deerwalk
DWIT College

Lab Report 6

Submitted by:

Nabina Khadka

0516

2019

Submitted to:

Birodh Rijal

(Artificial Intelligence Lecturer)

1.2.1 Part A

Your task is to read the contents of the file and produce:

1. A table containing 20 most frequent words. The table contains three columns: rank, word and frequency.

First, file read and all the words were inserted in a list. Then, I iterated through the list to create a dictionary where the key was the word and the value was the frequency of the word. Then I displayed the key-value pairs in the descending order of the values (the frequencies).

I have used a library `tabulate`, so this library must be installed to run the script provided. Its use is to display outputs as included in the figures below.

```
PART A

Question 1
A table containing 20 most frequent words. The table contains three columns: rank, word and frequency.

======  ======  ===========
 Rank   Word     Frequency
======  ======  ===========
     1  the           26856
     2  and           24116
     3  i             22412
     4  to            19225
     5  of            16018
     6  you           14097
     7  a             13986
     8  my            12283
     9  that          11171
    10  in            10640
    11  is             9271
    12  d              8608
    13  not            8466
    14  it             7783
    15  me             7759
    16  for            7645
    17  s              7264
    18  with           7157
    19  be             6891
    20  your           6756
======  ======  ===========
```

2. **A table, containing list of bottom frequencies. The table contains three columns: frequency, word count and example words. You are supposed to print word counts for frequencies 10 to 1. The rows in this table show how many words have frequency 10, 9 , 8...1 with example of some of the words.**

To, solve this question I iterated the dictionary in ascending order of values (the frequencies). The element was added to the example list. Whenever a change in frequency was detected say 1 to 2, then a new list was created which has 3 elements the rank, frequency and the example list. Then I printed the list in ascending order of rank.

```
Question 2
A table, containing list of bottom frequencies.


===========  ============  =====================================
  Frequency   Word Count   Examples
===========  ============  =====================================
          1         8543   guiltian ,quasi ,combless ,felonious
          2         3229   chime ,ribbed ,collied ,invert
          3         1831   ild ,talons ,forthcoming ,allure
          4         1311   bruit ,bravest ,allons ,soothsayer
          5          904   sayings ,approves ,stabbed ,turtles
          6          743   syria ,bagot ,rival ,incur
          7          530   fog ,caper ,gentlewomen ,od
          8          429   posterity ,dealt ,appearing ,certainty
          9          373   menas ,mildly ,appeared ,filth
         10          322   arrows ,agent ,necks ,dig
===========  ============  =====================================
```

3. A table containing 20 most frequent word-pairs (bigrams). The table contains three columns: rank, word pair and frequency.

I used the word list to create a list of bigrams, then I created a dictionary as in question 1. The key part was the bigram and the value was the frequency of the bigram. Then I iterated through the dictionary in descending order of the value (frequency bigram).

```
Question 3
A table containing 20 most frequent word-pairs (bigrams). The table contains three columns: rank, word pair and frequency.

======  ===========  ===========
 Rank   Word Pair      Frequency
======  ===========  ===========
     1  i am                1858
     2  i ll                1784
     3  my lord             1699
     4  i have              1631
     5  in the              1585
     6  i will              1582
     7  to the              1518
     8  of the              1380
     9  it is               1087
    10  to be                971
    11  that i               964
    12  and i                830
    13  i do                 829
    14  the king             784
    15  and the              728
    16  you are              724
    17  of my                696
    18  is the               692
    19  i would              674
    20  he is                658
======  ===========  ===========
```

1.2.2 Part B

With the frequency counts of the word at our hand we calculate some basic probability estimates.

1. **Calculate the relative frequency (probability estimate) of the words: (a) "the" (b) "become" (d) "brave" (e) "treason" [Note: P(the) = count(the) / N . Here, count(the) is the frequency of "the" and "N" is the total word count.]**

The relative frequency of a word is calculated as P(word) = count(word)/total_no_of_word

The count of a word is determined from the dictionary (where the key is the word and the value id the frequency). The total_no_of_words is the length of the wordlist itself.

```
PART B:

Question 1:
Calculate the relative frequency (probability estimate) of the words:

The relative frequency of 'the' is 0.032018236183372836
The relative frequency of 'become' is 0.006264410318875886
The relative frequency of 'brave' is 0.006829947361552182
The relative frequency of 'treason' is 0.004176273545917258
```

2. **Calculate the following word conditional probabilities: (a) P(court | The) (b) P(word | his) (c) P(qualities | rare) (d) P(men | young) [Read P(B | A) as "the probability with which word B follows word A". Note: P(B | A) = count(A;B) | count(A)]**

The probability is calculated as

P(A/B) = count(a,b)/count(a)

count(a,b) was extracted from the bigram dictionary and count(a) was extracted from word dictionary.

```
Question 2:
Calculate the following word conditional probabilities:

P(court | The) = 0.001377718200774501
P(word | his) = 0.001377718200774501
P(qualities | rare) = 0.017857142857142856
P(men | young) = 0.026004728132387706
```

3. **Calculate the probability: (a) P(have, sent) (b) P(will, look, upon) (c) P(I, am, no, baby) (d) P(wherefore, art, thou, Romeo).**

Using Markov assumption, the probability of P(A,B,C,D) is calculated as

P(A,B,C,D) = P(A) * P(B|A) * P(C|B) * P(D|C)

```
Question 3
Calculate the probability:

P(have, sent) = 0.0050036091992613732
P(will, look, upon) = 9.24793159207516e-07
P(I, am, no, baby) = 1.2479985075725128e-08
P(wherefore, art, thou, Romeo) = 2.2224737028306635e-10
```

4. **Calculate probabilities in Q3 assuming each word is independent of other words (independence assumption).**

If the words are considered to be independent the probabilities are given as

P(A,B,C,D) = P(A)*P(B)*P(C)*P(D), the result is shone below.

```
Question 4
Calculate probabilities in Q3 assuming each word is independent of other words

P(have, sent) = 2.277676449721073e-06
P(will, look, upon) = 1.3496632929942172e-08
P(I, am, no, baby) = 6.284914714326714e-12
P(wherefore, art, thou, Romeo) = 1.7479309085767736e-13
```

5. **Find the most probable word to follow this sequence of words: (a) I am no (b) wherefore art thou**

To solve this all the words that come after the last words (here 'no' and 'thou') were added to a list. For this the word list was used. Then, conditional probability of those words coming after the given word sequence was calculated and the element having the highest probability was used. The output is illustrated below.

```
Question 5
Find the most probable word to follow this sequence of words:

a. I am no
I am no more
b. wherefore art thou
wherefore art thou art
```