LSTM on Amazon fine food review

In [1]:
```
import sqlite3
import pandas as pd
import numpy as np
import re
import string
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import StandardScaler
from sklearn.manifold import TSNE
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer as sno
setofstopwords=set(stopwords.words('english'))
nltk.download('stopwords')
```

```
[nltk_data] Error loading stopwords: <urlopen error [Errno 11001]
[nltk_data]     getaddrinfo failed>
```

Out[1]:
```
False
```

Loading from database

In [2]:
```
conn= sqlite3.connect('database.sqlite')
data= pd.read_sql_query('''
SELECT * FROM Reviews WHERE Score!=3
''',conn)
data.shape
```

Out[2]:
```
(525814, 10)
```

Removing not helpful reviews

In [3]:
```
data['Score']=data['Score'].map(lambda x:'Positive' if x>3 else 'Negati
ve')
sorteddata= data.sort_values('ProductId',axis=0)
finaldata= sorteddata.drop_duplicates(subset={'UserId','ProfileName',\
        'Time','Text'}, keep='first',inplace=False)

finaldata= finaldata[finaldata['HelpfulnessNumerator'] <= finaldata['He
lpfulnessDenominator']]
data= finaldata.sort_values('Time',axis=0)
data.shape
```

Out[3]:
```
(364171, 10)
```

Cleaning HTML, punctuations, apply stemming, lowercasing etc without removing stopwords

In [4]:
```
def cleanhtml(sentance): #substitute expression contained in <> with '
 '
    cleaned= re.sub(re.compile('<.*?>'),' ',sentance)
    return cleaned
#function for removing punctuations chars
def cleanpunc(sentance):
    cleaned= re.sub(r'[?|!|\'|"|#]',r'',sentance)
    cleaned= re.sub(r'[.|,|)|(|\|/]',r'',sentance)
    return cleaned
snowstem= sno('english')

i=0
str1=' '
final_string=[]
all_positive_words=[] # store words from +ve reviews here
all_negative_words=[] # store words from -ve reviews here.
for sent in data['Text'].values:
    filtered_sentence=[]
```

```
    #print(sent);
    sent=cleanhtml(sent) # remove HTml tags
    for w in sent.split():
        # we have used cleanpunc(w).split(), one more split function he
re
        # because consider w="abc.def", cleanpunc(w) will return "abc d
ef"
        # if we dont use .split() function then we will be considring
 "abc def"
        # as a single word, but if you use .split() function we will ge
t "abc", "def"
        for cleaned_words in cleanpunc(w).split():
            if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
                s=(snowstem.stem(cleaned_words.lower())).encode('utf8')
                filtered_sentence.append(s)
                if(data['Score'].values)[i] =='Positive':
                    all_positive_words.append(s)
                if(data['Score'].values)[i] =='Negative':
                    all_negative_words.append(s)
            else:
                continue
    str1 = b" ".join(filtered_sentence) #final string of cleaned words
    final_string.append(str1)

# storing data till now
data['CleanedText']=final_string
#adding a column of CleanedText which displays the data after pre-proce
ssing of the review
data['CleanedText']=data['CleanedText'].str.decode("utf-8")
    # store final table into an SQlLite table for future.
conn = sqlite3.connect('cleanedTextData.sqlite')
c=conn.cursor()
conn.text_factory = str
data.to_sql('Reviews', conn,  schema=None, if_exists='replace', \
        index=True, index_label=None, chunksize=None, dtype=None)
conn.close()
```

In [5]:
```
data.head()
```

Out[5]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	He
138706	150524	0006641040	ACITT7DI6IDDL	shari zychinski	0	0
138683	150501	0006641040	AJ46FKXOVC7NR	Nicholas A Mesiano	2	2
417839	451856	B00004CXX9	AIUWLEQ1ADEG5	Elizabeth Medina	0	0
346055	374359	B00004CI84	A344SMIA5JECGM	Vincent P. Ross	1	2
417838	451855	B00004CXX9	AJH6LUC1UT1ON	The Phantom of the Opera	0	0

In [6]:
```
data['Text'][2]
```

Out[6]:
```
'This is a confection that has been around a few centuries.  It is a li
ght, pillowy citrus gelatin with nuts - in this case Filberts. And it i
```

```
s cut into tiny squares and then liberally coated with powdered sugar.
And it is a tiny mouthful of heaven.  Not too chewy, and very flavorfu
l.  I highly recommend this yummy treat.  If you are familiar with the
story of C.S. Lewis\' "The Lion, The Witch, and The Wardrobe" - this is
the treat that seduces Edmund into selling out his Brother and Sisters
to the Witch.'
```

In [7]:
```
data['CleanedText'][2]
```

Out[7]:
```
'this confect that has been around few centuri light pillowi citrus gel
atin with nut this case filbert and cut into tini squar and then liber
coat with powder sugar and tini mouth heaven not too chewi and veri fla
vor high recommend this yummi treat you are familiar with the stori lio
n the witch and the this the treat that seduc edmund into sell out his
brother and sister the witch'
```

Taking 100k datapoints

In [8]:
```
Data= data[:100000]
Data= Data[['CleanedText','Score']]
Data['Score']= Data['Score'].map(lambda x:1 if x=='Positive' else 0)

Data_x= Data['CleanedText']
Data_y= Data['Score']
```

In [9]:
```
Data_x.index= [i for i in range(0,10**5)]
Data_x
```

Out[9]:
```
0         this witti littl book make son laugh loud reci...
1         can rememb see the show when air televis year ...
2         beetlejuic well written movi everyth about fro...
3         twist rumplestiskin captur film star michael k...
4         beetlejuic excel and funni movi keaton hilari ...
5         this one movi that should your movi collect fi...
6         myself alway enjoy this movi veri funni and en...
7         bought few these after apart was infest with f...
8         what happen when you say his name three michae...
```

9	get look for beatlejuic french version video r...
10	get crazi realli imposs today not find the fre...
11	this was realli good idea and the final produc...
12	just receiv shipment and could hard wait tri t...
13	have just recent purchas the woodstream corp g...
14	this are much easier use than the wilson past ...
15	these are easi use they not make mess and offe...
16	this such great film even know how sum first a...
17	beetlejuic wonder amus comed romp that explor ...
18	sick scad nasti toothpick all over counter whe...
19	thought this movi was funni michael keaton bee...
20	mani movi have dealt with the figur death and ...
21	know whi anyon would ever use those littl liqu...
22	michael keaton bring distinguish characterist ...
23	continu amaz the shoddi treatment that some mo...
24	just warn you when tri trick you the widescree...
25	bought these decor some dia los muerto skull w...
26	winona ryder the gothic princess doom see for ...
27	this was favorit book mine when was littl girl...
28	for year have been tri simul truli italian esp...
29	when vacat adam and barbara maitland meet thei...
	...
99970	like strong black coffe like tast strong have ...
99971	absolut love this tea drink tea everyday and h...
99972	about year age our cat would have bladder infe...
99973	bought the turkish delight and then read the r...
99974	husband recent ate outsid cafe geneva was late...
99975	was suffer from cold for almost year spent lot...
99976	neighbor and love this candi and difficult fin...
99977	dog absolut love yummi chummi use for train tr...
99978	this second purchas and plan stay stock with t...
99979	was excit when found whole wheat isra couscous...
99980	bought dozen these monkey lollipop parti favor...
99981	order three set these from entirelypet even ca...
99982	special protein plus far favorit cereal the on...
99983	was given the opportun review this product bel...
99984	gloria jean hit out the park this whi not ther...
99985	yuck this coffe tast terribl doe not compet th...
99986	order this item becaus when purchas from local...

```
99987    compar other nutrit bar out there find these a...
99988    love peanut butter love that they contain ton ...
99989    two young adult siberian huski love these they...
99990    love teaand realli hate write bad review mayb ...
99991    husband and were alway afraid make fish like m...
99992    ice breaker ice cube peppermint sugar free gum...
99993    say that name this coffe blend appropri found ...
99994    was look for coffe replac and this fit the bil...
99995    this veri tasti protein shake and give you nic...
99996    this the best tast oliv and healthi for you to...
99997    sack melitta coffe label fine grind blanc noir...
99998    final babi food that tast love that great prot...
99999    first ventur salt and happi bought this and sa...
Name: CleanedText, Length: 100000, dtype: object
```

Making Vocabulary set and Frequency dictionary of words

In [10]:
```
# collecting all words in single list
list_= []
for i in Data_x:
    list_ += i
list_= ''.join(list_)
allWords=list_.split()
```

In [11]:
```
vocabulary= set(allWords)
```

In [12]:
```
vocabulary_list= list(vocabulary)
```

In [13]:
```
#frequency dictionary
freq_dict= {}
for word in vocabulary_list:
    freq_dict[word]= allWords.count(word)
```

