 0.00030500062386491246)
(('mrs.', 'robinson'), 0.00030500062386491246)
(('raises', 'eyebrows'), 0.00030500062386491246)
(('mr', 'grey'), 0.0002911369591437801)
(('want', 'go'), 0.0002911369591437801)
(('oh', 'my'), 0.0002772732944226477)
(('cocks', 'head'), 0.0002634096297015153)
(('hand', 'leads'), 0.0002495459649803829)
(('june', 'est'), 0.0002495459649803829)
(('make', 'way'), 0.0002495459649803829)


Top 50 bigrams by frequencies of Twilight:
(('shook', 'head'), 0.0007518124049046809)
(('looked', 'away'), 0.0005728094513559474)
(('edward', 'cullen'), 0.0005370088606462006)
(('let', 'go'), 0.00046540767922670725)
(('edward', 'said'), 0.0004475073838718339)
(('looked', 'like'), 0.0004296070885169605)
(('mr', 'banner'), 0.0003938064978072138)
(('one', 'hand'), 0.0003580059070974671)
(('first', 'time'), 0.00034010561174259375)
(('parking', 'lot'), 0.00034010561174259375)
(('dr.', 'cullen'), 0.00030430502103284706)
(('port', 'angeles'), 0.00030430502103284706)
(('sounded', 'like'), 0.00030430502103284706)
(('around', 'corner'), 0.0002864047256779737)
(('closed', 'eyes'), 0.0002864047256779737)
(('alice', 'jasper'), 0.0002685044303231003)
(('rolled', 'eyes'), 0.0002685044303231003)
(('looking', 'away'), 0.000250604134968227)
(('walked', 'away'), 0.000250604134968227)
(('want', 'know'), 0.000250604134968227)
(('eyes', 'narrowed'), 0.00023270383961335363)
(('kept', 'eyes'), 0.00023270383961335363)
(('never', 'seen'), 0.00023270383961335363)
(('see', 'face'), 0.00023270383961335363)
(('eyes', 'still'), 0.00021480354425848025)
(('last', 'night'), 0.00021480354425848025)
(('long', 'time'), 0.00021480354425848025)
(('make', 'sure'), 0.00021480354425848025)
(('opened', 'eyes'), 0.00021480354425848025)
```

```
(('voice', 'sounded'), 0.00021480354425848025)
(('alice', 'said'), 0.0001969032489036069)
(('anything', 'else'), 0.0001969032489036069)
(('bella', 'said'), 0.0001969032489036069)
(('door', 'open'), 0.0001969032489036069)
(('even', 'though'), 0.0001969032489036069)
(('seemed', 'like'), 0.0001969032489036069)
(('asked', 'voice'), 0.00017900295354873356)
(('behind', 'us'), 0.00017900295354873356)
(('bit', 'lip'), 0.00017900295354873356)
(('eyes', 'closed'), 0.00017900295354873356)
(('far', 'away'), 0.00017900295354873356)
(('first', 'day'), 0.00017900295354873356)
(('going', 'tell'), 0.00017900295354873356)
(('looked', 'see'), 0.00017900295354873356)
(('opened', 'door'), 0.00017900295354873356)
(('said', 'voice'), 0.00017900295354873356)
(('say', 'anything'), 0.00017900295354873356)
(('voice', 'still'), 0.00017900295354873356)
(('across', 'face'), 0.0001611026581938602)
(('alice', 'asked'), 0.0001611026581938602)
```

```
In [23]: #3. list the top 50 bigrams by their Mutual Information scores (using min frequency 5
         #For fifty shades
         finder2 = BigramCollocationFinder.from_words(stoppedfiftywords)
         finder2.apply_freq_filter(5)
         scored = finder2.score_ngrams(bigram_measures.pmi)
         print ("Top 50 bigrams by mutual information scores of Fifty Shades:")
         for bscore in scored[:50]:
             print (bscore)
         print('\n')


         #For Twilight
         finder2 = BigramCollocationFinder.from_words(stoppedtwilightwords)
         scored = finder2.score_ngrams(bigram_measures.pmi)
         print ("Top 50 bigrams by mutual information scores of Twilight:")
         for bscore in scored[:50]:
             print (bscore)
```

```
Top 50 bigrams by mutual information scores of Fifty Shades:
(('boxer', 'briefs'), 13.816403709653716)
(('steering', 'wheel'), 13.55336930381992)
(('anal', 'intercourse'), 12.845550055313232)
(('scalp', 'prickles'), 12.678900185903776)
```

```
(('orange', 'juice'), 12.526897092458727)
(('heaven', 'sake'), 12.24524700845759)
(('safety', 'procedures'), 12.231441208932559)
(('charlie', 'tango'), 12.138331804541076)
(('acts', 'involving'), 11.89040429109749)
(('fifty', 'shades'), 11.816403709653715)
(('class', 'lounge'), 11.746014381762318)
(('cable', 'ties'), 11.67890018590378)
(('brow', 'furrows'), 11.668011869761042)
(('foil', 'packet'), 11.500901883925783)
(('dr.', 'flynn'), 11.437892086399986)
(('finds', 'release'), 11.3727970581781)
(('dr.', 'greene'), 11.330976882483473)
(('squirm', 'uncomfortably'), 11.138331804541076)
(('index', 'finger'), 11.093937685182622)
(('personal', 'trainer'), 11.050868963290737)
(('parking', 'lot'), 10.968406803098762)
(('computer', 'loan'), 10.830903279348831)
(('shoes', 'socks'), 10.746014381762318)
(('nowhere', 'seen'), 10.602278904300867)
(('sexual', 'activity'), 10.602278904300867)
(('raises', 'eyebrows'), 10.569857426608486)
(('june', 'est'), 10.546705573538906)
(('shrug', 'apologetically'), 10.535447395822658)
(('riding', 'crop'), 10.47726632473413)
(('without', 'hesitation'), 10.475366791818644)
(('ear', 'buds'), 10.437892086399986)
(('bar', 'stools'), 10.290334897986128)
(('private', 'joke'), 10.290334897986128)
(('laters', 'baby'), 10.231441208932559)
(('sweat', 'pants'), 10.231441208932559)
(('english', 'breakfast'), 10.231441208932557)
(('husband', 'number'), 10.219468567266482)
(('mrs.', 'jones'), 10.217766272035481)
(('katherine', 'kavanagh'), 10.18837248704067)
(('control', 'freak'), 10.166346180710674)
(('fully', 'aware'), 10.16105188104116)
(('inner', 'goddess'), 10.15947343991993)
(('linen', 'shirt'), 10.121523516854522)
(('breathing', 'ragged'), 10.103685661734183)
(('chest', 'drawers'), 10.09393768518262)
(('allotted', 'times'), 10.083049369039886)
(('mrs.', 'robinson'), 10.062488046557567)
(('enterprises', 'holdings'), 9.988584685036395)
(('holdings', 'inc.'), 9.988584685036393)
(('holdings', 'inc'), 9.988584685036392)
```

