NLP Assignment

Name: Nikhita Rapolu

Roll No: Se20ucse115

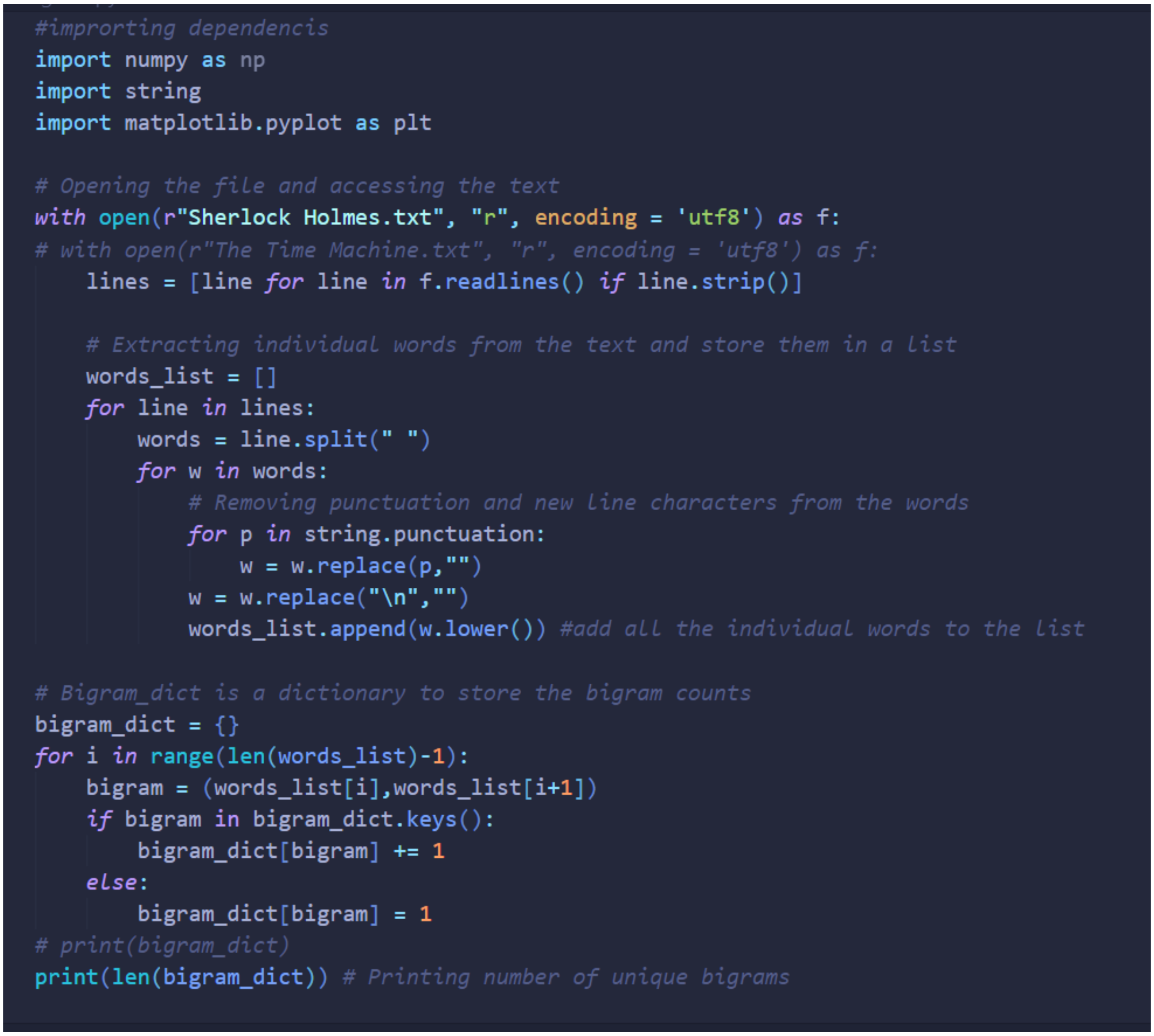
Collocation is the expression of multiple words that frequently occur together in the corpus. They can be used to identify idiomatic expressions and help us understand the meaning and usage of words in context. In NLP, collocation analysis is used to identify and extract such word combinations automatically from large amounts of text. In natural language processing, one common method for verifying the collocation property of a word combination is to calculate the statistical measure called Pointwise Mutual Information (PMI).

Unigrams: These are single words or tokens that are considered as individual units, and their frequency count is calculated in a corpus of text.

