import json

config = json.loads(open('C:/Users/Administrator/Desktop/jsonex.json').read())

l=config[0]

l['reviewText']

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

from nltk.tokenize import sent\_tokenize, word\_tokenize

example\_sent =l['reviewText']

from nltk.corpus import stopwords

stop\_words = set(stopwords.words('english'))

word\_tokens = word\_tokenize(example\_sent)

filtered\_sentence = [w for w in word\_tokens if not w in stop\_words]

filtered\_sentence = []

for w in word\_tokens:

if w not in stop\_words:

filtered\_sentence.append(w)

from nltk.sentiment.vader import SentimentIntensityAnalyzer

test\_subset=filtered\_sentence

sid = SentimentIntensityAnalyzer()

pos\_word\_list=[]

neu\_word\_list=[]

neg\_word\_list=[]

for word in test\_subset:

if (sid.polarity\_scores(word)['compound']) >= 0.5:

pos\_word\_list.append(word)

elif (sid.polarity\_scores(word)['compound']) <= -0.5:

neg\_word\_list.append(word)

else:

neu\_word\_list.append(word)

from collections import Counter

Counter = Counter(neu\_word\_list)

most\_occur = Counter.most\_common()

lis = []

lis2 = []

length=len(most\_occur)

length

for i in range(0,length):

p=most\_occur[i]

lis.append(p[1])

for i in range(0,length):

p=most\_occur[i]

lis2.append(p[0])

from collections import Counter

count = Counter(pos\_word\_list)

most\_occur1 = count.most\_common()

from collections import Counter

coun = Counter(neg\_word\_list)

most\_occur2 = coun.most\_common()

import matplotlib.pyplot as plt; plt.rcdefaults()

import numpy as np

import matplotlib.pyplot as plt

objects = tuple(lis2)#lis2 words

y\_pos = np.arange(len(objects))

performance = lis#lis values

plt.bar(y\_pos, performance, align='center', alpha=0.5)

plt.xticks(y\_pos, objects)

plt.xlabel('Usage')

plt.title('words count')

plt.show()

lis3 = []

lis4 = []

length=len(most\_occur1)

length

for i in range(0,length):

q=most\_occur1[i]

lis3.append(q[1])

for i in range(0,length):

q=most\_occur1[i]

lis4.append(q[0])

import matplotlib.pyplot as plt; plt.rcdefaults()

import numpy as np

import matplotlib.pyplot as plt

objects = tuple(lis4)#lis4 words

y\_pos = np.arange(len(objects))

performance = lis3#lis3 values

plt.bar(y\_pos, performance, align='center', alpha=0.5)

plt.xticks(y\_pos, objects)

plt.xlabel('Usage')

plt.title('words count')

plt.show()

lis5 = []

lis6 = []

length=len(most\_occur2)

length

for i in range(0,length):

q=most\_occur2[i]

lis5.append(q[1])

for i in range(0,length):

q=most\_occur2[i]

lis6.append(q[0])

import matplotlib.pyplot as plt; plt.rcdefaults()

import numpy as np

import matplotlib.pyplot as plt

objects = tuple(lis6)#lis4 words

y\_pos = np.arange(len(objects))

performance = lis5#lis3 values

plt.bar(y\_pos, performance, align='center', alpha=0.5)

plt.xticks(y\_pos, objects)

plt.xlabel('Usage')

plt.title('words count')

plt.show()