In [2]:

```python
from google.colab import drive
drive.mount('/content/gdrive')
```

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6
qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%
b&scope=email%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdocs.test%20https%3A%2F%2Fwww.googleapis.
2Fauth%2Fdrive%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photos.readonly%20https%3A%2F%2Fww
ogleapis.com%2Fauth%2Fpeopleapi.readonly&response_type=code

Enter your authorization code:
....
Mounted at /content/gdrive

In [3]:

```python
import pandas as pd
import io


final =  pd.read_csv('gdrive/My Drive/CSV/final_cleaned.csv')
final.head()
```

Out[3]:

	Unnamed: 0	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Sco
0	138706	150524	0006641040	ACITT7DI6IDDL	shari zychinski	0	0	1
1	138683	150501	0006641040	AJ46FKXOVC7NR	Nicholas A Mesiano	2	2	1
2	417839	451856	B00004CXX9	AIUWLEQ1ADEG5	Elizabeth Medina	0	0	1
3	346055	374359	B00004CI84	A344SMIA5JECGM	Vincent P. Ross	1	2	1
4	417838	451855	B00004CXX9	AJH6LUC1UT1ON	The Phantom of the Opera	0	0	1

In [4]:

```python
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
from keras.layers import Dropout
```

In [5]:

```python
final['Score'].value_counts()
```

Out[5]:

```
 1    307061
-1     57110
Name: Score, dtype: int64
```

In [6]:

```python
final.Score.replace(to_replace= -1, value=0, inplace=True)
final['Score'].value_counts()
```

Out[6]:

```
1    307061
0     57110
Name: Score, dtype: int64
```

In [7]:

```python
final.head()
```

Out[7]:

	Unnamed: 0	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Sco
0	138706	150524	0006641040	ACITT7DI6IDDL	shari zychinski	0	0	1
1	138683	150501	0006641040	AJ46FKXOVC7NR	Nicholas A Mesiano	2	2	1
2	417839	451856	B00004CXX9	AIUWLEQ1ADEG5	Elizabeth Medina	0	0	1
3	346055	374359	B00004CI84	A344SMIA5JECGM	Vincent P. Ross	1	2	1
4	417838	451855	B00004CXX9	AJH6LUC1UT1ON	The Phantom of the Opera	0	0	1

In [0]:

```python
final = final[0:50000] #taking the first 50k reviews
```

In [9]:

```python
z = final['New'].values
z
```

²

```
array(['everi book educwitti littl book make son laugh loud recit car drive along alway sing
refrain hes learn whale india droop love new word book introduc silli classic book will bet son st
ill abl recit memori colleg',
       'whole seri great way spend time childrememb see show air televis year ago child sister
later bought day thirti someth use seri book song student teach preschool turn whole school purcha
s along book children tradit live',
       'entertainingl funnibeetlejuic well written movi everyth excel act special effect delight ch
ose view movi',
       ...,
       'baconnaiswould thought great stuff vegetarian mayo tast like bacon work well burger sandwic
h salad egg salad etc dip complaint jar small ounc even though sodium content relat low tast salti
er side actual take baconnais mix regular mayo cut back salti make smaller jar last longer',
       'good stuffenergi berri bttls satisfi product drink cup coffe go bed sleep like babi even dr
ink red bull result there energi drink call cocain work yet convient energi keep car desk purs whe
nev start nod grab bottl within minut good hour longer sister call florida ask name eneri drink hu
sband travel',
       'best farfirst introduc british friend show make proper roast potato use spice cours
continu tradit also add fresh pop popcorn homemad crouton anoth favorit use grill zuccini say conf
id product realli cant dissapoint good'],
      dtype=object)
```

◄| |►

In [10]:

```python
from nltk import FreqDist

#https://stackoverflow.com/questions/41699065/create-vocabulary-dictionary-for-text-mining


train_set = final["New"]
word_dist = FreqDist()
for s in train_set:
    word_dist.update(s.split())

word_dist = dict(word_dist)
word_dist
```

Out[10]:

```
{'everi': 2989,
 'book': 424,
 'educwitti': 1,
 'littl': 6732,
 'make': 11801,
 'son': 1036,
 'laugh': 84,
 'loud': 43,
 'recit': 3,
 'car': 261,
 'drive': 220,
 'along': 700,
 'alway': 2882,
 'sing': 47,
 'refrain': 3,
 'hes': 341,
 'learn': 395,
 'whale': 4,
 'india': 164,
 'droop': 6,
 'love': 13563,
 'new': 1984,
 'word': 474,
 'introduc': 318,
 'silli': 58,
 'classic': 328,
 'will': 155,
 'bet': 119,
 'still': 3215,
 'abl': 1449,
 'memori': 247,
 'colleg': 172,
 'whole': 2831,
 'seri': 56,
```