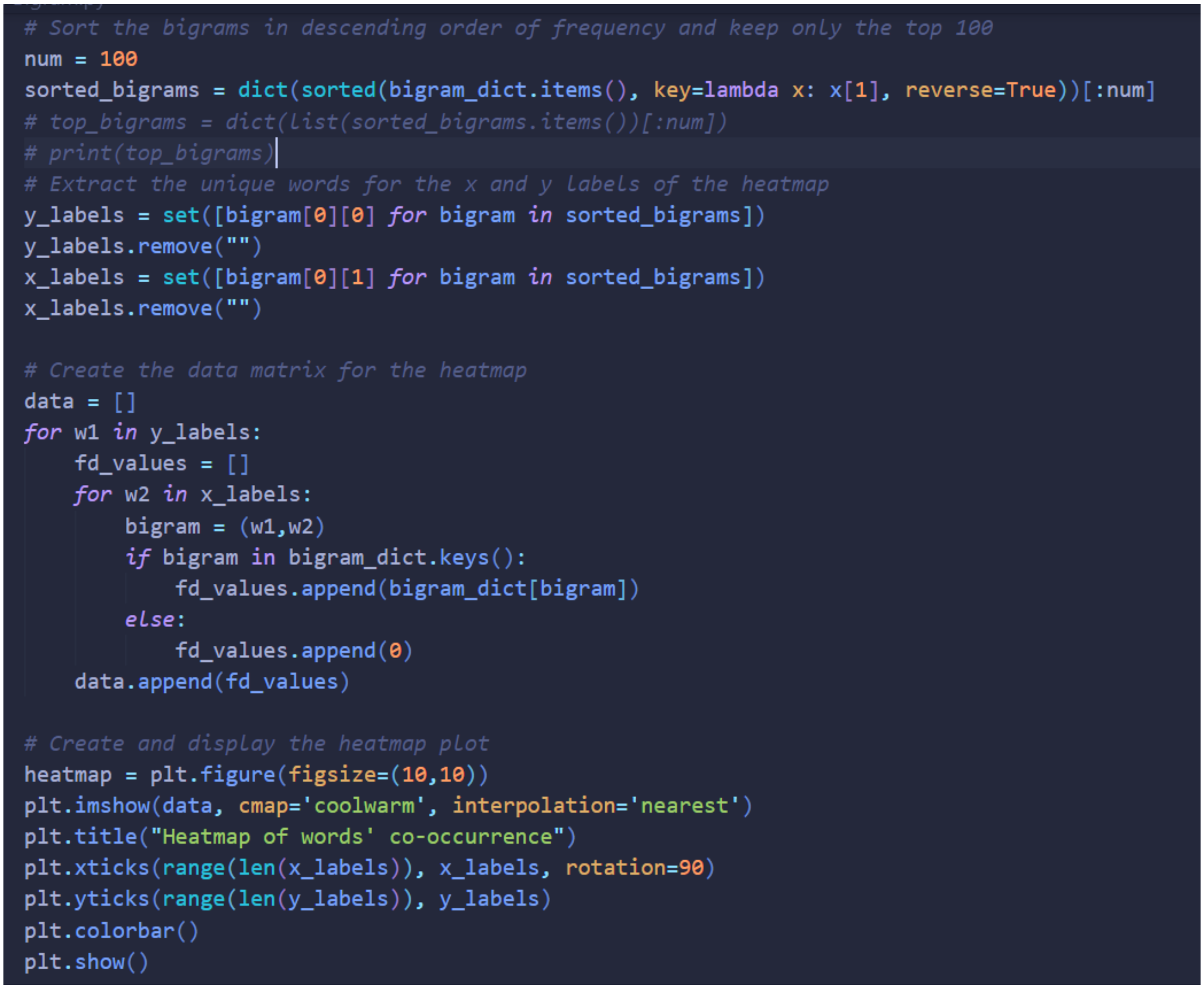
Bigrams: These are pairs of adjacent words in a text, and their frequency count is calculated in a corpus of text.

Trigrams: These are sequences of three consecutive words in a text, and their frequency count is calculated in a corpus of text.

The code is to verify the collocation property for bigrams. The file is opened and the lines are read into a list. Each line is split into individual words, which are cleaned up (punctuation and new line characters are removed) and added to a list. The list of words is used to create a dictionary of bigram counts. Each bigram is represented as a tuple of two consecutive words. The counts are accumulated in the dictionary. The dictionary of bigram counts is sorted in descending order of frequency and the top 100 bigrams are selected. The unique words in the top 100 bigrams are extracted to create the x and y labels for the heatmap. The data matrix for the heatmap is created using the bigram counts and the unique words as row and column indices. The heatmap is created using Matplotlib and displayed. The resulting heatmap provides a visual representation of the co-occurrence patterns of the top 100 bigrams in the text.

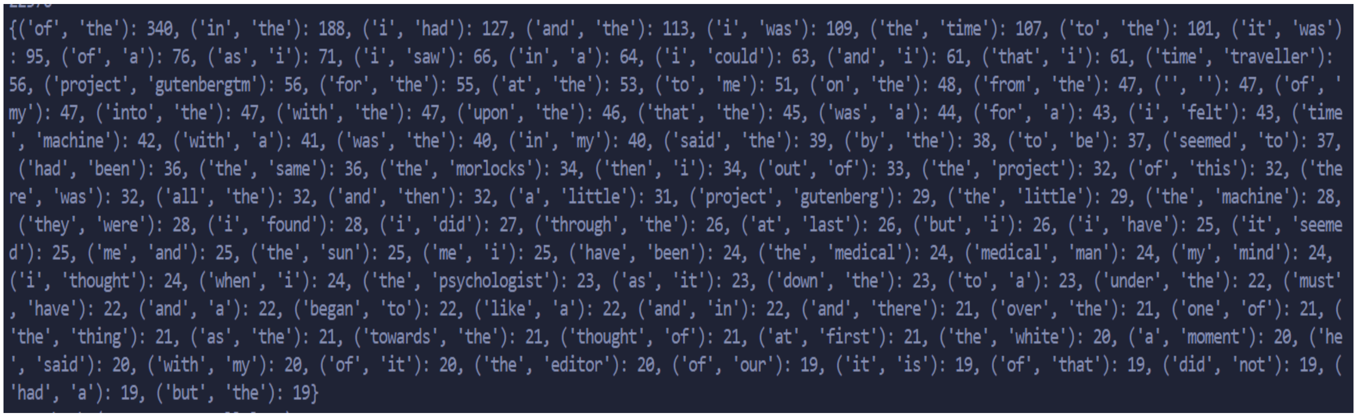


```
#improrting dependencis
import numpy as np
import string
import matplotlib.pyplot as plt

# Opening the file and accessing the text
with open(r"Sherlock Holmes.txt", "r", encoding = 'utf8') as f:
# with open(r"The Time Machine.txt", "r", encoding = 'utf8') as f:
    lines = [line for line in f.readlines() if line.strip()]

    # Extracting individual words from the text and store them in a list
    words_list = []
    for line in lines:
        words = line.split(" ")
        for w in words:
            # Removing punctuation and new line characters from the words
            for p in string.punctuation:
                w = w.replace(p,"")
            w = w.replace("\n","")
            words_list.append(w.lower()) #add all the individual words to the list

# Bigram_dict is a dictionary to store the bigram counts
bigram_dict = {}
for i in range(len(words_list)-1):
    bigram = (words_list[i],words_list[i+1])
    if bigram in bigram_dict.keys():
        bigram_dict[bigram] += 1
    else:
        bigram_dict[bigram] = 1
# print(bigram_dict)
print(len(bigram_dict)) # Printing number of unique bigrams
```

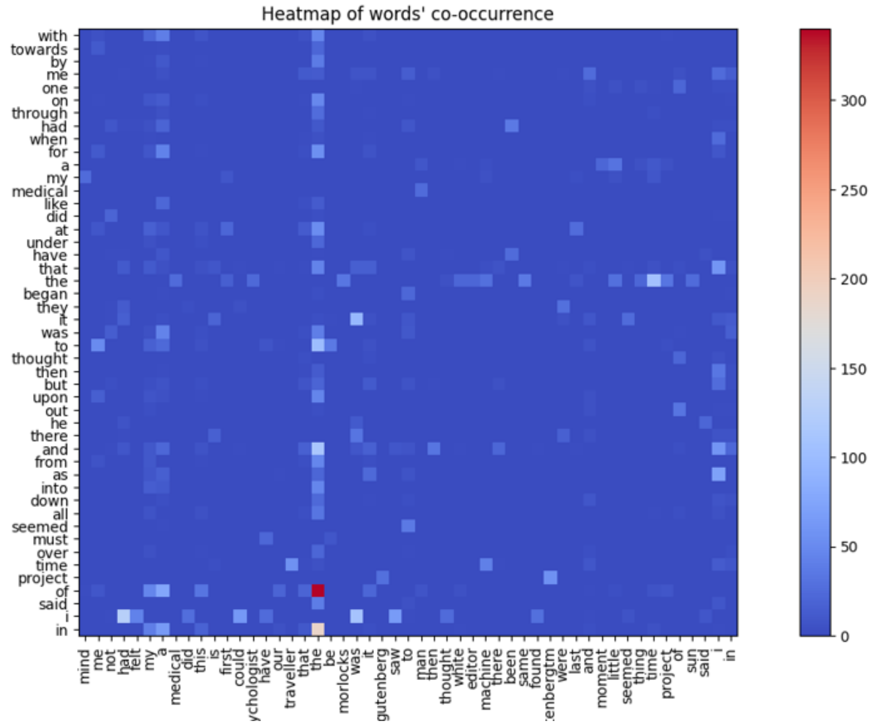


```
# Sort the bigrams in descending order of frequency and keep only the top 100
num = 100
sorted_bigrams = dict(sorted(bigram_dict.items(), key=lambda x: x[1], reverse=True))[:num]
# top_bigrams = dict(list(sorted_bigrams.items())[:num])
# print(top_bigrams)
# Extract the unique words for the x and y labels of the heatmap
y_labels = set([bigram[0][0] for bigram in sorted_bigrams])
y_labels.remove("")
x_labels = set([bigram[0][1] for bigram in sorted_bigrams])
x_labels.remove("")

# Create the data matrix for the heatmap
data = []
for w1 in y_labels:
    fd_values = []
    for w2 in x_labels:
        bigram = (w1,w2)
        if bigram in bigram_dict.keys():
            fd_values.append(bigram_dict[bigram])
        else:
            fd_values.append(0)
    data.append(fd_values)

# Create and display the heatmap plot
heatmap = plt.figure(figsize=(10,10))
plt.imshow(data, cmap='coolwarm', interpolation='nearest')
plt.title("Heatmap of words' co-occurrence")
plt.xticks(range(len(x_labels)), x_labels, rotation=90)
plt.yticks(range(len(y_labels)), y_labels)
plt.colorbar()
plt.show()
```

