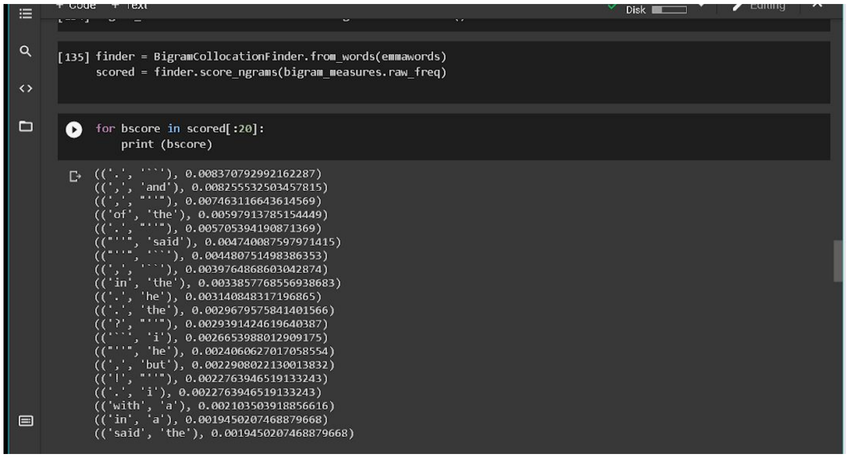
IST 664 NATURAL LANGUAGE PROCESSING
LAB 2
Book: chesterton-thursday.txt(The Man Who Was Thursday: A Nightmare by G. K. Chesterton)

Results :
Unfiltered list of words

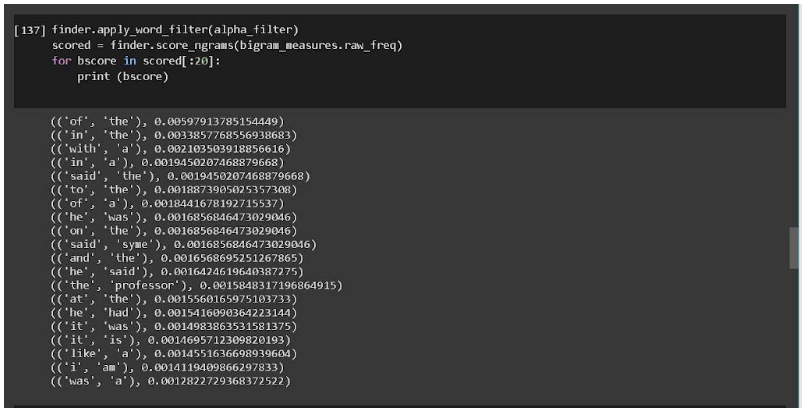


```
[135] finder = BigramCollocationFinder.from_words(emmawords)
      scored = finder.score_ngrams(bigram_measures.raw_freq)

      for bscore in scored[:20]:
          print (bscore)

((('.', '``'), 0.008370792992162287)
((',', 'and'), 0.008255532503457815)
((',', "''"), 0.007463116643614569)
(('of', 'the'), 0.00597913785154449)
(('.', "''"), 0.005705394190871369)
(("''", 'said'), 0.004740087597971415)
(("''", '``'), 0.004480751498386353)
((',', '``'), 0.0039764868603042874)
(('in', 'the'), 0.0033857768556938683)
(('.', 'he'), 0.003140848317196865)
(('.', 'the'), 0.0029679575841401566)
(('?', "''"), 0.0029391424619640387)
(('``', 'i'), 0.0026653988012909175)
(("''", 'he'), 0.0024060627017058554)
((',', 'but'), 0.0022908022130013832)
(('!', "''"), 0.0022763946519133243)
(('.', 'i'), 0.0022763946519133243)
(('with', 'a'), 0.0021035039188856616)
(('in', 'a'), 0.0019450207468879668)
(('said', 'the'), 0.0019450207468879668)
```

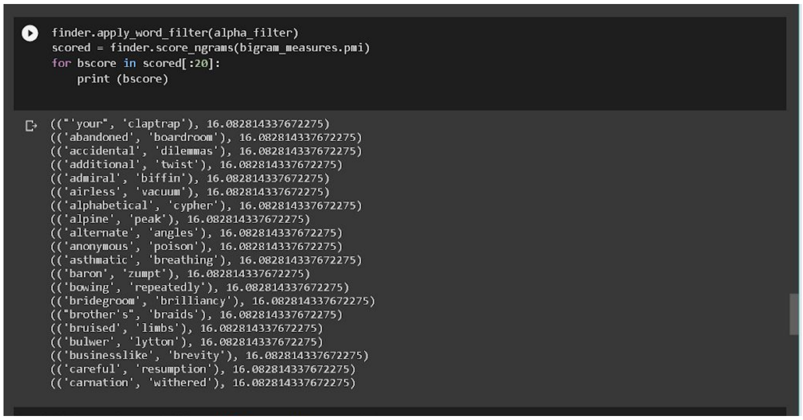
Non-alphabetic characters removed



```
[137] finder.apply_word_filter(alpha_filter)
      scored = finder.score_ngrams(bigram_measures.raw_freq)
      for bscore in scored[:20]:
          print (bscore)

      (('of', 'the'), 0.00597913785154449)
      (('in', 'the'), 0.0033857768556938683)
      (('with', 'a'), 0.0021035039188856616)
      (('in', 'a'), 0.0019450207468879668)
      (('said', 'the'), 0.0019450207468879668)
      (('to', 'the'), 0.0018873905025357308)
      (('of', 'a'), 0.0018441678192715537)
      (('he', 'was'), 0.0016856846473029046)
      (('on', 'the'), 0.0016856846473029046)
      (('said', 'syme'), 0.0016856846473029046)
      (('and', 'the'), 0.0016568695251267865)
      (('he', 'said'), 0.0016424619640387275)
      (('the', 'professor'), 0.0015848317196864915)
      (('at', 'the'), 0.0015560165975103733)
      (('he', 'had'), 0.0015416090364223144)
      (('it', 'was'), 0.0014983863531581375)
      (('it', 'is'), 0.0014695712309820193)
      (('like', 'a'), 0.0014551636698939604)
      (('i', 'am'), 0.0014119409866297833)
      (('was', 'a'), 0.0012822729368372522)
```

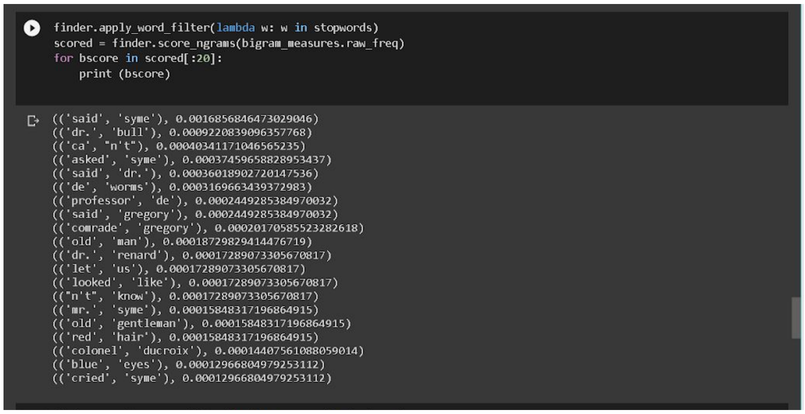
Non-alphabetic characters removed and the PMI scoring mechanism



```
finder.apply_word_filter(alpha_filter)
scored = finder.score_ngrams(bigram_measures.pmi)
for bscore in scored[:20]:
    print (bscore)
```

```
(("'your", 'claptrap'), 16.082814337672275)
(('abandoned', 'boardroom'), 16.082814337672275)
(('accidental', 'dilemmas'), 16.082814337672275)
(('additional', 'twist'), 16.082814337672275)
(('admiral', 'biffin'), 16.082814337672275)
(('airless', 'vacuum'), 16.082814337672275)
(('alphabetical', 'cypher'), 16.082814337672275)
(('alpine', 'peak'), 16.082814337672275)
(('alternate', 'angles'), 16.082814337672275)
(('anonymous', 'poison'), 16.082814337672275)
(('asthmatic', 'breathing'), 16.082814337672275)
(('baron', 'zumpt'), 16.082814337672275)
(('bowing', 'repeatedly'), 16.082814337672275)
(('bridegroom', 'brilliancy'), 16.082814337672275)
(("brother's", 'braids'), 16.082814337672275)
(('bruised', 'limbs'), 16.082814337672275)
(('bulwer', 'lytton'), 16.082814337672275)
(('businesslike', 'brevity'), 16.082814337672275)
(('careful', 'resumption'), 16.082814337672275)
(('carnation', 'withered'), 16.082814337672275)
```

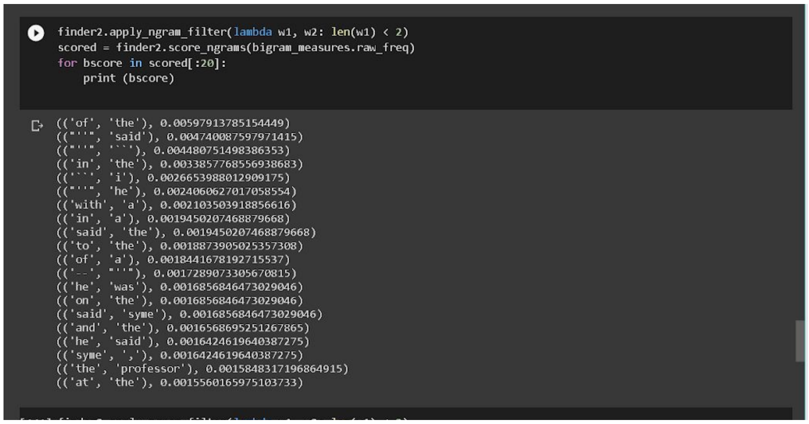
Stop words removed



```
finder.apply_word_filter(lambda w: w in stopwords)
scored = finder.score_ngrams(bigram_measures.raw_freq)
for bscore in scored[:20]:
    print (bscore)
```

```
(('said', 'syme'), 0.0016856846473029046)
(('dr.', 'bull'), 0.0009220839096357768)
(('ca', "n't"), 0.00040341171046565235)
(('asked', 'syme'), 0.00037459658828953437)
(('said', 'dr.'), 0.00036018902720147536)
(('de', 'worms'), 0.0003169663439372983)
(('professor', 'de'), 0.0002449285384970032)
(('said', 'gregory'), 0.0002449285384970032)
(('comrade', 'gregory'), 0.00020170585523282618)
(('old', 'man'), 0.00018729829414476719)
(('dr.', 'renard'), 0.00017289073305670817)
(('let', 'us'), 0.00017289073305670817)
(('looked', 'like'), 0.00017289073305670817)
(("n't", 'know'), 0.00017289073305670817)
(('mr.', 'syme'), 0.00015848317196864915)
(('old', 'gentleman'), 0.00015848317196864915)
(('red', 'hair'), 0.00015848317196864915)
(('colonel', 'ducroix'), 0.00014407561088059014)
(('blue', 'eyes'), 0.00012966804979253112)
(('cried', 'syme'), 0.00012966804979253112)
```

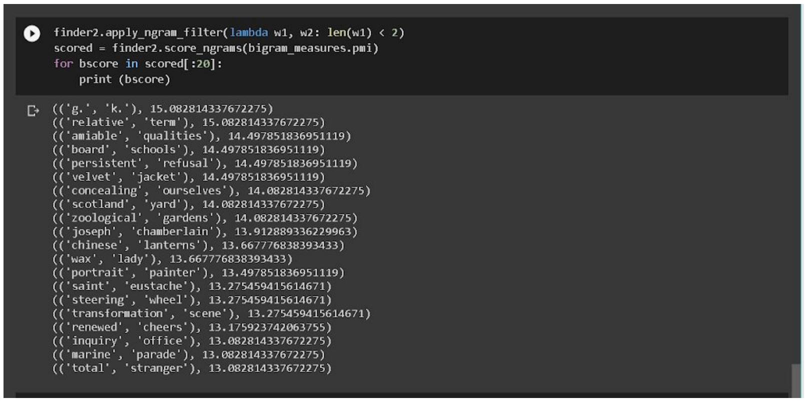
Words with length less than 2 removed



```
finder2.apply_ngram_filter(lambda w1, w2: len(w1) < 2)
scored = finder2.score_ngrams(bigram_measures.raw_freq)
for bscore in scored[:20]:
    print (bscore)
```

```
(('of', 'the'), 0.00597913785154449)
(("''", 'said'), 0.004740087597971415)
(("''", '``'), 0.004480751498386353)
(('in', 'the'), 0.0033857768556938683)
(('``', 'i'), 0.0026653988012909175)
(("''", 'he'), 0.0024060627017058554)
(('with', 'a'), 0.002103503918856616)
(('in', 'a'), 0.0019450207468879668)
(('said', 'the'), 0.0019450207468879668)
(('to', 'the'), 0.0018873905025357308)
(('of', 'a'), 0.0018441678192715537)
(('--', "''"), 0.0017289073305670815)
(('he', 'was'), 0.0016856846473029046)
(('on', 'the'), 0.0016856846473029046)
(('said', 'syme'), 0.0016856846473029046)
(('and', 'the'), 0.0016568695251267865)
(('he', 'said'), 0.0016424619640387275)
(('syme', ','), 0.0016424619640387275)
(('the', 'professor'), 0.0015848317196864915)
(('at', 'the'), 0.0015560165975103733)
```