'great': 20108,
'way': 4247,
'spend': 376,
'time': 7991,
'childrememb': 1,
'see': 2190,
'show': 444,
'air': 268,
'televis': 15,
'year': 4976,
'ago': 1163,
'child': 391,
'sister': 224,
'later': 626,
'bought': 2571,
'day': 4929,
'thirti': 37,
'someth': 2730,
'use': 13775,
'song': 35,
'student': 99,
'teach': 50,
'preschool': 21,
'turn': 1096,
'school': 382,
'purchas': 3121,
'children': 581,
'tradit': 676,
'live': 1401,
'entertainingl': 1,
'funnibeetlejuic': 1,
'well': 6066,
'written': 89,
'movi': 402,
'everyth': 1096,
'excel': 3647,
'act': 115,
'special': 1098,
'effect': 777,
'delight': 796,
'chose': 99,
'view': 55,
'modern': 47,
'fairi': 3,
'taletwist': 1,
'rumplestiskin': 1,
'captur': 24,
'film': 187,
'star': 1698,
'michael': 55,
'keaton': 66,
'geena': 34,
'davi': 49,
'prime': 147,
'tim': 87,
'burton': 97,
'masterpiec': 24,
'rumbl': 5,
'absurd': 14,
'wonder': 3933,
'pace': 87,
'point': 647,
'dull': 42,
'moment': 182,
'fantastbeetlejuic': 1,
'funni': 212,
'hilari': 24,
'wacki': 9,
'beetlejuic': 113,
'help': 2242,
'think': 3996,
'one': 14636,
'best': 9955,
'ever': 3501,
'made': 4037,
'sure': 2130,
'youll': 1369,

```
'agre': 455,
'good': 19175,
'watch': 674,
'greatone': 9,
'collect': 131,
'fill': 1517,
'comedi': 32,
'action': 43,
'whatev': 409,
'els': 1085,
'want': 4400,
'call': 1379,
'clamshel': 3,
'edit': 94,
'versionalway': 1,
'enjoy': 4352,
'entertain': 130,
'didnt': 2163,
'hesit': 214,
'pick': 714,
'guess': 646,
'market': 1393,
'plan': 517,
'famili': 2271,
'elimin': 152,
'strong': 2263,
'profan': 1,
'element': 70,
'usual': 1797,
'version': 1100,
'warn': 320,
'uncut': 13,
'avoid': 506,
'death': 100,
'flibought': 1,
'apart': 269,
'infest': 29,
'fruit': 2655,
'fli': 153,
'hour': 970,
'trap': 307,
'mani': 3609,
'within': 574,
'practic': 289,
'gone': 605,
'may': 1889,
'long': 2381,
'term': 177,
'solut': 111,
'crazi': 387,
'consid': 769,
'buy': 7941,
'surfac': 83,
'sticki': 320,
'tri': 12326,
'touch': 460,
'bettlejuic': 6,
'bettlejuichappen': 1,
'say': 3259,
'name': 784,
'three': 1316,
'keaten': 1,
'two': 3623,
'coupl': 1181,
'old': 2204,
'stori': 176,
'hous': 1141,
'come': 3563,
'back': 2525,
'suppli': 504,
'store': 5725,
'sudden': 110,
'get': 10526,
'caught': 110,
'insid': 638,
'bridg': 16,
'start': 1915,
```

'tumbl': 5,
'lake': 40,
'board': 58,
'catch': 110,
'theyv': 169,
'got': 2474,
'hope': 1171,
'small': 2476,
'dog': 5332,
'step': 167,
'slide': 32,
'water': 4162,
'minut': 1764,
'find': 7437,
'home': 1648,
'somehow': 124,
'somehad': 1,
'light': 1979,
'fireplac': 6,
'done': 592,
'magic': 293,
'dead': 146,
'guy': 293,
'known': 355,
'appear': 333,
'survic': 1,
'soon': 609,
'wish': 1152,
'never': 3063,
'troublemak': 2,
'save': 1428,
'account': 59,
'said': 1353,
'cant': 3287,
'leav': 1575,
'theirselv': 1,
'anoth': 1992,
'world': 864,
'giant': 100,
'sandworm': 3,
'stellar': 17,
'awesom': 987,
'play': 355,
'lead': 144,
'role': 31,
'researchget': 1,
'realli': 7276,
'imposs': 212,
'today': 505,
'french': 650,
'vhs': 6,
'could': 3430,
'pleas': 1482,
'tell': 1099,
'tks': 4,
'research': 310,
'beatlejuic': 3,
'video': 53,
'versionget': 1,
'look': 4148,
'productrealli': 21,
'idea': 720,
'final': 870,
'product': 14559,
'outstand': 347,
'decal': 2,
'window': 56,
'everybodi': 104,
'ask': 874,
'thumb': 137,
'wow': 448,
'slickerreceiv': 1,
'shipment': 390,
'hard': 2755,
'wait': 823,
'instead': 1426,
'sticker': 28,