In [14]:
```
freq_dict
```

Out[14]:
```
{'mouthhonest': 1,
```

```
'gardenhad': 1,
'almondthe': 2,
'teas': 20,
'thoughfour': 1,
'shortmi': 2,
'snackwith': 3,
'someuntil': 1,
'grcoeri': 1,
'shotgreat': 1,
'support': 351,
'companiorder': 4,
'caesar': 48,
'itemwonder': 1,
'secondthe': 4,
'caloriecount': 1,
'cruchier': 1,
'chump': 1,
'ozbut': 1,
'malto': 4,
'balconi': 4,
'enjoyreceiv': 1,
'themalsobest': 1,
'yearwhi': 1,
'treatlemon': 1,
'choicget': 2,
'findrecent': 2,
'unveil': 1,
'satisfiedit': 1,
'sampltwo': 1,
'superh': 1,
'alkali': 30,
'everialthough': 1,
'impressionsgood': 1,
'coquett': 1,
'stink': 122,
'caloribig': 1,
'alf': 1,
'tamaratasti': 1,
'stuffer': 57,
```

```
'bananafirst': 4,
'sac': 2,
'lqqk': 1,
'miss': 1141,
'enjoyhomemad': 1,
'mexican': 337,
'rip': 253,
'negoti': 6,
'meatfishpotato': 1,
'minmost': 1,
'greatwork': 3,
'breakout': 13,
'spreadabl': 34,
'quickfianc': 1,
'halvqualiti': 1,
'greatunbeliev': 1,
'andson': 3,
'benifici': 1,
'swissgold': 2,
'steadilong': 1,
'oppositei': 1,
'corprat': 1,
'economiactual': 1,
'tastesh': 1,
'gratztwo': 1,
'lemonlead': 1,
'bottlhigh': 1,
'alrightthis': 1,
'thesenot': 4,
'disappointher': 1,
'randal': 3,
'advertiswhen': 1,
'wakam': 28,
'peaceful': 2,
'snackbabi': 1,
'parka': 1,
'nar': 1,
'eatthis': 36,
'inprov': 3,
```

```
'teethanyhow': 1,
'tryhowev': 1,
'enjoymonth': 1,
'indianapoli': 5,
'grap': 3,
'wedgewood': 1,
'savewhi': 1,
'pleasafter': 2,
'yassou': 1,
'muc': 1,
'itqual': 1,
'highnice': 1,
'calorilet': 1,
'wholefood': 27,
'powerscoop': 2,
'highboy': 1,
'rst': 1,
'dicalcium': 5,
'numberhave': 1,
'juicberri': 1,
'chopperthere': 1,
'shipper': 52,
'thespoon': 1,
'mba': 1,
'teadelici': 3,
'diaz': 1,
'gumnoth': 1,
'senticosus': 1,
'claimbest': 1,
'gobbl': 286,
'hisherthese': 1,
'zoo': 9,
'heightthese': 1,
'themokay': 2,
'nsa': 2,
'grandkidthis': 1,
'tisan': 3,
'vodkanot': 1,
'amazoncarbsmart': 1,
```

```
'morelive': 2,
'spreadthese': 1,
'eitherlove': 4,
'saidwonder': 1,
'bottom': 1068,
'delicipaid': 1,
'servicbefor': 2,
'ase': 1,
'mustnot': 1,
'availsister': 1,
'jasminwhen': 1,
'selectfinger': 1,
'intactthought': 1,
'orderlive': 1,
'topnot': 2,
'gross': 271,
'tulear': 4,
'sickthi': 1,
'forobsess': 1,
'chiliwith': 1,
'conditthought': 1,
'taffi': 94,
'firegot': 1,
'priceflour': 1,
'flavornescaf': 1,
'amzon': 4,
'sweedifsh': 1,
'epoiss': 3,
'lifestyljust': 1,
'figblueberri': 1,
'pizzathese': 1,
'athlet': 53,
'perier': 1,
'wrongcalori': 1,
'tribig': 3,
'sellerlong': 1,
'alonwas': 1,
'somethis': 17,
'favrit': 1,
```

```
'peachose': 1,
'caeser': 2,
'goodlif': 4,
'stongerlove': 1,
'valuother': 1,
'wrongthe': 4,
'queliti': 1,
'conditwork': 1,
'gubment': 1,
'housalthough': 1,
'satisfiafter': 1,
'kenosha': 1,
'missjust': 1,
'vino': 20,
'stickbeen': 1,
'beanrice': 1,
'basestummbl': 1,
'trampl': 1,
'sellthese': 2,
'easnot': 1,
'extrathis': 7,
'agrehave': 1,
'choosen': 1,
'notdog': 2,
'gulpeven': 1,
'toucheven': 1,
'blandeven': 1,
'regretwas': 1,
'fewsteaz': 1,
'toronto': 14,
'productasid': 1,
'shiparriv': 1,
'sojournseek': 1,
'vineother': 1,
'armageddon': 1,
'ullstrup': 1,
'havestumbl': 1,
'akita': 13,
'needmake': 1,
```

```
'rerout': 2,
'bonesent': 1,
'heresmooth': 1,
'recommendtasti': 1,
'cocamidopropyl': 1,
'englishuse': 1,
'servgot': 1,
'bucketoften': 1,
'bananabeen': 1,
'muchqualiti': 1,
'miricl': 2,
'gojust': 2,
'boxesthrough': 1,
'bled': 1,
'buta': 2,
'benwick': 1,
'livedoe': 1,
'largthese': 1,
'sweetthes': 2,
'quicksyrup': 1,
'whale': 8,
'lumpusual': 1,
'mouthwas': 2,
'whicih': 2,
'heroin': 9,
'seanson': 1,
'deliverionli': 1,
'accountit': 1,
'heathand': 1,
'choicthese': 4,
'nutritiquot': 1,
'improb': 2,
'wouldwil': 1,
'wondertahitian': 1,
'timesnot': 1,
'autopuchas': 1,
'maywant': 1,
'sparx': 2,
'informnot': 1,
```

```
'cleanrememb': 1,
'whofound': 1,
'sir': 16,
'chick': 51,
'moldthe': 1,
'nermal': 1,
'cannumi': 1,
'themsart': 1,
'nsaid': 5,
'aristocrat': 1,
'fuit': 4,
'reaquir': 1,
'beardavinci': 1,
'yumbeefeat': 1,
'contentbeen': 1,
'ami': 52,
'lavazza': 328,
'uselik': 1,
'againmedium': 2,
'ponti': 2,
'trisleepytim': 1,
'debt': 2,
'itsself': 1,
'simliar': 2,
'washigh': 1,
'headlamp': 1,
'momentthe': 1,
'coconutanis': 1,
'quaint': 8,
'lindtlove': 1,
'gourm': 1,
'thesedoe': 1,
'pekoeasi': 1,
'heruse': 2,
'daisi': 12,
'loveamazon': 1,
'coffeeand': 4,
'nutfirst': 2,
'amazoni': 7,
```

```
'veggievegan': 1,
'varietydo': 1,
'crispihave': 1,
'tastithis': 36,
'cliqu': 1,
'stabilis': 1,
'mixadd': 1,
'thesemedic': 1,
'breakfastbrunchlunch': 1,
'clubbig': 1,
'mellow': 347,
'makenatur': 1,
'batchthese': 1,
'diefor': 1,
'evendelici': 1,
'faultless': 1,
'pricehas': 1,
'stroner': 1,
'afterlove': 1,
'alimoni': 1,
'lovesprinkelz': 1,
'pizzafrench': 1,
'justoff': 1,
'roastercoffe': 1,
'bitechew': 1,
'naturesway': 1,
'nowvet': 1,
'nominbought': 1,
'starlook': 1,
'knowsee': 1,
'coronado': 1,
'waylight': 2,
'snackhad': 3,
'jarlike': 1,
'indeokay': 1,
'litterhave': 1,
'choindroitin': 1,
'rought': 1,
'nasturtium': 1,
```

```
'glinda': 1,
'promptlike': 1,
'cheesefor': 1,
'couldsometh': 1,
'lookingsmel': 1,
'eattook': 1,
'cupboardhealthi': 1,
'togethexcel': 1,
'allessi': 2,
'minutknow': 1,
'mozzarello': 1,
'obel': 1,
'involvpurchas': 1,
'sultri': 4,
'localnot': 1,
'vis': 2,
'guylook': 1,
'nuthow': 1,
'allergihello': 1,
'lotion': 55,
'digest': 902,
'nectari': 1,
'contributinglead': 2,
'brook': 18,
'servhave': 7,
'expensfirst': 1,
'waterbe': 1,
'houe': 1,
'goodaltern': 1,
'liaison': 2,
'sunup': 1,
'angosturaput': 1,
'alsonow': 1,
'avail': 3476,
'wonderlike': 4,
'replacdog': 1,
'freshwish': 1,
'fashionwhole': 1,
'downhate': 1,
```

```
'dissappointedi': 1,
'grim': 2,
'mukwa': 1,
'handwritten': 4,
'storg': 1,
'bettermad': 2,
'specialagre': 1,
'localfavourit': 1,
'fruitilike': 1,
'pennant': 3,
'signedkeisha': 1,
'bottomzevia': 1,
'riceand': 1,
'marchwhen': 1,
'cerealprobabl': 1,
'likemi': 1,
'suburban': 4,
'noodllove': 7,
'soruc': 2,
'youtaylor': 1,
'twicethis': 1,
'seseam': 1,
'grand': 113,
'callmyself': 1,
'stinksi': 1,
'granolajust': 1,
'themwhol': 1,
'againexact': 2,
'replimuch': 1,
'prep': 96,
'typicalthey': 1,
'yikeamaz': 1,
'mosit': 2,
'sleepvalerian': 1,
'againour': 9,
'delicihad': 5,
'worldget': 1,
'sabdariffa': 1,
'wentbought': 1,
```

```
'newsimpli': 1,
'deez': 1,
'marcipan': 1,
'fwd': 1,
'breed': 193,
'redwas': 1,
'bettertast': 1,
'whith': 3,
'bivalvia': 1,
'lifestylexcel': 1,
'amazoncame': 1,
'togetherth': 2,
'mealit': 3,
'shopmix': 1,
'tastinot': 1,
'silkikid': 1,
'havefind': 1,
'accordaccord': 1,
'canwhen': 1,
'nextlike': 1,
'rider': 10,
'wsubscrib': 1,
'regretswil': 1,
'butwho': 2,
'allchocol': 1,
'continutawni': 1,
'localn': 1,
'finlandthis': 1,
'areare': 1,
'priceran': 1,
'findgreat': 1,
'whenevbought': 1,
'forneed': 2,
'welsh': 19,
'unfood': 1,
'everyontitl': 1,
'tamarin': 1,
'uniqui': 1,
'givewhile': 1,
```

```
'ibrik': 12,
'thunderstorm': 4,
'mealhave': 11,
'scenariolove': 1,
'organdi': 1,
'amazonarriv': 1,
'mornbecaus': 1,
'spotless': 3,
'hospituse': 1,
'thingsugh': 1,
'chemicalnoxiousphoni': 1,
'hearti': 359,
'bluewhit': 1,
'quithave': 1,
'told': 1116,
'jamthink': 1,
'husand': 1,
'visionari': 1,
'investdog': 1,
'ingredithe': 5,
'ritterhad': 1,
'attende': 4,
'spoonwill': 1,
'alfredobroccoli': 1,
'magicthis': 1,
'meowpppuuuuurrrrrrrrrrm': 1,
'carriglad': 1,
'crucifixion': 1,
'occurlove': 1,
'tastthey': 4,
'womderopinion': 1,
'marythis': 1,
'swallowthe': 1,
'comut': 1,
'tastrate': 1,
'entirpurchas': 1,
'beleiv': 15,
'dipit': 1,
'splitgood': 1,
```

```
'pitathis': 1,
'oppoertun': 1,
'nieghbor': 2,
'ther': 11,
'somethingit': 1,
'browniesal': 1,
'veriproduct': 2,
'coconuthusband': 1,
'rasp': 4,
'cazzo': 1,
'delicijust': 4,
'punchalway': 1,
'michell': 7,
'guessperhap': 1,
'boxsalti': 1,
'budbeen': 1,
'thewould': 1,
'gooierprocess': 1,
'eatthose': 1,
'wastokay': 1,
'chice': 1,
'beathave': 7,
'journeythis': 2,
'fierc': 8,
'therebest': 2,
'psychosomat': 2,
'allfrom': 1,
'compomis': 1,
'extraafter': 1,
'selectionsthi': 1,
'asept': 29,
'tastihodgson': 1,
'recommendbread': 1,
'juicesbroth': 1,
'frequentcall': 1,
'whitegreat': 1,
'relishlove': 1,
'bthey': 1,
'thisone': 5,
```

```
'housgrate': 1,
'cather': 1,
'feed': 1672,
'keuriggrew': 1,
'boywa': 1,
'freerestaur': 1,
'thelifeboat': 1,
'outskirt': 2,
'kidcannot': 1,
'wayfirst': 1,
'guessavenu': 1,
'useheat': 1,
'crockpot': 41,
'withread': 1,
'dewi': 2,
'ketchupnever': 1,
'bestcook': 1,
'enoughveri': 1,
'galaxi': 6,
'worthpasta': 1,
'overestim': 1,
'healthipleas': 1,
'aqua': 9,
'againfool': 1,
'whatta': 1,
'darljeel': 2,
'supermarkdet': 1,
'guestsnobodi': 1,
'betternow': 1,
'brew': 2850,
'definithave': 2,
'stuffnecess': 1,
'split': 265,
'taghave': 1,
'fivethe': 1,
'servcompar': 1,
'enjoyneed': 1,
'becauswill': 1,
'gawdaw': 1,
```

```
'cerelacseem': 1,
'real': 3963,
'puffthis': 1,
'klum': 1,
'forevpop': 1,
'array': 32,
'foodeveryon': 1,
'anathema': 1,
'floatwas': 1,
'americanfor': 1,
'movabl': 1,
'tupperwarfor': 1,
'fillingi': 2,
'sweetsomewhat': 1,
'thotri': 1,
'overrun': 7,
'capellini': 1,
'tangbeen': 1,
'burney': 1,
'likeconstant': 1,
'tribeliev': 1,
'soundwierd': 1,
'complain': 426,
'hallmark': 11,
'thatnot': 6,
'gon': 1,
'laterfirst': 1,
'onlinonli': 1,
'diffen': 1,
'insultlove': 1,
'brainerwhat': 1,
'pressescoffe': 1,
'rudolf': 1,
'hothusband': 1,
'cinnimoninever': 1,
'caningrew': 1,
'enjoycould': 1,
'gruijter': 1,
'youdiscov': 3,
```

```
'refreger': 1,
'cert': 4,
'canafter': 1,
'specialhad': 1,
'workwhole': 1,
'stalesoft': 1,
'alec': 47,
'usair': 1,
'wouldint': 1,
'western': 70,
'evok': 20,
'greatfunni': 1,
'outadam': 1,
'flare': 13,
'astorianot': 1,
'pleasmother': 1,
'pizzamother': 1,
'rees': 69,
'optionthe': 3,
'anythingmayb': 1,
'forgave': 2,
'blandexcept': 1,
'hast': 7,
'riblove': 1,
'biscuitlove': 1,
'colester': 3,
'rub': 390,
'economlove': 1,
'mangoamaz': 1,
'daywill': 1,
'unusuthese': 1,
'megreat': 1,
'becomwas': 1,
'suddenthese': 1,
'excelbuy': 1,
'letbought': 1,
'justif': 4,
'withoutbought': 1,
'hourwas': 1,
```

```
'fishybut': 1,
'herbgreat': 2,
'agreboth': 1,
'wyour': 2,
'han': 13,
'groundthis': 1,
'othereasi': 1,
'squeaki': 8,
'focuss': 1,
'tooyear': 1,
'againenjoy': 5,
'bewargreat': 1,
'knee': 52,
'dietlive': 1,
'chocolstart': 2,
'almosi': 1,
'teaplanter': 1,
'drugtea': 1,
'everlove': 7,
'tastnormal': 2,
'machinrealli': 2,
'lusciouslemoni': 1,
'buttercat': 1,
'cutom': 4,
'mixwowi': 1,
'milkthink': 1,
'economafter': 1,
'truffIthis': 3,
'iperespresso': 2,
'modeno': 3,
'krasner': 2,
'potent': 132,
'brag': 16,
'deign': 4,
'olio': 20,
'stuffbanana': 1,
'coffeshould': 1,
'nonperish': 2,
'vetrecent': 1,
```

```
'rooftop': 1,
'snickerdoodlwas': 1,
'gnossthese': 1,
'timehave': 39,
'amazinggg': 1,
'clin': 2,
'kathiveri': 1,
'falernum': 2,
'drinkcare': 1,
'pleasurfor': 1,
'eitherdog': 5,
'pomno': 1,
'wasonli': 1,
'fireilli': 1,
'hallbought': 1,
'meanlove': 1,
'bottlagre': 1,
'trimake': 2,
'epicuriouscom': 1,
'mri': 1,
'daymarmit': 1,
'stargloria': 1,
'koenigwife': 1,
'buyyet': 1,
'receivlike': 2,
'ooncentr': 1,
'musubi': 1,
'realizese': 1,
'canfor': 2,
'deliev': 7,
'chichen': 4,
'gravyi': 1,
'montego': 2,
'producteasi': 3,
'crabbi': 1,
'porkmak': 1,
'wastpinol': 1,
'daydead': 1,
'thereof': 11,
```

```
'eatenjust': 1,
'wasbuffalo': 1,
'surprissquirt': 1,
'craveare': 1,
'muesli': 197,
'sizehusband': 2,
'ovaltinwould': 1,
'ineedcoffe': 1,
'headachpuerh': 1,
'increas': 706,
'orihibit': 1,
'fruition': 34,
'skinstomach': 1,
'forusual': 2,
'coppicino': 1,
'blendwhol': 1,
'lattic': 1,
'scanhuge': 1,
'friedman': 1,
'pranw': 1,
'flush': 94,
'wks': 8,
'dfw': 1,
'wanttast': 1,
'timesinc': 1,
'thentoday': 1,
'legion': 2,
'appetitveri': 1,
'insteadfrom': 1,
'bulkyou': 3,
'recommendlittl': 3,
'weekthe': 8,
'beatlike': 3,
'havehuy': 1,
'zzzzzz': 1,
'onlinabsolut': 1,
'reorderthe': 1,
'advent': 10,
'anyonall': 1,
```

```
'dass': 1,
'grillingsmok': 1,
'acidophuilus': 2,
'itif': 6,
'amazonthe': 34,
'bellato': 1,
'pancaktri': 1,
'intest': 1,
'misomix': 1,
'sway': 10,
'didand': 1,
'herhard': 1,
'pricefamili': 7,
'poundcak': 3,
'carbolit': 1,
'nabob': 17,
'saladther': 1,
'getdaughter': 1,
'youmix': 1,
'branchmississippi': 1,
'orinin': 1,
'worthbought': 13,
'threewas': 1,
'oilnever': 1,
'scooter': 2,
'lovelate': 1,
'beforthis': 9,
'hummthese': 1,
'todaythere': 1,
'gonot': 2,
'distributwonder': 1,
'flourpaprika': 1,
'groupfor': 1,
'dietw': 1,
'centpleas': 1,
'herecat': 3,
'olivexcel': 1,
'sierra': 10,
'easichystal': 1,
```

```
'introducedthank': 1,
'pournice': 1,
'happimake': 1,
'lomalinda': 1,
'horid': 1,
'racheal': 1,
'alsobeen': 1,
'getorder': 7,
'meatsure': 1,
'podmerchtrident': 1,
'redenbach': 20,
'twowhat': 1,
'zum': 1,
'primal': 42,
'woodexot': 1,
'meatthese': 1,
'packagcrave': 1,
'tasticompar': 1,
'whohave': 1,
'vinci': 15,
'lasgna': 1,
'littlveri': 1,
'messbecaus': 1,
'horribljust': 1,
'steami': 13,
'mousthis': 1,
'crispwestbra': 1,
'amazonigourmetbeen': 1,
'hoop': 5,
'qualitikid': 1,
'clown': 9,
'enjoyprovid': 1,
'throatthis': 1,
'goodjack': 1,
'unavailget': 1,
'wonderbread': 1,
'chickenwhich': 1,
'goldish': 1,
'flourbad': 2,
```

```
'freshthrough': 1,
'waydescript': 1,
'choicexcel': 1,
'paterson': 2,
'lovegrandson': 1,
'offeredi': 1,
'stem': 122,
'temperaturlove': 2,
'she': 8126,
'agrumato': 1,
'notwas': 6,
'oragn': 1,
'happihusband': 3,
'oneonc': 1,
'betsidrink': 1,
'morndelici': 1,
'thisdescript': 1,
'valuehigh': 1,
'hardthis': 6,
'hahathis': 3,
'offhas': 1,
'oneforget': 1,
'cannedbox': 1,
'alreadihome': 1,
'everytimhave': 1,
'alaskalove': 1,
'tacki': 12,
'waterthey': 1,
'westlak': 1,
'wayself': 1,
'stockorder': 5,
'almonduo': 2,
'chaleng': 1,
'grean': 4,
'stape': 1,
'cassorol': 1,
'fanlight': 1,
'flashand': 1,
'inulin': 27,
```

```
'toomin': 1,
'occasfor': 1,
'acidboth': 1,
'plustoo': 1,
'liqour': 2,
'vend': 117,
'stold': 1,
'monthfar': 1,
'saltinessacid': 1,
'keton': 3,
'wholethis': 3,
'anymorbought': 2,
'richerdeer': 1,
'organicnaturalal': 1,
'acceptd': 1,
'cookieinclud': 1,
'kcup': 88,
'amazonnormal': 1,
'unawarhave': 1,
'nantuket': 1,
'mirthese': 2,
'productprepar': 1,
'heralthough': 1,
'overboardhave': 1,
'thyroidadren': 1,
'afterbit': 4,
'highseem': 2,
'giftworthi': 1,
'childrenuse': 1,
'blotch': 2,
'priceyou': 16,
'largestcoarsest': 2,
'stockpilgreat': 1,
'dextros': 47,
'medicianl': 1,
'plainhave': 1,
'poni': 11,
'everright': 1,
'pawpurchas': 2,
```

```
'greatther': 1,
'goodit': 12,
'triad': 1,
'grossgia': 1,
'youbig': 2,
'thumbgrain': 1,
'thishowev': 1,
'cinamonorang': 1,
'easilinabisco': 1,
'bayou': 2,
'noticthe': 1,
'outhard': 1,
'unprocess': 48,
'sweetit': 2,
'productsicilian': 1,
'whichfound': 1,
'uniqubought': 1,
'increasproduct': 1,
'candidefinit': 1,
'wellspecial': 1,
'brandall': 3,
'shakingit': 1,
'custom': 1271,
'sport': 195,
'realize': 4,
'holderi': 1,
'doggiyear': 1,
'flavortri': 14,
'bloatpicki': 1,
'disappointnone': 1,
'guestoutstand': 1,
'eleg': 112,
'complaintarriv': 1,
'unrat': 2,
'asda': 1,
'willexcel': 1,
'onlithese': 2,
'freeda': 1,
'bratwurstthere': 1,
```

```
'somethng': 2,
'cookiold': 1,
'sightlove': 1,
'fearsom': 1,
'deliciousspici': 1,
'honesti': 21,
'varietilove': 2,
'painter': 1,
'organicth': 2,
'tawari': 1,
'sanit': 7,
'leo': 6,
'waterbeen': 1,
'ordersize': 1,
'bigcat': 1,
'brushthese': 1,
'organicnot': 1,
'lastgood': 1,
'gumso': 1,
'overburn': 1,
'trulithese': 2,
'beignetthis': 1,
'morga': 2,
'cinnamonwonder': 1,
'buysh': 1,
'lbulgaricus': 1,
'roseann': 1,
'stockand': 1,
'blandpurchas': 1,
'junkthis': 3,
'pleasantmiss': 1,
'smackt': 1,
'medlarg': 1,
'outsidthis': 2,
'themthi': 2,
'nmr': 1,
'yeahthis': 2,
'partbe': 1,
'poloamaz': 1,
```

```
 'wualiti': 1,
 'xlichew': 1,
 'suggestwonder': 1,
 'forrespons': 1,
 'lovequick': 1,
 'withdrawl': 8,
 'claudett': 1,
 'hmay': 1,
 'themnumi': 1,
 'thingsespeci': 2,
 'actualmani': 1,
 'intolerthe': 1,
 'fornana': 1,
 'trapthese': 1,
 'thrownthis': 1,
 'seaweedi': 1,
 'solvyes': 1,
 'perfectizz': 1,
 'bahlsen': 16,
 'mooshyso': 1,
 'inexperienc': 3,
 'caloriwhat': 2,
 'sugarsinc': 2,
 'ricepaellarisotto': 1,
 ...}
```

In [15]:
```
import pickle
with open('freq_dict.pkl','wb') as file:
    pickle.dump(freq_dict,file)
```

Creating rank list of frequent words upto 5000

In [16]:
```
from operator import itemgetter
sorted_list= []
for k, v in sorted(freq_dict.items(), key=itemgetter(1),reverse=True):
    sorted_list.append(k)
```

In [17]:
```
sorted_list
```

Out[17]:
```
['the',
 'and',
 'this',
 'for',
 'that',
 'with',
 'you',
 'have',
 'but',
 'are',
 'not',
 'they',
 'was',
 'like',
 'tast',
 'flavor',
 'them',
 'these',
 'good',
 'tea',
 'one',
 'use',
 'can',
 'product',
 'veri',
 'great',
 'just',
 'tri',
 'all',
 'from',
 'love',
 'make',
 'has',
 'when',
 'get',
 'more',
 'other',
 'will',
```

```
                'than',
                'coffe',
                'had',
                'out',
                'would',
                'some',
                'buy',
                'food',
                'onli',
                'eat',
                'about',
                'time',
                'your',
                'find',
                'realli',
                'also',
                'best',
                'much',
                'too',
                'littl',
                'order',
                'even',
                'amazon',
                'becaus',
                'drink',
                'which',
                'were',
                'price',
                'bag',
                'there',
                'store',
                'been',
                'mix',
                'what',
                'chocol',
                'ani',
                'better',
                'well',
                'box',
```

```
                'sugar',
                'now',
                'year',
                'their',
                'after',
                'sweet',
                'found',
                'day',
                'dog',
                'want',
                'then',
                'high',
                'look',
                'our',
                'give',
                'cup',
                'over',
                'first',
                'add',
                'water',
                'brand',
                'recommend',
                'most',
                'she',
                'made',
                'think',
                'packag',
                'way',
                'who',
                'treat',
                'two',
                'nice',
                'work',
                'mani',
                'enjoy',
                'sinc',
                'favorit',
                'need',
                'thing',
```

```
                'know',
                'bar',
                'keep',
                'bit',
                'come',
                'differ',
                'milk',
                'could',
                'purchas',
                'say',
                'snack',
                'still',
                'lot',
                'free',
                'delici',
                'pack',
                'ship',
                'hot',
                'her',
                'take',
                'never',
                'review',
                'organ',
                'into',
                'without',
                'perfect',
                'wonder',
                'fresh',
                'everi',
                'doe',
                'ever',
                'befor',
                'how',
                'ingredi',
                'local',
                'sauc',
                'cook',
                'cat',
                'few',
```

```
                'alway',
                'easi',
                'bought',
                'put',
                'natur',
                'someth',
                'stuff',
                'seem',
                'cooki',
                'it',
                'oil',
                'whole',
                'healthi',
                'green',
                'contain',
                'did',
                'got',
                'enough',
                'hard',
                'while',
                'ad',
                'right',
                'qualiti',
                'rice',
                'those',
                'same',
                'back',
                'regular',
                'less',
                'dri',
                'last',
                'candi',
                'small',
                'salt',
                'cereal',
                'here',
                'calori',
                'again',
                'howev',
```

```
                'long',
                'serv',
                'fruit',
                'groceri',
                'each',
                'actual',
                'size',
                'tasti',
                'quick',
                'quit',
                'feel',
                'see',
                'far',
                'sure',
                'old',
                'strong',
                'excel',
                'definit',
                'off',
                'though',
                'both',
                'month',
                'textur',
                'his',
                'peopl',
                'chip',
                'bread',
                'juic',
                'bottl',
                'be',
                'help',
                'chicken',
                'bean',
                'problem',
                'start',
                'anoth',
                'big',
                'real',
                'through',
```

```
                'down',
                'fat',
                'smell',
                'item',
                'butter',
                'open',
                'bad',
                'case',
                'soup',
                'almost',
                'famili',
                'blend',
                'kid',
                'chees',
                'may',
                'pretti',
                'sever',
                'happi',
                'usual',
                'per',
                'thought',
                'friend',
                'should',
                'low',
                'go',
                'top',
                'black',
                'new',
                'bake',
                'gluten',
                'diet',
                'amount',
                'anyth',
                'avail',
                'varieti',
                'compani',
                'pasta',
                'thank',
                'worth',
```

```
                'light',
                'recip',
                'minut',
                'peanut',
                'own',
                'around',
                'expens',
                'onc',
                'arriv',
                'ice',
                'kind',
                'nut',
                'home',
                'chew',
                'prefer',
                'reason',
                'full',
                'protein',
                'where',
                'week',
                'health',
                'morn',
                'half',
                'especi',
                'came',
                'abl',
                'dark',
                'syrup',
                'meal',
                'carri',
                'receiv',
                'until',
                'powder',
                'probabl',
                'white',
                'spice',
                'save',
                'cost',
                'expect',
```

```
                'bitter',
                'gift',
                'him',
                'leav',
                'fill',
                'piec',
                'brew',
                'pleas',
                'honey',
                'might',
                'said',
                'breakfast',
                'such',
                'hand',
                'roast',
                'star',
                'away',
                'vanilla',
                'corn',
                'call',
                'fact',
                'rich',
                'place',
                'larg',
                'compar',
                'extra',
                'cream',
                'absolut',
                'three',
                'instead',
                'live',
                'whi',
                'coconut',
                'sweeten',
                'hope',
                'cake',
                'noth',
                'read',
                'wheat',
```

```
                'type',
                'disappoint',
                'surpris',
                'red',
                'least',
                'let',
                'season',
                'coupl',
                'market',
                'cracker',
                'slight',
                'wish',
                'care',
                'satisfi',
                'smooth',
                'ago',
                'fine',
                'decid',
                'husband',
                'meat',
                'jar',
                'end',
                'turn',
                'soda',
                'second',
                'addit',
                'next',
                'yet',
                'deal',
                'pepper',
                'longer',
                'chang',
                'plus',
                'although',
                'stop',
                'offer',
                'run',
                'includ',
                'color',
```

```
                'went',
                'tell',
                'list',
                'must',
                'sell',
                'anyon',
                'money',
                'either',
                'person',
                'packet',
                'cherri',
                'spici',
                'altern',
                'fiber',
                'noodl',
                'orang',
                'dish',
                'gave',
                'glad',
                'almond',
                'mayb',
                'salad',
                'side',
                'appl',
                'oliv',
                'part',
                'hous',
                'sometim',
                'amaz',
                'stick',
                'notic',
                'etc',
                'fast',
                'soft',
                'version',
                'heat',
                'seed',
                'cheaper',
                'past',
```

```
                'soy',
                'cold',
                'everyth',
                'origin',
                'believ',
                'fan',
                'everyon',
                'els',
                'cut',
                'rather',
                'mouth',
                'pay',
                'pod',
                'choic',
                'myself',
                'crunchi',
                'flour',
                'stock',
                'hour',
                'brown',
                'conveni',
                'oatmeal',
                'experi',
                'took',
                'tomato',
                'gum',
                'potato',
                'bowl',
                'weight',
                'ounc',
                'special',
                'lemon',
                'close',
                'often',
                'popcorn',
                'son',
                'nutrit',
                'direct',
                'espresso',
```

```
                'cinnamon',
                'exact',
                'total',
                'energi',
                'egg',
                'clean',
                'easili',
                'valu',
                'pound',
                'grain',
                'ask',
                'prepar',
                'mean',
                'granola',
                'consist',
                'goe',
                'plain',
                'onlin',
                'base',
                'normal',
                'feed',
                'strawberri',
                'switch',
                'artifici',
                'set',
                'machin',
                'result',
                'vitamin',
                'life',
                'caffein',
                'mild',
                'becom',
                'combin',
                'near',
                'pot',
                'recent',
                'particular',
                'cours',
                'microwav',
```

```
                'bite',
                'simpli',
                'pop',
                'cannot',
                'stay',
                'final',
                'label',
                'bodi',
                'instant',
                'complet',
                'replac',
                'carb',
                'beef',
                'date',
                'ground',
                'salti',
                'similar',
                'wait',
                'certain',
                'sodium',
                'dure',
                'babi',
                'process',
                'night',
                'search',
                'teeth',
                'substitut',
                'between',
                'plastic',
                'lunch',
                'pleasant',
                'suggest',
                'aroma',
                'ginger',
                'consid',
                'continu',
                'shop',
                'line',
                'bring',
```

```
                'cocoa',
                'super',
                'effect',
                'benefit',
                'name',
                'check',
                'content',
                'fair',
                'pick',
                'four',
                'daughter',
                'veget',
                'except',
                'servic',
                'chewi',
                'bulk',
                'gram',
                'finish',
                'deliv',
                'smaller',
                'pancak',
                'chai',
                'extrem',
                'rate',
                'oat',
                'daili',
                'along',
                'individu',
                'pure',
                'addict',
                'guess',
                'rememb',
                'state',
                'fish',
                'left',
                'point',
                'eaten',
                'provid',
                'sit',
```

```
                'pill',
                'yummi',
                'berri',
                'christma',
                'note',
                'delight',
                'entir',
                'overal',
                'formula',
                'given',
                'decaf',
                'custom',
                'singl',
                'gummi',
                'follow',
                'itself',
                'someon',
                'jerki',
                'wife',
                'area',
                'glass',
                'acid',
                'alreadi',
                'carbon',
                'insid',
                'deliveri',
                'raw',
                'pet',
                'huge',
                'interest',
                'fantast',
                'coat',
                'world',
                'thick',
                'mint',
                'bold',
                'sold',
                'later',
                'french',
```

```
                'pour',
                'supermarket',
                'idea',
                'aftertast',
                'mine',
                'truli',
                'refresh',
                'wrong',
                'licoric',
                'discov',
                'chili',
                'watch',
                'toy',
                'roll',
                'throw',
                'import',
                'cover',
                'beverag',
                'garlic',
                'dinner',
                'larger',
                'boil',
                'true',
                'due',
                'hold',
                'share',
                'creami',
                'sourc',
                'liquid',
                'consum',
                'ate',
                'saw',
                'hit',
                'togeth',
                'allergi',
                'rest',
                'caus',
                'wast',
                'dress',
```

```
                'hint',
                'preserv',
                'maker',
                'restaur',
                'yes',
                'issu',
                'miss',
                'mind',
                'general',
                'five',
                'balanc',
                'produc',
                'clear',
                'gone',
                'break',
                'slice',
                'sour',
                'sale',
                'herb',
                'difficult',
                'sent',
                'fun',
                'told',
                'suppos',
                'melt',
                'lower',
                'warm',
                'mention',
                'loos',
                'simpl',
                'done',
                'within',
                'drop',
                'return',
                'worri',
                'stomach',
                'unfortun',
                'crunch',
                'veggi',
```

```
                'awesom',
                'plant',
                'option',
                'dip',
                'bottom',
                'anyway',
                'beat',
                'wrap',
                'excit',
                'crave',
                'amazoncom',
                'impress',
                'healthier',
                'tradit',
                'plan',
                'thin',
                'batch',
                'starbuck',
                'easier',
                'allow',
                'picki',
                'seen',
                'unlik',
                'onion',
                'level',
                'anywher',
                'shape',
                'bone',
                'vet',
                'under',
                'raisin',
                'twice',
                'banana',
                'grow',
                'fit',
                'figur',
                'drinker',
                'immedi',
                'avoid',
```

```
                'beauti',
                'fri',
                'soon',
                'yogurt',
                'stir',
                'happen',
                'test',
                'busi',
                'grey',
                'opinion',
                'kitchen',
                'diabet',
                'yourself',
                'possibl',
                'tuna',
                'toast',
                'italian',
                'show',
                'crisp',
                'websit',
                'unless',
                'number',
                'sort',
                'none',
                'chemic',
                'tend',
                'blueberri',
                'cool',
                'form',
                'requir',
                'seal',
                'hate',
                'moist',
                'sandwich',
                'refriger',
                'children',
                'remind',
                'skin',
                'concern',
```

```
                'do',
                'vinegar',
                'sampl',
                'send',
                'digest',
                'uniqu',
                'six',
                'matter',
                'browni',
                'homemad',
                'shake',
                'today',
                'perhap',
                'pie',
                'stand',
                'tin',
                'cheap',
                'mapl',
                'basic',
                'site',
                'medium',
                'manufactur',
                'agre',
                'wine',
                'blue',
                'describ',
                'subscrib',
                'short',
                'trip',
                'blood',
                'earl',
                'touch',
                'condit',
                'flower',
                'keurig',
                'suppli',
                'lover',
                'main',
                'shipment',
```

```
                'tini',
                'extract',
                'trap',
                'standard',
                'sprinkl',
                'typic',
                'previous',
                'stevia',
                'biscuit',
                'gourmet',
                'steep',
                'tart',
                'splenda',
                'appreci',
                'expir',
                'hook',
                'broken',
                'realiz',
                'improv',
                'muffin',
                'readi',
                'american',
                'concentr',
                'bland',
                'instruct',
                'travel',
                'complaint',
                'mill',
                'anymor',
                'move',
                'imagin',
                'overpow',
                'raspberri',
                'tree',
                'bear',
                'higher',
                'remov',
                'word',
                'somewhat',
```

```
                'research',
                'quantiti',
                'portion',
                'reduc',
                'visit',
                'mom',
                'dessert',
                'whatev',
                'offic',
                'shelf',
                'incred',
                'caramel',
                'english',
                'doubl',
                'count',
                'creat',
                'pan',
                'otherwis',
                'pocket',
                'learn',
                'sound',
                'straight',
                'seller',
                'pizza',
                'leaf',
                'decent',
                'stale',
                'frozen',
                'broth',
                'subtl',
                'sea',
                'tablespoon',
                'heavi',
                'thrill',
                'spoon',
                'grape',
                'waffl',
                'kept',
                'spread',
```

```
                'claim',
                'senseo',
                'lose',
                'summer',
                'cashew',
                'pictur',
                'weak',
                'brought',
                'ball',
                'forward',
                'dollar',
                'salmon',
                'parti',
                'control',
                'system',
                'chunk',
                'train',
                'kick',
                'mess',
                'curri',
                'serious',
                'flake',
                'puppi',
                'spend',
                'across',
                'mustard',
                'terribl',
                'increas',
                'vegan',
                'felt',
                'knew',
                'appear',
                'charg',
                'inform',
                'various',
                'chanc',
                'pricey',
                'plenti',
                'chop',
```

```
                'lack',
                'sick',
                'hazelnut',
                'room',
                'outsid',
                'stronger',
                'teaspoon',
                'alon',
                'age',
                'known',
                'herbal',
                'afternoon',
                'relat',
                'fall',
                'heart',
                'heard',
                'freez',
                'yeast',
                'present',
                'troubl',
                'vegetarian',
                'play',
                'job',
                'comment',
                'paper',
                'descript',
                'crust',
                'celiac',
                'style',
                'power',
                'smoke',
                'pouch',
                'whether',
                'jelli',
                'burn',
                'grill',
                'premium',
                'sensit',
                'late',
```

```
 'prompt',
 'fridg',
 'crazi',
 'mother',
 'pretzel',
 'book',
 'school',
 'pass',
 'crispi',
 'everyday',
 'paid',
 'indian',
 'aw',
 'understand',
 'nutti',
 'child',
 'grind',
 'safe',
 'introduc',
 'quinoa',
 'sticki',
 'allerg',
 'agav',
 'supplement',
 'remain',
 'equal',
 ...]
```

In [18]:
```
Data_x[1]
```

Out[18]:
```
'can rememb see the show when air televis year ago when was child siste
r later bought the which have this day thirti somethingi use this seri
book song when did student teach for preschool turn the whole school no
w purchas along with the book for children the tradit live'
```

In [19]:
```
top_words= 5000
sorted_list= sorted_list[:5000]
```

Transforming Sentences of words to sequence of rank number of words

In [20]:
```
column=[]
for sent in Data_x:
    lis=[]
    for word in sent.split():
        if word in sorted_list:
            lis.append(word)
    column.append(' '.join(lis))
```

In [21]:
```
with open('column.pkl','wb') as file:
    pickle.dump(column,file)
```

In [22]:
```
final_x=[]
for sent in Data_x:
    lis=[]
    for word in sent.split():
        if word in sorted_list:
            lis.append(sorted_list.index(word)+1)
    final_x.append(lis)
```

In [23]:
```
Xtest= final_x[:30000]
Ytest= Data_y[:30000]
Xtrain= final_x[30000:]
Ytrain= Data_y[30000:]
```

In [24]:
```
print(Xtrain[1])
```

```
[8, 216, 209, 1974, 106, 8, 368, 25, 3984, 528, 3228, 3228, 721, 17, 48
8, 1101, 2, 583, 127, 5, 1070, 474, 5, 11, 697, 8, 47, 28, 1, 557, 16,
3228, 3228, 9, 322, 1, 31, 89, 99, 12, 10, 156, 621, 42, 91, 151, 69, 5
41, 35, 1133, 17, 61, 5, 8, 352, 1, 69]
```

Applying LSTM models

```
In [25]: from keras.preprocessing import sequence
         max_review_length=600
         Xtrain = sequence.pad_sequences(Xtrain, maxlen=max_review_length)
         Xtest= sequence.pad_sequences(Xtest, maxlen=max_review_length)
```

```
Using TensorFlow backend.
```

```
In [26]: print(Xtrain[1])
```

```
[    0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    8  216  209
 1974  106    8  368   25 3984  528 3228 3228  721   17  488 1101    2
  583  127    5 1070  474    5   11  697    8   47   28    1  557   16
 3228 3228    9  322    1   31   89   99   12   10  156  621   42   91
  151   69  541   35 1133   17   61    5    8  352    1   69]
```

In [27]:
```
from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())
```

```
[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 14921882868709283565
, name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 3206820659
locality {
  bus_id: 1
  links {
  }
}
incarnation: 192071761630059385
physical_device_desc: "device: 0, name: GeForce 940MX, pci bus id: 000
0:01:00.0, compute capability: 5.0"
]
```