The Time Traveller

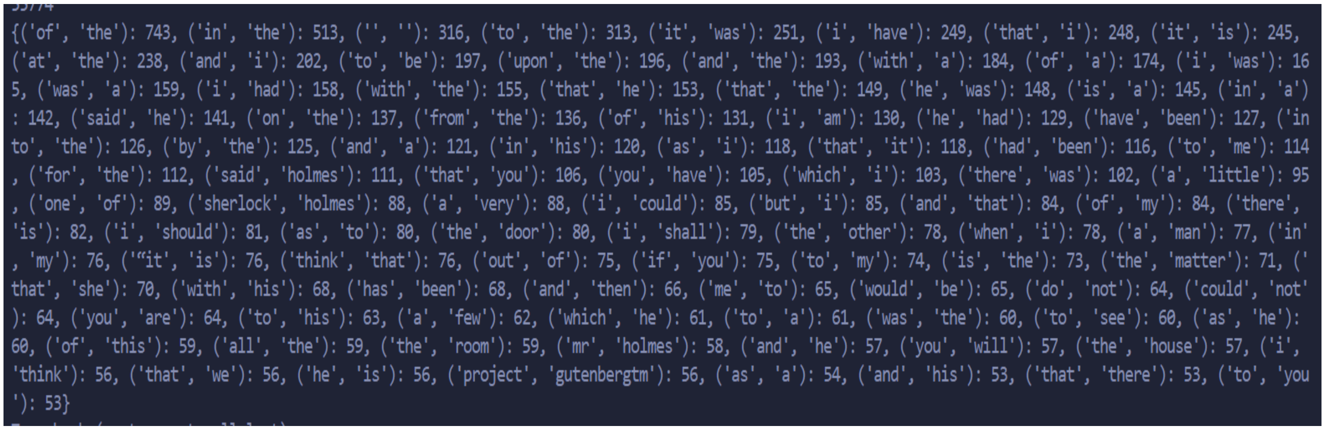


```
{('of', 'the'): 340, ('in', 'the'): 188, ('i', 'had'): 127, ('and', 'the'): 113, ('i', 'was'): 109, ('the', 'time'): 107, ('to', 'the'): 101, ('it', 'was')
: 95, ('of', 'a'): 76, ('as', 'i'): 71, ('i', 'saw'): 66, ('in', 'a'): 64, ('i', 'could'): 63, ('and', 'i'): 61, ('that', 'i'): 61, ('time', 'traveller'):
56, ('project', 'gutenbergtm'): 56, ('for', 'the'): 55, ('at', 'the'): 53, ('to', 'me'): 51, ('on', 'the'): 48, ('from', 'the'): 47, ('', ''): 47, ('of', '
my'): 47, ('into', 'the'): 47, ('with', 'the'): 47, ('upon', 'the'): 46, ('that', 'the'): 45, ('was', 'a'): 44, ('for', 'a'): 43, ('i', 'felt'): 43, ('time
', 'machine'): 42, ('with', 'a'): 41, ('was', 'the'): 40, ('in', 'my'): 40, ('said', 'the'): 39, ('by', 'the'): 38, ('to', 'be'): 37, ('seemed', 'to'): 37,
 ('had', 'been'): 36, ('the', 'same'): 36, ('the', 'morlocks'): 34, ('then', 'i'): 34, ('out', 'of'): 33, ('the', 'project'): 32, ('of', 'this'): 32, ('the
re', 'was'): 32, ('all', 'the'): 32, ('and', 'then'): 32, ('a', 'little'): 31, ('project', 'gutenberg'): 29, ('the', 'little'): 29, ('the', 'machine'): 28,
 ('they', 'were'): 28, ('i', 'found'): 28, ('i', 'did'): 27, ('through', 'the'): 26, ('at', 'last'): 26, ('but', 'i'): 26, ('i', 'have'): 25, ('it', 'seeme
d'): 25, ('me', 'and'): 25, ('the', 'sun'): 25, ('me', 'i'): 25, ('have', 'been'): 24, ('the', 'medical'): 24, ('medical', 'man'): 24, ('my', 'mind'): 24,
('i', 'thought'): 24, ('when', 'i'): 24, ('the', 'psychologist'): 23, ('as', 'it'): 23, ('down', 'the'): 23, ('to', 'a'): 23, ('under', 'the'): 22, ('must'
, 'have'): 22, ('and', 'a'): 22, ('began', 'to'): 22, ('like', 'a'): 22, ('and', 'in'): 22, ('and', 'there'): 21, ('over', 'the'): 21, ('one', 'of'): 21, (
'the', 'thing'): 21, ('as', 'the'): 21, ('towards', 'the'): 21, ('thought', 'of'): 21, ('at', 'first'): 21, ('the', 'white'): 20, ('a', 'moment'): 20, ('he
', 'said'): 20, ('with', 'my'): 20, ('of', 'it'): 20, ('the', 'editor'): 20, ('of', 'our'): 19, ('it', 'is'): 19, ('of', 'that'): 19, ('did', 'not'): 19, (
'had', 'a'): 19, ('but', 'the'): 19}
```

