In [1]:

```
import collections
import functools

def word_frequency(input_string, print_dict =False, prob =True):
    word_frequency_list=[]
    word_count_dict = dict(collections.Counter(input_string.split()))
    if prob:
        word_count_dict = {k: v / total for total in (sum(word_count_dict.valu
es()),) for k, v in word_count_dict.items()}

    if print_dict:
        for key, value in word_count_dict.items():
            print('The word',key,'frequency is:', value)
    return word_count_dict

def unigram_scorer(input_string,prob_dict):
    p_sentence =1
    for word in input_string.split(' '):
        p_sentence = p_sentence *prob_dict[word]
    print ('probability of unigram model: ',p_sentence)

#main
input_string = 'the cat sat on the mat with a cat'
prob_dict =    word_frequency(input_string, True)
```

```
The word the frequency is: 0.2222222222222222
The word cat frequency is: 0.2222222222222222
The word sat frequency is: 0.1111111111111111
The word on frequency is: 0.1111111111111111
The word mat frequency is: 0.1111111111111111
The word with frequency is: 0.1111111111111111
The word a frequency is: 0.1111111111111111
```

In [2]:

```
unigram_scorer('the cat sat on the mat with a cat', prob_dict)
```

```
probability of unigram model:  4.129879666741113e-08
```

In [3]:

```
def read_data(data):
    dat=[]
    for i in range(len(data)):
        for word in data[i].split():
            dat.append(word)
    print(dat)
    return dat

def create_bigram(data):
    listOfBigrams = []
    bigramCounts = {}
    unigramCounts = {}
    for i in range(len(data)-1):

        if i < len(data) - 1 and data[i+1].islower():

            listOfBigrams.append((data[i], data[i + 1]))

            if (data[i], data[i+1]) in bigramCounts:
                bigramCounts[(data[i], data[i + 1])] += 1
            else:
                bigramCounts[(data[i], data[i + 1])] = 1

        if data[i] in unigramCounts:
            unigramCounts[data[i]] += 1
        else:
            unigramCounts[data[i]] = 1
    return listOfBigrams, unigramCounts, bigramCounts

def bigram_probability(listOfBigrams, unigramCounts, bigramCounts):
    listOfProb = {}
    for bigram in listOfBigrams:
        word1 = bigram[0]
        word2 = bigram[1]
        listOfProb[bigram] = (bigramCounts.get(bigram))/(unigramCounts.get(wor
d1))
    return listOfProb

def bigram_scorer(inputList):
    splt=inputList.split()
    outputProb1 = 1
    bilist=[]
    bigrm=[]

    for i in range(len(splt) - 1):
        if i < len(splt) - 1:
            bilist.append((splt[i], splt[i + 1]))

    print("\n The bigrams in given sentence are ")
    print(bilist)
    for i in range(len(bilist)):
        if bilist[i] in bigramProb:
            outputProb1 *= bigramProb[bilist[i]]
        else:
            outputProb1 *= 0
    print('\n' + 'Probablility of sentence \"'+input_string+'\" = ' + str(outp
utProb1))
```

In [4]:

```
data= ['<s> a cat sat on the mat </s>']
#data= ['This is a  dog','This is a cat','I love my cat','This is my name ']
```

In [5]:

```
if __name__ == '__main__':
    data = read_data(data)
    listOfBigrams, unigramCounts, bigramCounts = create_bigram(data)

    print("\n All the possible Bigrams are ")
    print(listOfBigrams)

    print("\n Bigrams along with their frequency ")
    print(bigramCounts)

    print("\n Unigrams along with their frequency ")
    print(unigramCounts)

    bigramProb = bigram_probability(listOfBigrams, unigramCounts, bigramCounts
)

    print("\n Bigrams along with their probability ")
    print(bigramProb)
    #####################
    ########Test1############
    #####################
    input_string="<s> a cat sat on the mat </s>"
    bigram_scorer(input_string)
    #####################
    ########Test2############
    #####################
    input_string="<s> a cat sat on the car </s>"
    bigram_scorer(input_string)
```

```
['<s>', 'a', 'cat', 'sat', 'on', 'the', 'mat', '</s>']

 All the possible Bigrams are
[('<s>', 'a'), ('a', 'cat'), ('cat', 'sat'), ('sat', 'on'), ('on', 'the'),
('the', 'mat'), ('mat', '</s>')]

 Bigrams along with their frequency
{('<s>', 'a'): 1, ('a', 'cat'): 1, ('cat', 'sat'): 1, ('sat', 'on'): 1, ('o
n', 'the'): 1, ('the', 'mat'): 1, ('mat', '</s>'): 1}

 Unigrams along with their frequency
{'<s>': 1, 'a': 1, 'cat': 1, 'sat': 1, 'on': 1, 'the': 1, 'mat': 1}

 Bigrams along with their probability
{('<s>', 'a'): 1.0, ('a', 'cat'): 1.0, ('cat', 'sat'): 1.0, ('sat', 'on'): 1.
0, ('on', 'the'): 1.0, ('the', 'mat'): 1.0, ('mat', '</s>'): 1.0}

 The bigrams in given sentence are
[('<s>', 'a'), ('a', 'cat'), ('cat', 'sat'), ('sat', 'on'), ('on', 'the'),
('the', 'mat'), ('mat', '</s>')]

Probablility of sentence "<s> a cat sat on the mat </s>" = 1.0

 The bigrams in given sentence are
[('<s>', 'a'), ('a', 'cat'), ('cat', 'sat'), ('sat', 'on'), ('on', 'the'),
('the', 'car'), ('car', '</s>')]

Probablility of sentence "<s> a cat sat on the car </s>" = 0.0
```

In [6]:

```python
def bigram_scorer_with_smoothing(inputList):
    splt=inputList.split()
    outputProb1 = 1
    bilist=[]
    bigrm=[]

    for i in range(len(splt) - 1):
        if i < len(splt) - 1:

            bilist.append((splt[i], splt[i + 1]))

    print("\n The bigrams in given sentence are ")
    print(bilist)
    for i in range(len(bilist)):
        if bilist[i] in bigramProb:
            outputProb1 *= bigramProb[bilist[i]]
        elif bilist[i][0] in unigramCounts:
            outputProb1 *= 1/ (unigramCounts[bilist[i][0]] + len(unigramCounts
))
        else:
            outputProb1 *= 1/  len(unigramCounts)
    print('\n' + 'Probablility of sentence \"'+input_string+'\" = ' + str(outp
utProb1))
```

In [7]:

```
    #####################
    ########Test2#############
    #####################
    input_string="<s> This is my car </s>"
    bigram_scorer_with_smoothing(input_string)
```

```
 The bigrams in given sentence are
[('<s>', 'This'), ('This', 'is'), ('is', 'my'), ('my', 'car'), ('car', '</s
>')]

Probablility of sentence "<s> This is my car </s>" = 5.206164098292377e-05
```

In [ ]: