Tanay Jhawar

BE4 Roll No:22

EXPERIMENT 6

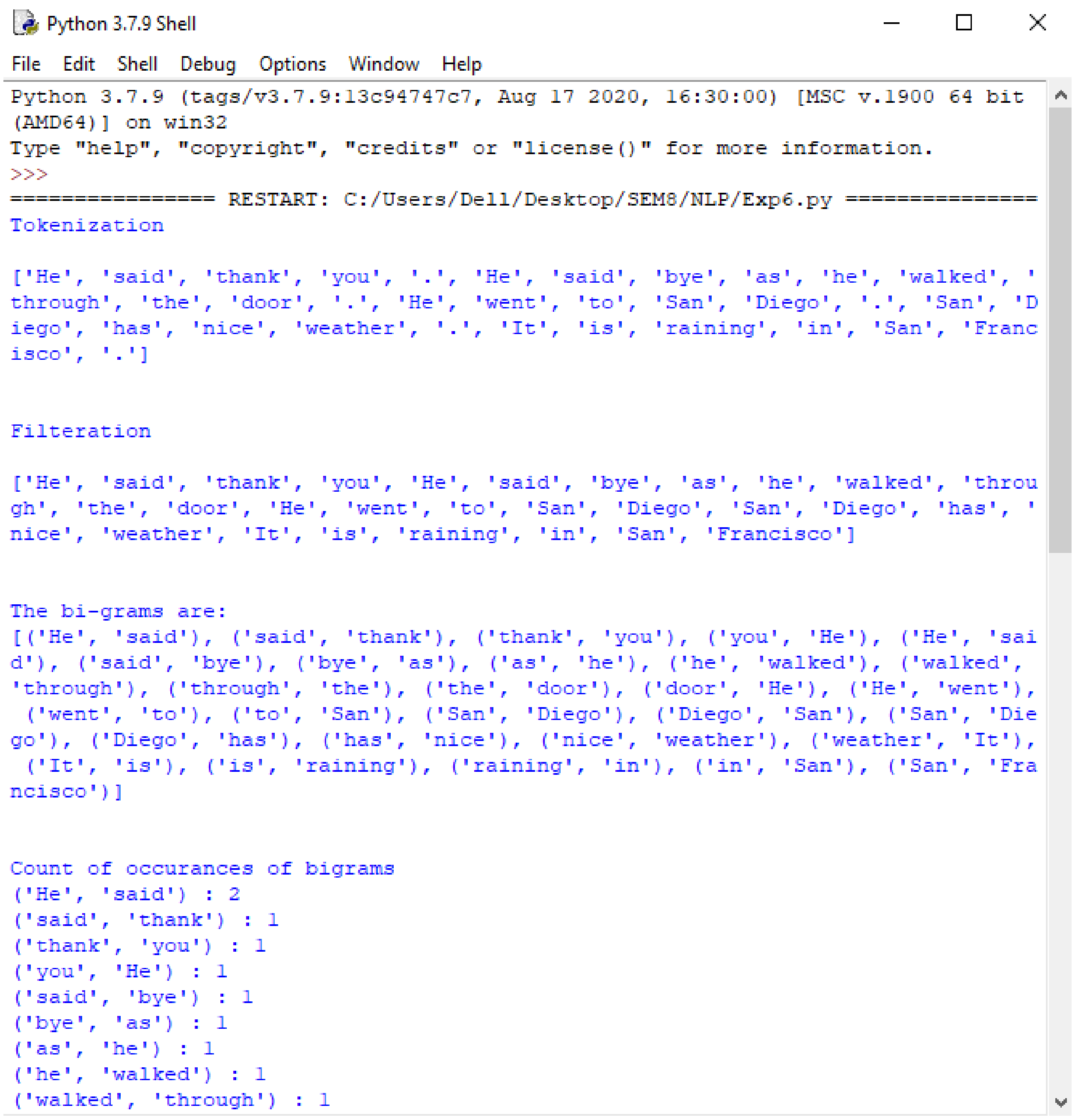
AIM: N-gram model in python to predict sentence probability.

SOURCE CODE:

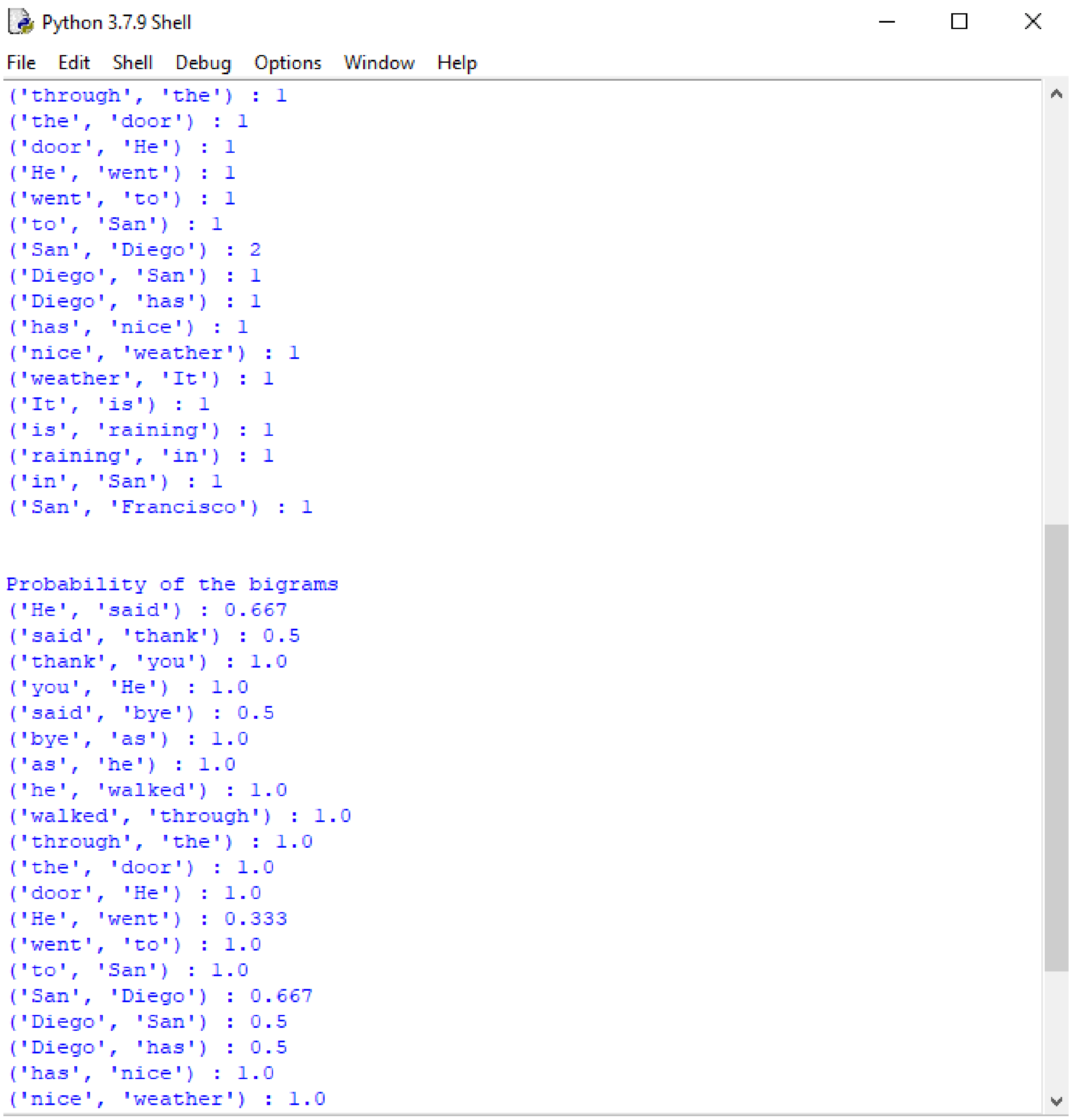
```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from collections import Counter


f = open("exp 6.txt","r")
for line in f:
    l1=[] #Count of bigrams
    l2=[] #First word of bigram
    l3=[] #Probability of bigram
    l4=[] #tuples of bigrams
    prod = 1
#wordtokenizer
    print("Tokenization with NLTK \n")
    print(word_tokenize(line))
    print("\n")


#Filteration
    print("Filteration \n")
    bad_chars = [';',',','!',':','*','#','<','>','?','@','.']
    words = word_tokenize(line)
    swords = []
    swords = list(filter(lambda i: i not in bad_chars,words))
    print(swords)
    print("\n")
    pair_words = []

#Bigram generation
    for i in range(len(swords)-1):
        pair_words.append((swords[i],swords[i+1]))
    print("The bi-grams are:")
    print(pair_words)
    print("\n")

    cnt = dict(Counter(pair_words))
    print("Count of occurances of bigrams")
    for pair, number in cnt.items():
        l1.append(number)
        l2.append(pair[0])
        l4.append(pair)
        print(pair, ":", number)
    print("\n")

    print("Probability of the bigrams")
    for x in range(len(l2)):
        p = swords.count(l2[x])
        l3.append(round(l1[x]/p,3))
        prod = l3[x]*prod
        print(l4[x], ":", l3[x])
     print("\n")

    print("Probability of the sentence.")
    print(prod)
```

INPUT TEXT: He said thank you. He said bye as he walked through the door. He went to San Diego. San Diego has nice weather. It is raining in San Francisco.