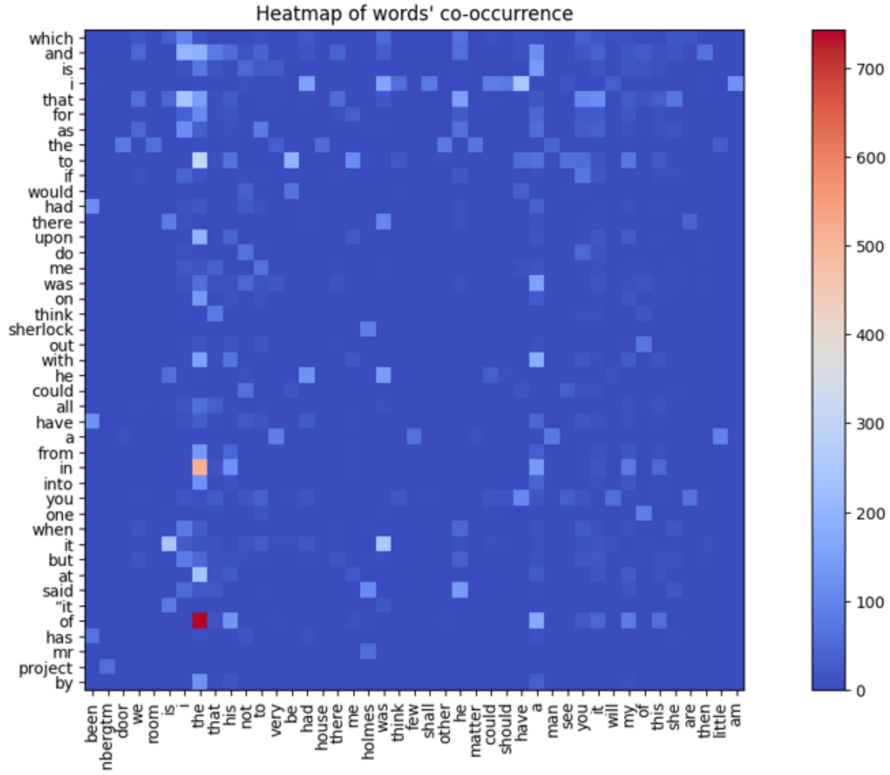


Heatmap of words' co-occurrence

Sherlock Holmes



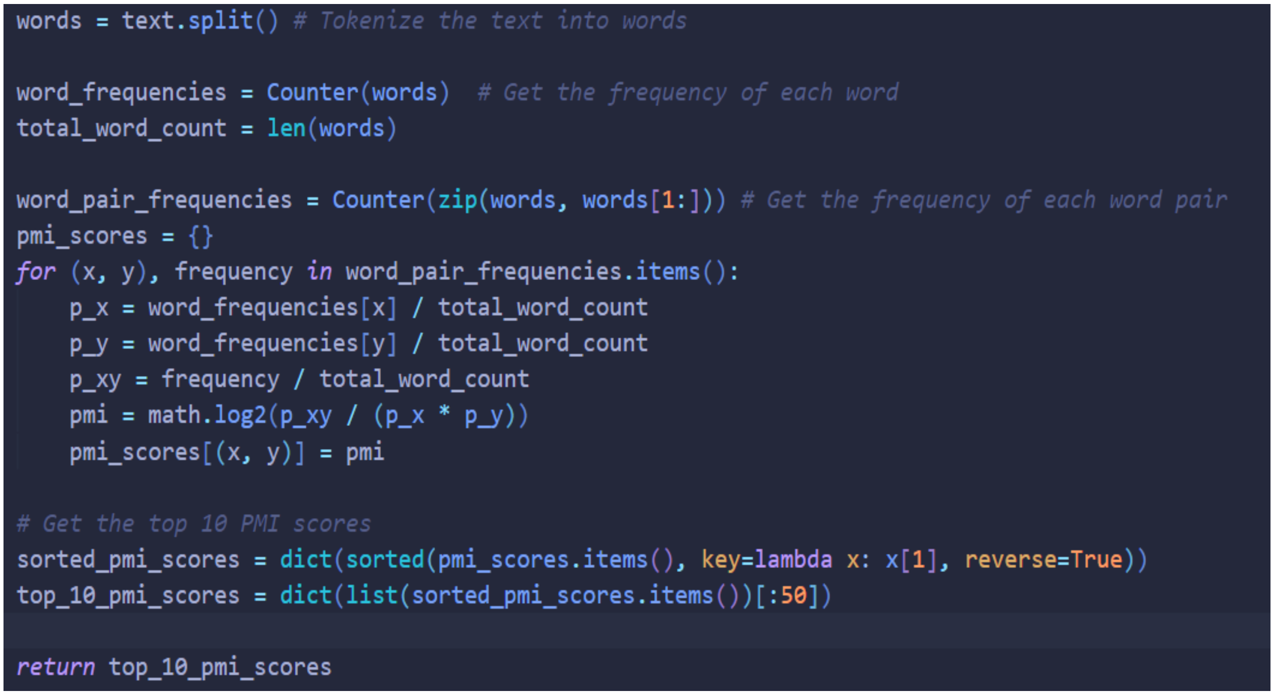
```
{('of', 'the'): 743, ('in', 'the'): 513, ('', ''): 316, ('to', 'the'): 313, ('it', 'was'): 251, ('i', 'have'): 249, ('that', 'i'): 248, ('it', 'is'): 245,
('at', 'the'): 238, ('and', 'i'): 202, ('to', 'be'): 197, ('upon', 'the'): 196, ('and', 'the'): 193, ('with', 'a'): 184, ('of', 'a'): 174, ('i', 'was'): 16
5, ('was', 'a'): 159, ('i', 'had'): 158, ('with', 'the'): 155, ('that', 'he'): 153, ('that', 'the'): 149, ('he', 'was'): 148, ('is', 'a'): 145, ('in', 'a')
: 142, ('said', 'he'): 141, ('on', 'the'): 137, ('from', 'the'): 136, ('of', 'his'): 131, ('i', 'am'): 130, ('he', 'had'): 129, ('have', 'been'): 127, ('in
to', 'the'): 126, ('by', 'the'): 125, ('and', 'a'): 121, ('in', 'his'): 120, ('as', 'i'): 118, ('that', 'it'): 118, ('had', 'been'): 116, ('to', 'me'): 114
, ('for', 'the'): 112, ('said', 'holmes'): 111, ('that', 'you'): 106, ('you', 'have'): 105, ('which', 'i'): 103, ('there', 'was'): 102, ('a', 'little'): 95
, ('one', 'of'): 89, ('sherlock', 'holmes'): 88, ('a', 'very'): 88, ('i', 'could'): 85, ('but', 'i'): 85, ('and', 'that'): 84, ('of', 'my'): 84, ('there',
'is'): 82, ('i', 'should'): 81, ('as', 'to'): 80, ('the', 'door'): 80, ('i', 'shall'): 79, ('the', 'other'): 78, ('when', 'i'): 78, ('a', 'man'): 77, ('in'
, 'my'): 76, ('“it', 'is'): 76, ('think', 'that'): 76, ('out', 'of'): 75, ('if', 'you'): 75, ('to', 'my'): 74, ('is', 'the'): 73, ('the', 'matter'): 71, ('
that', 'she'): 70, ('with', 'his'): 68, ('has', 'been'): 68, ('and', 'then'): 66, ('me', 'to'): 65, ('would', 'be'): 65, ('do', 'not'): 64, ('could', 'not'
): 64, ('you', 'are'): 64, ('to', 'his'): 63, ('a', 'few'): 62, ('which', 'he'): 61, ('to', 'a'): 61, ('was', 'the'): 60, ('to', 'see'): 60, ('as', 'he'):
60, ('of', 'this'): 59, ('all', 'the'): 59, ('the', 'room'): 59, ('mr', 'holmes'): 58, ('and', 'he'): 57, ('you', 'will'): 57, ('the', 'house'): 57, ('i',
'think'): 56, ('that', 'we'): 56, ('he', 'is'): 56, ('project', 'gutenbergtm'): 56, ('as', 'a'): 54, ('and', 'his'): 53, ('that', 'there'): 53, ('to', 'you
'): 53}
```



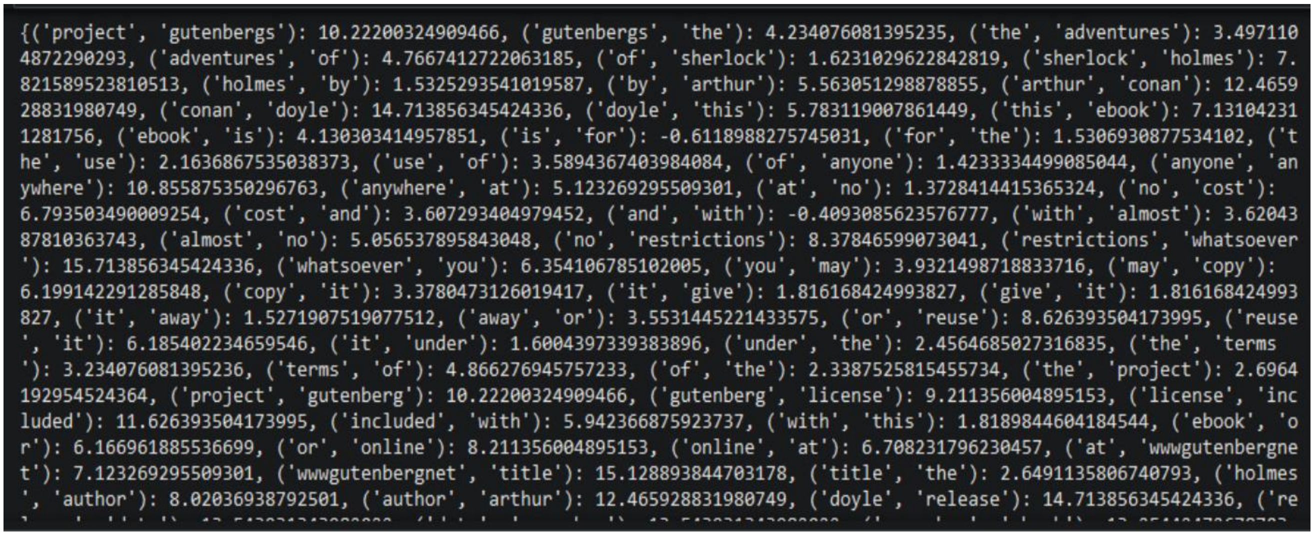
From the heatmap, we can see the top 100 bigrams and their respective frequencies.