In [28]:
```
import numpy
from keras.datasets import imdb
from keras.models import Sequential
```

```
from keras.layers import Dense
from keras.layers import LSTM,Dropout
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
# fix random seed for reproducibility
numpy.random.seed(7)
```

Model 1 same as of IMDB

In [30]:
```
# create the model
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words+1, embedding_vecor_length, input_length=m
ax_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['a
ccuracy'])
print(model.summary())
```

```
WARNING:tensorflow:From C:\Anaconda3\lib\site-packages\tensorflow\pytho
n\framework\op_def_library.py:263: colocate_with (from tensorflow.pytho
n.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 600, 32)           160032
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               53200
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 101
=================================================================
Total params: 213,333
Trainable params: 213,333
Non-trainable params: 0
```

```
_________________________________________________________________
None
```

In [31]:
```
history=  model.fit(Xtrain, Ytrain,
          batch_size=64,
          epochs=10,
          verbose=1,
          validation_data=(Xtest, Ytest))# Final evaluation of the mode
l
scores = model.evaluate(Xtest, Ytest, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```
WARNING:tensorflow:From C:\Anaconda3\lib\site-packages\tensorflow\pytho
n\ops\math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops)
is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 70000 samples, validate on 30000 samples
Epoch 1/10
70000/70000 [==============================] - 1783s 25ms/step - loss:
0.2449 - acc: 0.9067 - val_loss: 0.1778 - val_acc: 0.9352
Epoch 2/10
70000/70000 [==============================] - 1736s 25ms/step - loss:
0.1786 - acc: 0.9305 - val_loss: 0.1742 - val_acc: 0.9355
Epoch 3/10
70000/70000 [==============================] - 1733s 25ms/step - loss:
0.1604 - acc: 0.9382 - val_loss: 0.1718 - val_acc: 0.9359
Epoch 4/10
70000/70000 [==============================] - 1745s 25ms/step - loss:
0.1450 - acc: 0.9452 - val_loss: 0.1764 - val_acc: 0.9338
Epoch 5/10
70000/70000 [==============================] - 1733s 25ms/step - loss:
0.1317 - acc: 0.9508 - val_loss: 0.1808 - val_acc: 0.9380
Epoch 6/10
70000/70000 [==============================] - 1741s 25ms/step - loss:
0.1131 - acc: 0.9573 - val_loss: 0.1806 - val_acc: 0.9352
Epoch 7/10
70000/70000 [==============================] - 1737s 25ms/step - loss:
0.1010 - acc: 0.9632 - val_loss: 0.1917 - val_acc: 0.9310
```

```
Epoch 8/10
70000/70000 [==============================] - 1735s 25ms/step - loss:
0.0902 - acc: 0.9673 - val_loss: 0.1971 - val_acc: 0.9346
Epoch 9/10
70000/70000 [==============================] - 1734s 25ms/step - loss:
0.0796 - acc: 0.9716 - val_loss: 0.2208 - val_acc: 0.9317
Epoch 10/10
70000/70000 [==============================] - 1735s 25ms/step - loss:
0.0718 - acc: 0.9745 - val_loss: 0.2345 - val_acc: 0.9313
Accuracy: 93.13%
```

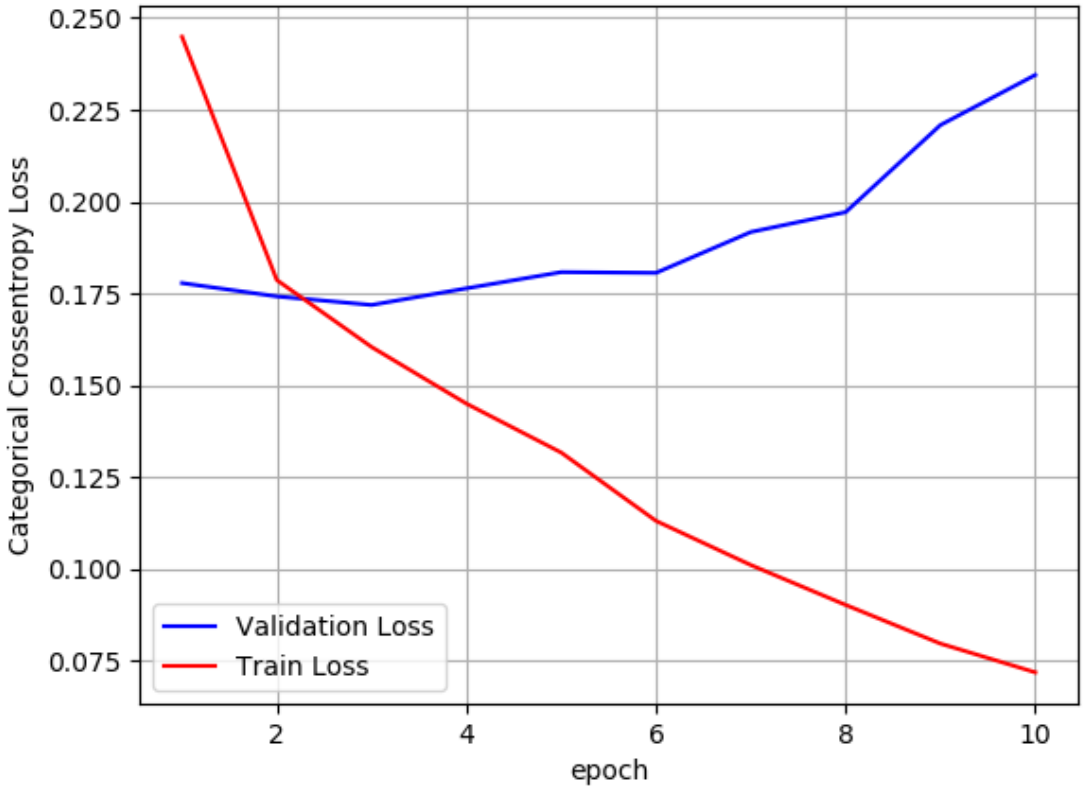
In [32]:
```
#import pickle
#with open('model.pkl','rb') as file:
#    model=pickle.load(file)
#with open('model2.pkl','rb') as file:
#    model2=pickle.load(file)
```

In [33]:
```
%matplotlib notebook
import matplotlib.pyplot as plt
import numpy as np
import time
# https://gist.github.com/greydanus/f6eee59eaf1d90fcb3b534a25362cea4
# https://stackoverflow.com/a/14434334
# this function is used to update the plots for each epoch and error
def plt_dynamic(x, vy, ty, ax, colors=['b']):
    ax.plot(x, vy, 'b', label="Validation Loss")
    ax.plot(x, ty, 'r', label="Train Loss")
    plt.legend()
    plt.grid()
    fig.canvas.draw()
```

In [35]:
```
score= model.evaluate(Xtest, Ytest, verbose=0)
print('Test score: ',score[0])
print('Test accuracy: ',score[1])
```

```
Test score:  0.2344630477592349
Test accuracy:  0.9313
```

In [37]:
```
fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')
x = list(range(1,11))
vy = model.history.history['val_loss']
ty = model.history.history['loss']
plt_dynamic(x, vy, ty, ax)
```



Model 2

In [38]:
```
# create the model
```

```
embedding_vecor_length = 32
model2 = Sequential()
model2.add(Embedding(top_words+1, embedding_vecor_length, input_length=
max_review_length))
model2.add(LSTM(100,return_sequences=True))
model2.add(Dropout(0.25))
model2.add(LSTM(80))
model2.add(Dropout(0.5))
model2.add(Dense(1, activation='sigmoid'))
model2.compile(loss='binary_crossentropy', optimizer='adam', metrics=[
'accuracy'])
print(model2.summary())
```

```
WARNING:tensorflow:From C:\Anaconda3\lib\site-packages\keras\backend\te
nsorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.n
n_ops) with keep_prob is deprecated and will be removed in a future ver
sion.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
1 - keep_prob`.

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 600, 32)           160032
_________________________________________________________________
lstm_2 (LSTM)                (None, 600, 100)          53200
_________________________________________________________________
dropout_1 (Dropout)          (None, 600, 100)          0
_________________________________________________________________
lstm_3 (LSTM)                (None, 80)                57920
_________________________________________________________________
dropout_2 (Dropout)          (None, 80)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 81
=================================================================
Total params: 271,233
Trainable params: 271,233
Non-trainable params: 0
_________________________________________________________________
None
```