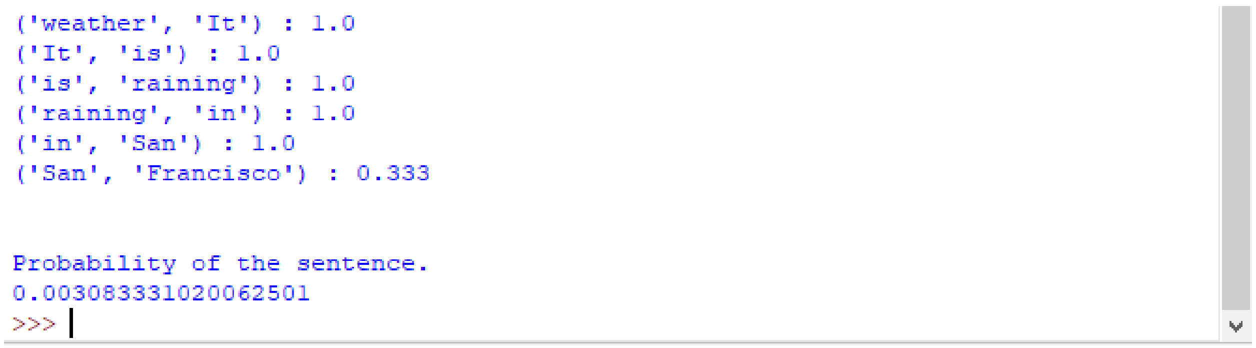
OUTPUT:



```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
================ RESTART: C:/Users/Dell/Desktop/SEM8/NLP/Exp6.py ===============
Tokenization

['He', 'said', 'thank', 'you', '.', 'He', 'said', 'bye', 'as', 'he', 'walked', '
through', 'the', 'door', '.', 'He', 'went', 'to', 'San', 'Diego', '.', 'San', 'D
iego', 'has', 'nice', 'weather', '.', 'It', 'is', 'raining', 'in', 'San', 'Franc
isco', '.']


Filteration

['He', 'said', 'thank', 'you', 'He', 'said', 'bye', 'as', 'he', 'walked', 'throu
gh', 'the', 'door', 'He', 'went', 'to', 'San', 'Diego', 'San', 'Diego', 'has', '
nice', 'weather', 'It', 'is', 'raining', 'in', 'San', 'Francisco']


The bi-grams are:
[('He', 'said'), ('said', 'thank'), ('thank', 'you'), ('you', 'He'), ('He', 'sai
d'), ('said', 'bye'), ('bye', 'as'), ('as', 'he'), ('he', 'walked'), ('walked',
'through'), ('through', 'the'), ('the', 'door'), ('door', 'He'), ('He', 'went'),
 ('went', 'to'), ('to', 'San'), ('San', 'Diego'), ('Diego', 'San'), ('San', 'Die
go'), ('Diego', 'has'), ('has', 'nice'), ('nice', 'weather'), ('weather', 'It'),
 ('It', 'is'), ('is', 'raining'), ('raining', 'in'), ('in', 'San'), ('San', 'Fra
ncisco')]


Count of occurances of bigrams
('He', 'said') : 2
('said', 'thank') : 1
('thank', 'you') : 1
('you', 'He') : 1
('said', 'bye') : 1
('bye', 'as') : 1
('as', 'he') : 1
('he', 'walked') : 1
('walked', 'through') : 1
```



```
('through', 'the') : 1
('the', 'door') : 1
('door', 'He') : 1
('He', 'went') : 1
('went', 'to') : 1
('to', 'San') : 1
('San', 'Diego') : 2
('Diego', 'San') : 1
('Diego', 'has') : 1
('has', 'nice') : 1
('nice', 'weather') : 1
('weather', 'It') : 1
('It', 'is') : 1
('is', 'raining') : 1
('raining', 'in') : 1
('in', 'San') : 1
('San', 'Francisco') : 1


Probability of the bigrams
('He', 'said') : 0.667
('said', 'thank') : 0.5
('thank', 'you') : 1.0
('you', 'He') : 1.0
('said', 'bye') : 0.5
('bye', 'as') : 1.0
('as', 'he') : 1.0
('he', 'walked') : 1.0
('walked', 'through') : 1.0
('through', 'the') : 1.0
('the', 'door') : 1.0
('door', 'He') : 1.0
('He', 'went') : 0.333
('went', 'to') : 1.0
('to', 'San') : 1.0
('San', 'Diego') : 0.667
('Diego', 'San') : 0.5
('Diego', 'has') : 0.5
('has', 'nice') : 1.0
('nice', 'weather') : 1.0
```



```
('weather', 'It') : 1.0
('It', 'is') : 1.0
('is', 'raining') : 1.0
('raining', 'in') : 1.0
('in', 'San') : 1.0
('San', 'Francisco') : 0.333


Probability of the sentence.
0.0030833310200625O1
>>> 
```