Calculating PMI from the bigrams:

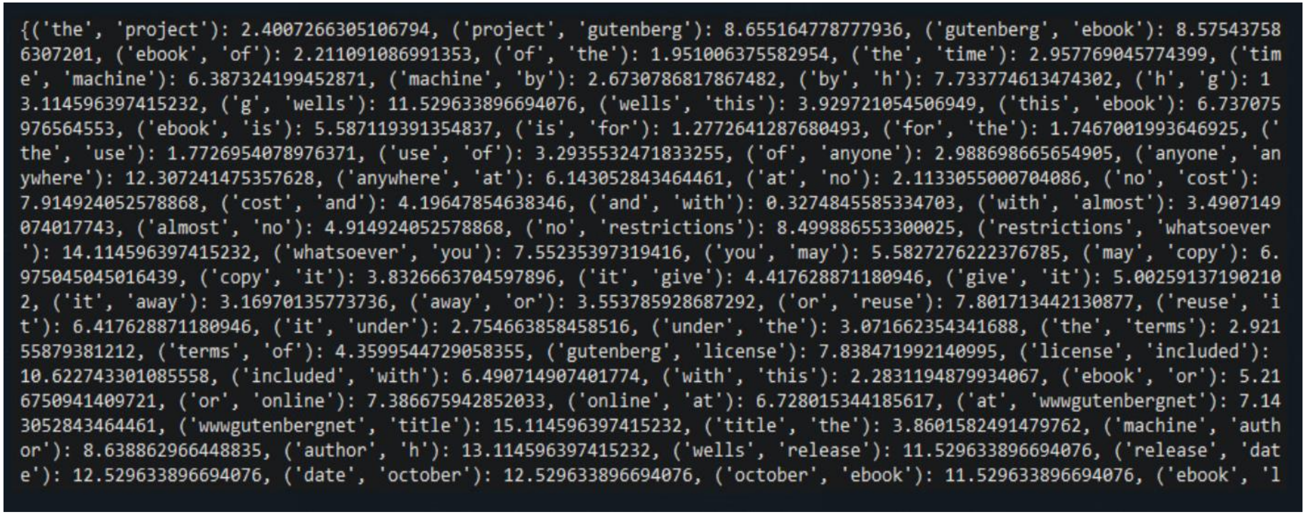
$$PMI(w^1, w^2) = log_2 \frac{P(w^1, w^2)}{P(w^1)P(w^2)}$$



```
words = text.split() # Tokenize the text into words

word_frequencies = Counter(words)  # Get the frequency of each word
total_word_count = len(words)

word_pair_frequencies = Counter(zip(words, words[1:])) # Get the frequency of each word pair
pmi_scores = {}
for (x, y), frequency in word_pair_frequencies.items():
    p_x = word_frequencies[x] / total_word_count
    p_y = word_frequencies[y] / total_word_count
    p_xy = frequency / total_word_count
    pmi = math.log2(p_xy / (p_x * p_y))
    pmi_scores[(x, y)] = pmi

# Get the top 10 PMI scores
sorted_pmi_scores = dict(sorted(pmi_scores.items(), key=lambda x: x[1], reverse=True))
top_10_pmi_scores = dict(list(sorted_pmi_scores.items())[:50])

return top_10_pmi_scores
```



```
{('project', 'gutenbergs'): 10.22200324909466, ('gutenbergs', 'the'): 4.234076081395235, ('the', 'adventures'): 3.497110
4872290293, ('adventures', 'of'): 4.7667412722063185, ('of', 'sherlock'): 1.6231029622842819, ('sherlock', 'holmes'): 7.
821589523810513, ('holmes', 'by'): 1.5325293541019587, ('by', 'arthur'): 5.563051298878855, ('arthur', 'conan'): 12.4659
28831980749, ('conan', 'doyle'): 14.713856345424336, ('doyle', 'this'): 5.783119007861449, ('this', 'ebook'): 7.13104231
1281756, ('ebook', 'is'): 4.130303414957851, ('is', 'for'): -0.6118988275745031, ('for', 'the'): 1.5306930877534102, ('t
he', 'use'): 2.1636867535038373, ('use', 'of'): 3.5894367403984084, ('of', 'anyone'): 1.4233334499085044, ('anyone', 'an
ywhere'): 10.855875350296763, ('anywhere', 'at'): 5.123269295509301, ('at', 'no'): 1.3728414415365324, ('no', 'cost'):
6.793503490009254, ('cost', 'and'): 3.607293404979452, ('and', 'with'): -0.4093085623576777, ('with', 'almost'): 3.62043
87810363743, ('almost', 'no'): 5.056537895843048, ('no', 'restrictions'): 8.37846599073041, ('restrictions', 'whatsoever
'): 15.713856345424336, ('whatsoever', 'you'): 6.354106785102005, ('you', 'may'): 3.9321498718833716, ('may', 'copy'):
6.199142291285848, ('copy', 'it'): 3.3780473126019417, ('it', 'give'): 1.816168424993827, ('give', 'it'): 1.816168424993
827, ('it', 'away'): 1.5271907519077512, ('away', 'or'): 3.5531445221433575, ('or', 'reuse'): 8.626393504173995, ('reuse
', 'it'): 6.185402234659546, ('it', 'under'): 1.6004397339383896, ('under', 'the'): 2.4564685027316835, ('the', 'terms
'): 3.234076081395236, ('terms', 'of'): 4.866276945757233, ('of', 'the'): 2.3387525815455734, ('the', 'project'): 2.6964
192954524364, ('project', 'gutenberg'): 10.22200324909466, ('gutenberg', 'license'): 9.211356004895153, ('license', 'inc
luded'): 11.626393504173995, ('included', 'with'): 5.942366875923737, ('with', 'this'): 1.8189844604184544, ('ebook', 'o
r'): 6.166961885536699, ('or', 'online'): 8.211356004895153, ('online', 'at'): 6.708231796230457, ('at', 'wwwgutenbergne
t'): 7.123269295509301, ('wwwgutenbergnet', 'title'): 15.128893844703178, ('title', 'the'): 2.6491135806740793, ('holmes
', 'author'): 8.02036938792501, ('author', 'arthur'): 12.465928831980749, ('doyle', 'release'): 14.713856345424336, ('re
```



```