Words with length less than 2 removed and PMI scoring mechanism



```
finder2.apply_ngram_filter(lambda w1, w2: len(w1) < 2)
scored = finder2.score_ngrams(bigram_measures.pmi)
for bscore in scored[:20]:
    print (bscore)
```

```
(('g.', 'k.'), 15.082814337672275)
(('relative', 'term'), 15.082814337672275)
(('amiable', 'qualities'), 14.497851836951119)
(('board', 'schools'), 14.497851836951119)
(('persistent', 'refusal'), 14.497851836951119)
(('velvet', 'jacket'), 14.497851836951119)
(('concealing', 'ourselves'), 14.082814337672275)
(('scotland', 'yard'), 14.082814337672275)
(('zoological', 'gardens'), 14.082814337672275)
(('joseph', 'chamberlain'), 13.912889336229963)
(('chinese', 'lanterns'), 13.667776838393433)
(('wax', 'lady'), 13.667776838393433)
(('portrait', 'painter'), 13.497851836951119)
(('saint', 'eustache'), 13.275459415614671)
(('steering', 'wheel'), 13.275459415614671)
(('transformation', 'scene'), 13.275459415614671)
(('renewed', 'cheers'), 13.175923742063755)
(('inquiry', 'office'), 13.082814337672275)
(('marine', 'parade'), 13.082814337672275)
(('total', 'stranger'), 13.082814337672275)
```

Steps:
1.Make a list of words from the data
2.Make an object of the BigramAssocMeasure class to use its functionalities
3.BigramCollocationFinder.from_words() use this function to make the list of bigrams with the list of words as the input
4.Use finder.score_ngrams() to score the bigrams to do further analysis.

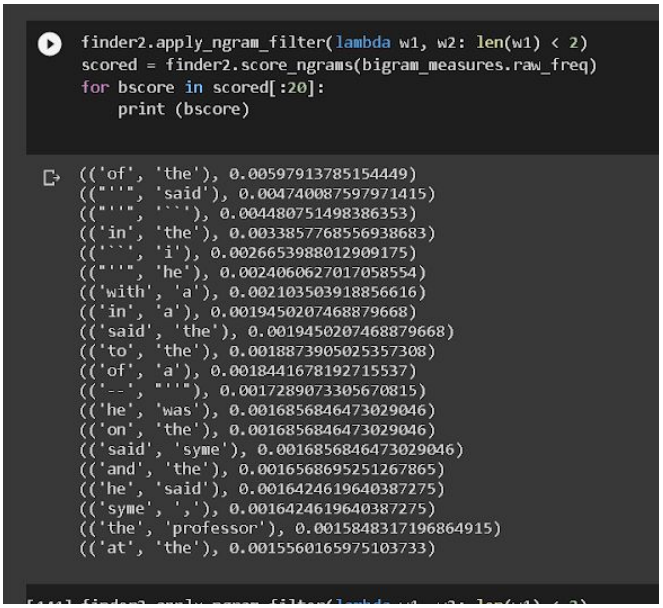
bigram_measures = nltk.collocations.BigramAssocMeasures()

finder = BigramCollocationFinder.from_words(emmawords)

scored = finder.score_ngrams(bigram_measures.raw_freq)

