Lab Report 5

Submitted by:

Shreeya Pandey

547

Submitted to:

Birodh Rijal

(AI)

Introduction

In this lab, we will consider a file named ‘shakespeare.txt’ which contains collection of writings from the author William Shakespeare. Here, we will deal with probabilities. We will be tokenizing the text file and calculate probability of occurrence of terms, conditional probabilities, probabilities of dependent or independent occurrences and to make some predictions.

Methodology:

First, the file was read and all the words were inserted in a list. Then, iteration through the list was done to create a dictionary where the key was the word and the value was the frequency of the word. The dictionary was sorted. Library tabulate was used. This library must be installed to run the script provided.

Finally, to predict what word will follow current sentence, chain conditional probability was calculated using Markov’s assumption.
(i.e P(A,B,C,D) = P(A)*P(B|A)*P(C|B)*P(D|C)). Here D is our prediction and A, B, C are our previous words. We take the word D as the word with the highest probability among all other possible candidates.

Implementation

This program was implemented using python.

Part A:

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

```
The top 20 frequent words are:
[(1, 'the', 26851), (2, 'and', 24077), (3, 'i', 20535), (4, 'to', 18561), (5, 'of', 16013), (6, 'you', 13856), (7, 'a', 13840), (8, 'my', 12282), (9, 'that', 10761), (10, 'in',
 10537), (11, 'is', 9152), (12, 'not', 8462), (13, 'me', 7758), (14, 'it', 7752), (15, 'for', 7584), (16, 'with', 7148), (17, 'be', 6852), (18, 'your', 6755), (19, 'this', 6608),
 (20, 'his', 6535)]
```

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.**

Frequency	Word Count	Examples
1.	11503	marl, purgative, darting
2.	3773	leven, overplus, ensconce
3.	2031	presageth, mountebank
4.	1374	lapwing, observed
5.	978	gord, engaged, dreaded
6.	778	redeemd, outrageous, unusual
7.	571	via, advocate, clothe
8.	462	Helenus, orsinos, exact
9.	391	te, wretchedness, lily
10.	321	relate, antique, channel

```
The bottom words are:
[(10, 321, ['othellos', 'birnam', 'goneril']), (9, 391, ['cassios', 'dunsinane', 'fleance']), (8, 462, ['servilius', 'cauldron', 'glamis']), (7, 571, ['chiron', 'mutius',
 'flaminius']), (6, 778, ['capulets', 'benvolio', 'rhodes']), (5, 978, ['saturninus', 'mercutios', 'montano']), (4, 1374, ['rapine', 'milius', 'goth']), (3, 2031, ['pallas',
 'stumps', 'andronici']), (2, 3773, ['virginius', 'weke', 'armoury']), (1, 11503, ['devoid', 'beastlike', 'relieves'])]
Time taken:  0.4117398262023926
```

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.	my lord	1685
3.	I have	1628
4.	in the	1584
5.	i will	1582
6.	to the	1518
7.	of the	1380
8.	it is	1087
9.	to be	968
10.	that i	935
11.	I do	829
12.	and I	736
13.	and he	728
14.	you are	724
15.	of my	696
16.	is the	692
17.	I would	674
18.	the king	664
19.	he is	658
20.	you have	652

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.

```
 B1
Probability of word "the" is:  0.033035065077269644
Probability of word "become" is:  0.00017716470042556436
Probability of word "brave" is:  0.00019315873588065006
Probability of word "treason" is:  0.00011318855860522168
```

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)]

```
 B2
Probability of "court | the" is:  0.0001316432148995513
Probability of "word | his" is:  2.2145587553195545e-05
Probability of "qualities | rare" is:  1.230310419621975e-06
Probability of "men | young" is:  1.1072793776597773e-05
```

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) Hint à use the chain rule (multiplication rule):

```
Probability of "have, sent" is:  2.702343854723852e-07
Probability of "will, look, upon"is:  6.532667708756251e-12
Probability of "I, am, no , baby " is:  4.021212806084287e-15
Probability of "wherefore, art, thou, romeo" is:  2.8662635320974777e-19
```

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.

```
Probability of "have, sent" if independent is:  2.4140938435533083e-06
Probability of "will, look, upon" if independent is:  1.3296268012755375e-08
Probability of "I, am, no , baby " if independent is:  5.757653307694146e-12
Probability of "wherefore, art, thou, romeo" if independent is:  1.7534240787231927e-13
```

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

(a) I am no (b) wherefore art thou

```
The most probable word to follow "I am no" is:  more
Prediction made in:  0.0377655029296875

The most probable word to follow "wherefore art thou" is:  romeo
Prediction made in:  0.03798365592956543
```

Conclusion:

Thus text processing, an important part of NLP was performed successfully using python.