In [39]:
```
history2=  model2.fit(Xtrain, Ytrain,
          batch_size=64,
          epochs=10,
          verbose=1,
          validation_data=(Xtest, Ytest))# Final evaluation of the mode
l
scores = model2.evaluate(Xtest, Ytest, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

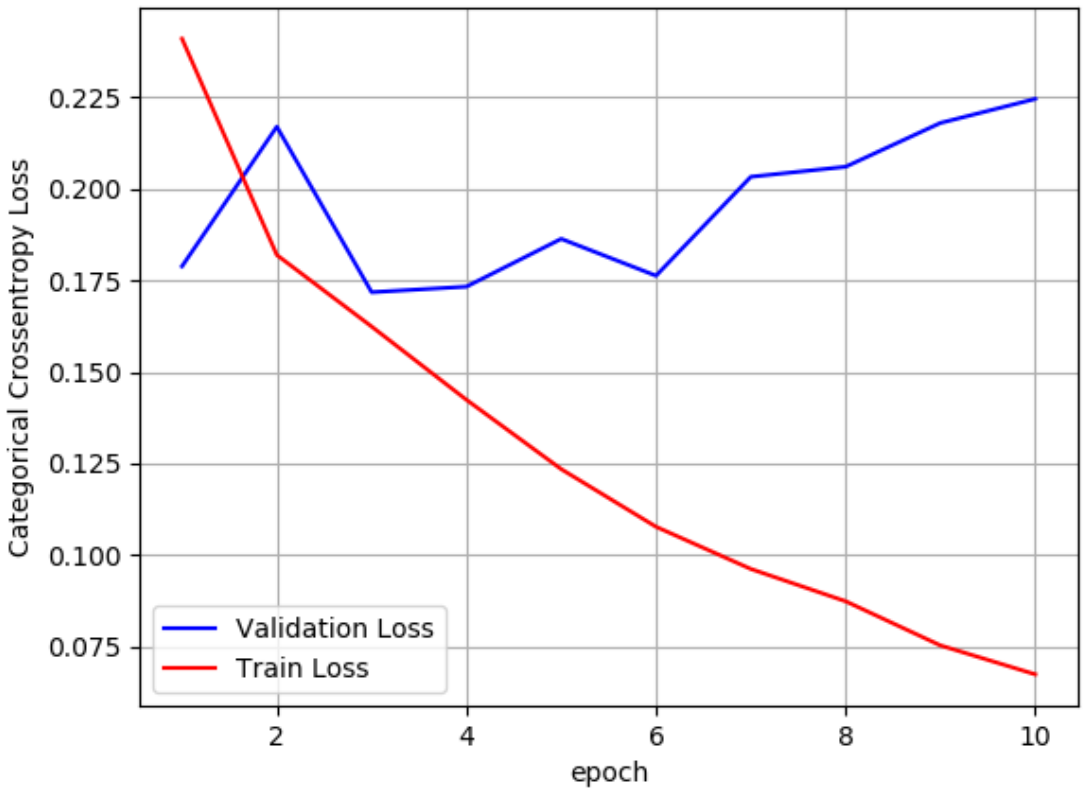
```
Train on 70000 samples, validate on 30000 samples
Epoch 1/10
70000/70000 [==============================] - 3748s 54ms/step - loss:
0.2410 - acc: 0.9094 - val_loss: 0.1789 - val_acc: 0.9314
Epoch 2/10
70000/70000 [==============================] - 3726s 53ms/step - loss:
0.1820 - acc: 0.9304 - val_loss: 0.2170 - val_acc: 0.9139
Epoch 3/10
70000/70000 [==============================] - 4011s 57ms/step - loss:
0.1624 - acc: 0.9375 - val_loss: 0.1718 - val_acc: 0.9372
Epoch 4/10
70000/70000 [==============================] - 3585s 51ms/step - loss:
0.1424 - acc: 0.9465 - val_loss: 0.1733 - val_acc: 0.9343
Epoch 5/10
70000/70000 [==============================] - 3590s 51ms/step - loss:
0.1235 - acc: 0.9548 - val_loss: 0.1864 - val_acc: 0.9331
Epoch 6/10
70000/70000 [==============================] - 3581s 51ms/step - loss:
0.1078 - acc: 0.9613 - val_loss: 0.1763 - val_acc: 0.9363
Epoch 7/10
70000/70000 [==============================] - 3581s 51ms/step - loss:
0.0962 - acc: 0.9662 - val_loss: 0.2033 - val_acc: 0.9269
Epoch 8/10
70000/70000 [==============================] - 3580s 51ms/step - loss:
0.0874 - acc: 0.9694 - val_loss: 0.2060 - val_acc: 0.9332
Epoch 9/10
70000/70000 [==============================] - 3580s 51ms/step - loss:
0.0754 - acc: 0.9748 - val_loss: 0.2180 - val_acc: 0.9330
Epoch 10/10
70000/70000 [==============================] - 3576s 51ms/step - loss:
```

```
0.0675 - acc: 0.9775 - val_loss: 0.2246 - val_acc: 0.9295
Accuracy: 92.95%
```

In [40]:
```
score= model2.evaluate(Xtest, Ytest, verbose=0)
print('Test score: ',score[0])
print('Test accuracy: ',score[1])
```

```
Test score:  0.2245582093084852
Test accuracy:  0.9295
```

In [41]:
```
fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')
x = list(range(1,11))
vy = model2.history.history['val_loss']
ty = model2.history.history['loss']
plt_dynamic(x, vy, ty, ax)
```



Testing our model on self made review sentence

In [42]:
```
# making a function which convert sentance to requred vectorized format
 that will feed well in model

def cleanhtml(sentance): #substitute expression contained in <> with '
 '
    cleaned= re.sub(re.compile('<.*?>'),' ',sentance)
    return cleaned
#function for removing punctuations chars
```

```
def cleanpunc(sentance):
    cleaned= re.sub(r'[?|!|\'|"|#]',r'',sentance)
    cleaned= re.sub(r'[.|,|)|(|\|/]',r'',sentance)
    return cleaned
snowstem= sno('english')

def predict_this(sentance):
    i=0
    str1=' '
    final_string=[]
    all_positive_words=[] # store words from +ve reviews here
    all_negative_words=[] # store words from -ve reviews here.
    sent= sentance
    filtered_sentence=[]
    #print(sent);
    sent=cleanhtml(sent) # remove HTml tags
    for w in sent.split():
        # we have used cleanpunc(w).split(), one more split function he
re
        # because consider w="abc.def", cleanpunc(w) will return "abc d
ef"
        # if we dont use .split() function then we will be considring
 "abc def"
        # as a single word, but if you use .split() function we will ge
t "abc", "def"
        for cleaned_words in cleanpunc(w).split():
            if((cleaned_words.isalpha()) & (len(cleaned_words)>2)):
                s=(snowstem.stem(cleaned_words.lower())).encode('utf8')
                filtered_sentence.append(s)
                if(data['Score'].values)[i] =='Positive':
                    all_positive_words.append(s)
                if(data['Score'].values)[i] =='Negative':
                    all_negative_words.append(s)
            else:
                continue
    str1 = b" ".join(filtered_sentence) #final string of cleaned words
    final_string.append(str1)

    final_string
```

```
    for i in final_string:
        final_string=i.decode("utf-8")

    lis=[]
    for word in final_string.split():
        if word in sorted_list:
            lis.append(sorted_list.index(word)+1)

    final_string= lis
    final_string = sequence.pad_sequences([final_string], maxlen=max_re
view_length)
    print(final_string)
    what= ''
    if (round(float(model2.predict(final_string)))==1):
        what= 'Positive'
        acc= round(float(model2.predict(final_string))*100,2)
    else:
        what='Negative'
        acc= 100- round(float(model2.predict(final_string))*100,2)
    print(what,'review with',acc,'% Accuracy')
```

In [43]:
```
sentance= 'food was very bad in taste'
predict_this(sentance)
```

```
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
```

```
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0   46   13   25  240   15]]
Negative review with 93.96 % Accuracy
```

In [44]:
```
sentance= 'taste of chocolate was fantastic'
predict_this(sentance)
```

```
[[  0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
```

```
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0   15   73   13  615]]
Positive review with 97.85 % Accuracy
```

In [45]:

```python
sentance= 'food was medium tasty'
predict_this(sentance)
```

```
[[   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
0
```

```
    0    0   46   13  800  202]]
Positive review with 94.21 % Accuracy
```

In [46]:
```python
print(data['Text'][1])
predict_this(data['Text'][1])
```

```
Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actu
ally small sized unsalted. Not sure if this was an error or if the vend
or intended to represent the product as "Jumbo".
[[    0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

```
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0   24  281  513 3647  189  276   65  200  188  201
  2768   11  208    3   13 2449    1 1184 1402 2553    1   24]]
Negative review with 98.2 % Accuracy
```

Summary

- Accuracy of First model is 93.13%
- Accuracy of Second model is 92.95%