Observations:
The filter chosen by me is the length of words

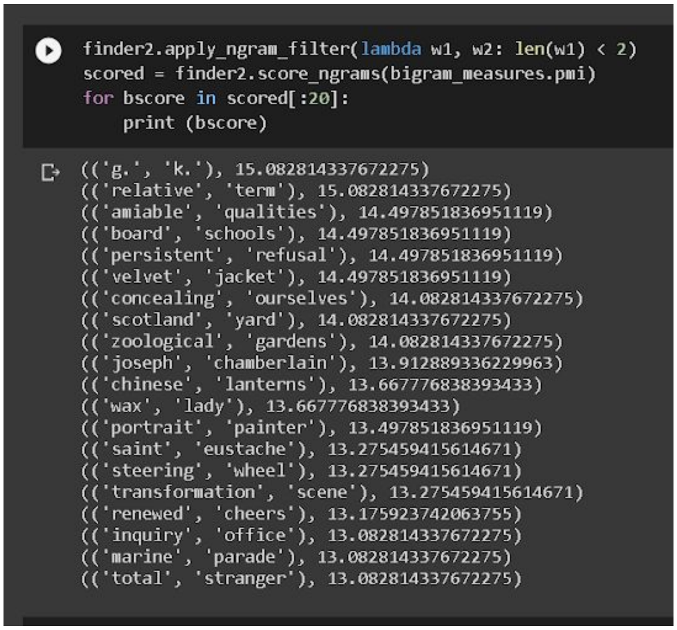
Raw Frequency scoring mechanism



```
finder2.apply_ngram_filter(lambda w1, w2: len(w1) < 2)
scored = finder2.score_ngrams(bigram_measures.raw_freq)
for bscore in scored[:20]:
    print (bscore)
```

```
(('of', 'the'), 0.00597913785154449)
(("''", 'said'), 0.004740087597971415)
(("''", '``'), 0.004480751498386353)
(('in', 'the'), 0.0033857768556938683)
(('``', 'i'), 0.0026653988012909175)
(("''", 'he'), 0.0024060627017058554)
(('with', 'a'), 0.0021035039188856616)
(('in', 'a'), 0.0019450207468879668)
(('said', 'the'), 0.0019450207468879668)
(('to', 'the'), 0.0018873905025357308)
(('of', 'a'), 0.0018441678192715537)
(('--', "''"), 0.0017289073305670815)
(('he', 'was'), 0.0016856846473029046)
(('on', 'the'), 0.0016856846473029046)
(('said', 'syme'), 0.0016856846473029046)
(('and', 'the'), 0.0016568695251267865)
(('he', 'said'), 0.0016424619640387275)
(('syme', ','), 0.0016424619640387275)
(('the', 'professor'), 0.0015848317196864915)
(('at', 'the'), 0.0015560165975103733)
```

PMI scoring mechanism



```
finder2.apply_ngram_filter(lambda w1, w2: len(w1) < 2)
scored = finder2.score_ngrams(bigram_measures.pmi)
for bscore in scored[:20]:
    print (bscore)
```

```
(('g.', 'k.'), 15.082814337672275)
(('relative', 'term'), 15.082814337672275)
(('amiable', 'qualities'), 14.497851836951119)
(('board', 'schools'), 14.497851836951119)
(('persistent', 'refusal'), 14.497851836951119)
(('velvet', 'jacket'), 14.497851836951119)
(('concealing', 'ourselves'), 14.082814337672275)
(('scotland', 'yard'), 14.082814337672275)
(('zoological', 'gardens'), 14.082814337672275)
(('joseph', 'chamberlain'), 13.912889336229963)
(('chinese', 'lanterns'), 13.667776838393433)
(('wax', 'lady'), 13.667776838393433)
(('portrait', 'painter'), 13.497851836951119)
(('saint', 'eustache'), 13.275459415614671)
(('steering', 'wheel'), 13.275459415614671)
(('transformation', 'scene'), 13.275459415614671)
(('renewed', 'cheers'), 13.175923742063755)
(('inquiry', 'office'), 13.082814337672275)
(('marine', 'parade'), 13.082814337672275)
(('total', 'stranger'), 13.082814337672275)
```

The filter with a length greater than 2 took care of two things it elimination small words and also took care of punctuation too because it can be seen from the two screenshots that ' " " ' was eliminated from the word list even though I didn't put a filter for any non-alphabetical characters

1. Joseph Chamberlain: I came across was a British statesman who was first a radical Liberal, In chapter 2 he and Gregory visit a bar where they encounter detective Sym

Lessons learned

1. Was able to clearly understand and access the file through Jupyter Notebook
2. Proper usage of the lambda function
3.Make bigrams from the list of words
4. Scoring the bigrams for further analysis