{('the', 'project'): 2.4007266305106794, ('project', 'gutenberg'): 8.655164778777936, ('gutenberg', 'ebook'): 8.57543758
6307201, ('ebook', 'of'): 2.211091086991353, ('of', 'the'): 1.951006375582954, ('the', 'time'): 2.957769045774399, ('tim
e', 'machine'): 6.387324199452871, ('machine', 'by'): 2.6730786817867482, ('by', 'h'): 7.733774613474302, ('h', 'g'): 1
3.114596397415232, ('g', 'wells'): 11.529633896694076, ('wells', 'this'): 3.929721054506949, ('this', 'ebook'): 6.737075
976564553, ('ebook', 'is'): 5.587119391354837, ('is', 'for'): 1.2772641287680493, ('for', 'the'): 1.7467001993646925, ('
the', 'use'): 1.7726954078976371, ('use', 'of'): 3.2935532471833255, ('of', 'anyone'): 2.988698665654905, ('anyone', 'an
ywhere'): 12.307241475357628, ('anywhere', 'at'): 6.143052843464461, ('at', 'no'): 2.1133055000704086, ('no', 'cost'):
7.914924052578868, ('cost', 'and'): 4.19647854638346, ('and', 'with'): 0.3274845585334703, ('with', 'almost'): 3.4907149
074017743, ('almost', 'no'): 4.914924052578868, ('no', 'restrictions'): 8.499886553300025, ('restrictions', 'whatsoever
'): 14.114596397415232, ('whatsoever', 'you'): 7.55235397319416, ('you', 'may'): 5.5827276222376785, ('may', 'copy'): 6.
975045045016439, ('copy', 'it'): 3.8326663704597896, ('it', 'give'): 4.417628871180946, ('give', 'it'): 5.00259137190210
2, ('it', 'away'): 3.16970135773736, ('away', 'or'): 3.553785928687292, ('or', 'reuse'): 7.801713442130877, ('reuse', 'i
t'): 6.417628871180946, ('it', 'under'): 2.754663858458516, ('under', 'the'): 3.071662354341688, ('the', 'terms'): 2.921
55879381212, ('terms', 'of'): 4.3599544729058355, ('gutenberg', 'license'): 7.838471992140995, ('license', 'included'):
10.622743301085558, ('included', 'with'): 6.490714907401774, ('with', 'this'): 2.2831194879934067, ('ebook', 'or'): 5.21
6750941409721, ('or', 'online'): 7.386675942852033, ('online', 'at'): 6.728015344185617, ('at', 'wwwgutenbergnet'): 7.14
3052843464461, ('wwwgutenbergnet', 'title'): 15.114596397415232, ('title', 'the'): 3.8601582491479762, ('machine', 'auth
or'): 8.638862966448835, ('author', 'h'): 13.114596397415232, ('wells', 'release'): 11.529633896694076, ('release', 'dat
e'): 12.529633896694076, ('date', 'october'): 12.529633896694076, ('october', 'ebook'): 11.529633896694076, ('ebook', 'l
```

If the PMI value for a word combination is high, it indicates that the two words are strongly associated and likely to form a collocation. A PMI value above 3 is considered significant, while values above 6 are very strong evidence of a collocation. However, PMI has some limitations, and other statistical measures like t-score and z-score are also used to verify collocation property. It's worth noting that collocations are not just based on statistical measures, but also on semantic and syntactic properties of the words.

From the Sherlock Holmes text we can analyze that the words “of the” has occurred 743 times. Co-occurrences may not be sufficient as phrases such as ‘of the’ may co-occur frequently, but are not meaningful.

“Sherlock Holmes” has a frequency of 96, this is a meaningful collocation as it occures together. Therefore, human judgment is also needed to validate the collocation property of a word combination.