```
Top 50 bigrams by mutual information scores of Twilight:
(('1st', 'ed'), 15.769657082282718)
(('40x', 'objective'), 15.769657082282718)
(('abstinence', 'resented'), 15.769657082282718)
(('acquaintances', 'considerately'), 15.769657082282718)
(('acquaintancesř', 'homesick'), 15.769657082282718)
(('actors', 'portray'), 15.769657082282718)
(('addicted', 'illegal'), 15.769657082282718)
(('adjoining', 'handkerchief-sized'), 15.769657082282718)
(('admired', 'civility'), 15.769657082282718)
(('advance', 'slinking'), 15.769657082282718)
(('advanced', 'placement'), 15.769657082282718)
(('affidavits', 'well-known'), 15.769657082282718)
(('afterward', 'ill'), 15.769657082282718)
(('agent', 'jodi'), 15.769657082282718)
(('agonizing', 'outcome'), 15.769657082282718)
(('alphabetized', 'listing'), 15.769657082282718)
(('american', 'pastime'), 15.769657082282718)
(('anemones', 'undulated'), 15.769657082282718)
(('angelfish', 'shark'), 15.769657082282718)
(('angerř', 'painř'), 15.769657082282718)
(('appallingly', 'luscious'), 15.769657082282718)
(('appatently', 'notion'), 15.769657082282718)
(('aquarium', 'bouquets'), 15.769657082282718)
(('areas', 'overpopulation'), 15.769657082282718)
(('areř', 'nice-looking'), 15.769657082282718)
(('aro', 'marcus'), 15.769657082282718)
(('array', 'sodas'), 15.769657082282718)
(('askance', 'luggage-less'), 15.769657082282718)
(('asset', 'community'), 15.769657082282718)
(('austen', 'selected'), 15.769657082282718)
(('austere', 'soaring'), 15.769657082282718)
(('avenue', 'americas'), 15.769657082282718)
(('aversion', "'his"), 15.769657082282718)
(('awakened', 'renewed'), 15.769657082282718)
(('awards', 'cluttering'), 15.769657082282718)
(('babysitters', 'handled'), 15.769657082282718)
(('bacon', 'requested'), 15.769657082282718)
(('balloon', 'arches'), 15.769657082282718)
(('battery', 'cables'), 15.769657082282718)
(('battle', 'blizzard'), 15.769657082282718)
(('bell-like', 'echoes'), 15.769657082282718)
(('billed', 'semiformal'), 15.769657082282718)
(('binding', 'motif'), 15.769657082282718)
(('biohazard', 'suit'), 15.769657082282718)
(('bleached', 'bone'), 15.769657082282718)
(('blend', 'succeed'), 15.769657082282718)
(('blondř', 'handsomer'), 15.769657082282718)
```

```
(('bond', 'convenience'), 15.769657082282718)
(('boundless', 'labyrinth'), 15.769657082282718)
(('bowls', 'valleys'), 15.769657082282718)
```

```
In [24]: #Run trigrams just for fun
         trigram_measures = nltk.collocations.TrigramAssocMeasures()
         finder3 = TrigramCollocationFinder.from_words(stoppedfiftywords)
         scored = finder3.score_ngrams(trigram_measures.raw_freq)
         print ("Top 50 trigrams by frequencies of Fifty Shades:")
         for bscore in scored[:50]:
             print (bscore)
         print('\n')

         finder3 = TrigramCollocationFinder.from_words(stoppedtwilightwords)
         scored = finder3.score_ngrams(trigram_measures.raw_freq)
         print ("Top 50 trigrams by frequencies of Twilight:")
         for bscore in scored[:50]:
             print (bscore)
         print('\n')
```

```
Top 50 trigrams by frequencies of Fifty Shades:
(('grey', 'enterprises', 'holdings'), 0.0009843201952003994)
(('christian', 'grey', 'subject'), 0.0009150018715947374)
(('christian', 'greyceo', 'grey'), 0.0008456835479890754)
(('greyceo', 'grey', 'enterprises'), 0.0008456835479890754)
(('anastasia', 'steele', 'subject'), 0.0008179562185468106)
(('enterprises', 'holdings', 'inc.'), 0.0007347742302200163)
(('date', 'may', 'anastasia'), 0.000720910565498884)
(('may', 'anastasia', 'steele'), 0.000720910565498884)
(('date', 'may', 'christian'), 0.0004436372710762363)
(('may', 'christian', 'grey'), 0.0004297736063551039)
(('holdings', 'inc.', 'anastasia'), 0.00040204627691283915)
(('inc.', 'anastasia', 'steele'), 0.00040204627691283915)
(('est', 'christian', 'grey'), 0.00033272795330717723)
(('head', 'one', 'side'), 0.00031886428858604485)
(('date', 'june', 'est'), 0.0002495459649803829)
(('anastasia', 'steele', 'dear'), 0.00023568230025925054)
(('cocks', 'head', 'one'), 0.00023568230025925054)
(('christian', 'grey', 'dear'), 0.00022181863553811815)
(('enterprises', 'holdings', 'inc'), 0.00022181863553811815)
(('mouth', 'drops', 'open'), 0.00020795497081698577)
(('steele', 'dear', 'miss'), 0.00020795497081698577)
(('miss', 'steele', 'says'), 0.00019409130609585339)
(('date', 'may', 'est'), 0.000180227641374721)
(('june', 'est', 'christian'), 0.000180227641374721)
(('may', 'est', 'christian'), 0.000180227641374721)
(('runs', 'hand', 'hair'), 0.00016636397665358862)
```

```
(('miss', 'steele', 'murmurs'), 0.00015250031193245623)
(('grey', 'dear', 'mr.'), 0.00013863664721132385)
(('date', 'june', 'anastasia'), 0.00012477298249019146)
(('dear', 'miss', 'steelei'), 0.00012477298249019146)
(('red', 'room', 'pain'), 0.00012477298249019146)
(('ana', 'christian', 'grey'), 0.00011090931776905908)
(('anastasia', 'steele', 'miss'), 0.00011090931776905908)
(('june', 'anastasia', 'steele'), 0.00011090931776905908)
(('mouth', 'presses', 'hard'), 0.00011090931776905908)
(('presses', 'hard', 'line'), 0.00011090931776905908)
(('takes', 'hand', 'leads'), 0.00011090931776905908)
(('taking', 'deep', 'breath'), 0.00011090931776905908)
(('white', 'linen', 'shirt'), 0.00011090931776905908)
(('long', 'index', 'finger'), 9.704565304792669e-05)
(('put', 'head', 'hands'), 9.704565304792669e-05)
(('takes', 'deep', 'breath'), 9.704565304792669e-05)
(('taking', 'hand', 'leads'), 9.704565304792669e-05)
(('gray', 'eyes', 'blaze'), 8.318198832679431e-05)
(('hair', 'behind', 'ear'), 8.318198832679431e-05)
(('hands', 'either', 'side'), 8.318198832679431e-05)
(('leans', 'forward', 'kisses'), 8.318198832679431e-05)
(('really', 'want', 'know'), 8.318198832679431e-05)
(('reply', 'anastasia', 'steele'), 8.318198832679431e-05)
(('submissive', 'personal', 'trainer'), 8.318198832679431e-05)


Top 50 trigrams by frequencies of Twilight:
(('took', 'deep', 'breath'), 0.00014320236283898684)
(('raised', 'one', 'eyebrow'), 0.00010740177212924013)
(('arm', 'around', 'waist'), 7.160118141949342e-05)
(('face', 'inches', 'mine'), 7.160118141949342e-05)
(('put', 'seat', 'belt'), 7.160118141949342e-05)
(('answer', 'yes', 'yes'), 5.3700886064620063e-05)
(('back', 'walked', 'away'), 5.3700886064620063e-05)
(('bella', 'edward', 'voice'), 5.3700886064620063e-05)
(('breathed', 'sigh', 'relief'), 5.3700886064620063e-05)
(('engine', 'roared', 'life'), 5.3700886064620063e-05)
(('going', 'seattle', 'day'), 5.3700886064620063e-05)
(('held', 'phone', 'ear'), 5.3700886064620063e-05)
(('kept', 'eyes', 'away'), 5.3700886064620063e-05)
(('last', 'time', 'seen'), 5.3700886064620063e-05)
(('left', 'jacket', 'car'), 5.3700886064620063e-05)
(('light', 'brown', 'hair'), 5.3700886064620063e-05)
(('like', 'everyone', 'else'), 5.3700886064620063e-05)
(('like', 'long', 'time'), 5.3700886064620063e-05)
(('lips', 'pressed', 'together'), 5.3700886064620063e-05)
(('mr', 'banner', 'asked'), 5.3700886064620063e-05)
(('pressed', 'lips', 'together'), 5.3700886064620063e-05)
```

```
(('shut', 'door', 'behind'), 5.3700886064620063e-05)
(('squeezed', 'eyes', 'shut'), 5.3700886064620063e-05)
(('took', 'face', 'hands'), 5.3700886064620063e-05)
(('took', 'step', 'back'), 5.3700886064620063e-05)
(('tried', 'keep', 'voice'), 5.3700886064620063e-05)
(('turned', 'walked', 'away'), 5.3700886064620063e-05)
(('able', 'know', 'thinking'), 3.580059070974671e-05)
(('alice', 'carlisle', 'asked'), 3.580059070974671e-05)
(('almost', 'ninety', 'years'), 3.580059070974671e-05)
(('another', 'bite', 'pizza'), 3.580059070974671e-05)
(('anything', 'monday', 'night'), 3.580059070974671e-05)
(('anything', 'wrong', 'bella'), 3.580059070974671e-05)
(('arms', 'across', 'chest'), 3.580059070974671e-05)
(('arms', 'tightly', 'across'), 3.580059070974671e-05)
(('asked', 'curiously', 'looked'), 3.580059070974671e-05)
(('asked', 'looked', 'see'), 3.580059070974671e-05)
(('asked', 'nothing', 'wrong'), 3.580059070974671e-05)
(('asked', 'one', 'answered'), 3.580059070974671e-05)
(('asked', 'raising', 'eyebrows'), 3.580059070974671e-05)
(('asked', 'turning', 'back'), 3.580059070974671e-05)
(('away', 'eyes', 'wandering'), 3.580059070974671e-05)
(('back', 'corner', 'truck'), 3.580059070974671e-05)
(('back', 'look', 'face'), 3.580059070974671e-05)
(('back', 'soon', 'promised'), 3.580059070974671e-05)
(('back', 'time', 'dance'), 3.580059070974671e-05)
(('believe', 'give', 'easily'), 3.580059070974671e-05)
(('bell', 'rang', 'last'), 3.580059070974671e-05)
(('bella', 'alice', 'said'), 3.580059070974671e-05)
(('bella', 'mike', 'said'), 3.580059070974671e-05)
```

Describe a problem or question that is based on the difference between the two documents Again, my question is: What do the themes of Fifty Shades and Twilight have in common and what do they differ?

From the bigram frequency distributions, I can see that both books might be narrated from a female’s perspective, as they both described the love interest of the female protagonist with quite a bit of words, which coincides with these books’ core audiences: female.

In "Top 50 bigrams by frequencies of Fifty Shades", bigrams (’christian’, ’grey’) is the no.1 on the list, as well as (’gray’, ’eyes’), (’grey’, ’subject’),(’grey’, ’subject’), which are all the instances of describing Christian Grey, the protagonist Anastasia Steele’s love interest. Not suprisingly, "christian"’s frequency is more than twice as high as "anastasia" on the word frequency list of Fifty Shades.

Similarly, in "Top 50 bigrams by frequencies of Twilight", (’edward’, ’cullen’) and (’edward’, ’said’) were no.3 and no.5 on the list, and "edward" appeared to be mentioned as twice as may as "bella" on the word frequency list. Again, this Edward Cullen guy was the love interest of the female protagonist Bella Swan.

There are also some similarities in terms of describing a person in two books, as the word "eyes" appeared to be second and first on both books' word frequency list, respectively. They might be used to describe Christian and Edward.

However, there is some differences in narrating style, as Fifty Shades seemed to be narrated from the first person point of view while Twilight was narrated from the third person POV. As I saw ('bella', 'said') in the Top 50 bigrams by frequencies of Twilight, but I couldn't seem to find ('anastasia', 'said') or something similar from Top 50 bigrams by frequencies of Fifty Shades.

One thing further reinforces the previous difference I made is, the characters in both books seem to be shaking their heads a lot, as ('shakes', 'head') and ('shook', 'head') made it on the both Top 50 bigram frequency lists respectively. However as you could see, Fifty shades uses the present tense "shakes" while Twilight uses past tense "shook", which might be because of the difference of POV in narrating of both books.

In addition, it looks like the subject matter of Fifty Shades is more mature oriented, with the appearances of bigrams like ('cocks', 'head'), ('dominant', 'shall') on the bigram raw frequency list, and ('boxer', 'briefs'),(('anal', 'intercourse') on the bigram MIS list, and someone likes to exclaim ('holy', 'shit') a lot in Fifty Shades, whereas similar mature subject matter and language were not to be found in the analysis of Twilight. This comes no surprise, as Anastasia and Christian are 20-something adults, while Bella and Edward were dating in the high school in the book.

This is just some preliminary analysis, but it has been a lot of fun running it, and I could not wait to apply more advanced NLP techniques to do some more in-depth analysis in the future.I am pretty sure the process of doing this homework is more fun than actually reading through these two books.