'remov': 387,
'easili': 867,
'daughter': 792,
'design': 286,
'sign': 170,
'print': 109,
'revers': 28,
'beauti': 657,
'shop': 697,
'program': 197,
'go': 1727,
'lot': 3340,
'fun': 812,
'everywher': 184,
'like': 21038,
'screen': 38,
'comput': 72,
'monitor': 70,
'end': 1116,
'gopher': 17,
'problemrecent': 1,
'woodstream': 5,
'corp': 7,
'lay': 92,
'easi': 3291,
'set': 850,
'work': 3881,
'success': 249,
'also': 7419,
'rememb': 709,
'wire': 40,
'attach': 72,
'tie': 50,
'steak': 367,
'prevent': 202,
'drag': 34,
'hole': 147,
'luck': 190,
'musteasi': 1,
'mess': 372,
'offer': 1116,
'vibrant': 37,
'color': 1119,
'taint': 31,
'decort': 2,
'would': 8177,
'high': 4946,
'recommend': 5101,
'anyon': 1229,
'decor': 116,
'usemuch': 1,
'easier': 560,
'wilson': 4,
'past': 980,
'frost': 367,
'simpl': 630,
'complaint': 424,
'must': 1183,
'often': 923,
'fresh': 3457,
'origin': 1051,
'master': 93,
'storytel': 4,
'burtongreat': 1,
'dont': 7132,
'even': 6484,
'know': 3610,
'sum': 42,
'first': 3756,
'complet': 732,
'unlik': 533,
'seen': 537,
'second': 1090,
'kind': 1668,
'spooki': 5,
'weird': 220,
'feel': 2242,

'probabl': 1458,
'art': 89,
'director': 25,
'welch': 15,
'anyth': 1782,
'invent': 62,
'fabul': 418,
'comed': 7,
'fanasi': 1,
'direct': 835,
'masterbeetlejuic': 1,
'amus': 31,
'romp': 2,
'explor': 33,
'incred': 498,
'possibl': 516,
'boundari': 6,
'tale': 17,
'recent': 724,
'marri': 41,
'led': 47,
'chaotic': 3,
'supernatur': 10,
'adam': 31,
'barbara': 48,
'maitland': 23,
'alec': 43,
'baldwin': 48,
'discov': 587,
'conflict': 8,
'rather': 947,
'human': 294,
'imperfect': 6,
'haunt': 18,
'plagu': 18,
'afterlif': 17,
'project': 51,
'seem': 2674,
'blind': 29,
'assign': 9,
'dispassion': 1,
'filmmak': 2,
'commerci': 211,
'reason': 1503,
'plot': 10,
'bizarr': 26,
'subject': 66,
'matter': 459,
'remark': 67,
'complement': 108,
'unusu': 180,
'macabr': 6,
'artist': 14,
'sensibl': 45,
'extraordinarili': 8,
'creat': 388,
'unbeliev': 118,
'brilliant': 54,
'guidanc': 4,
'imagin': 457,
'pee': 57,
'wee': 26,
'big': 2059,
'adventur': 59,
'batman': 24,
'wood': 109,
'sleepi': 42,
'hollow': 21,
'uniqu': 578,
'creativ': 78,
'landscap': 7,
'culmin': 2,
'essenti': 208,
'abund': 37,
'ironi': 4,
'outlandish': 4,
'yet': 1230,

```
'behavior': 41,
'grace': 53,
'bodi': 791,
'augment': 11,
'devious': 2,
'energet': 30,
'perform': 101,
'glenn': 5,
'shadix': 4,
'jeffrey': 16,
'jone': 38,
'winona': 35,
'ryder': 36,
'catherin': 20,
'hara': 2,
'bustl': 1,
'uninhibit': 1,
'hilar': 3,
'persist': 26,
'push': 121,
'level': 512,
'almost': 1885,
'affabl': 1,
'euphoria': 2,
'pair': 138,
'ingeni': 11,
'screenplay': 1,
'tour': 17,
'forc': 131,
'transform': 38,
'exuber': 1,
'jovial': 2,
'exercis': 120,
'extrem': 717,
'satisfi': 1382,
'philosoph': 1,
'percept': 14,
'though': 2308,
'unabl': 164,
'undeserv': 3,
'deep': 257,
'critic': 67,
'analysi': 17,
'undeni': 4,
'inspir': 69,
'concept': 86,
'flawless': 13,
'transfer': 49,
'dvd': 63,
'packag': 4077,
'includ': 1165,
'theatric': 9,
'trailer': 11,
'isol': 32,
'danni': 11,
'elfman': 10,
'music': 45,
'track': 103,
'choic': 1165,
'anamorph': 1,
'widescreen': 6,
'pan': 380,
'scan': 11,
'eventu': 115,
'need': 3732,
'wiltonsick': 1,
'scad': 1,
'nasti': 277,
'toothpick': 73,
'counter': 148,
'tint': 6,
'overdu': 5,
'except': 725,
'welcom': 116,
'offens': 31,
'tast': 22261,
'blend': 2168,
```

'opinion': 489,
'smooth': 1296,
'wilton': 6,
'experi': 1022,
'bit': 3489,
'hue': 17,
'right': 2837,
'worth': 1974,
'funnithought': 1,
'geeki': 1,
'husband': 1315,
'priceless': 6,
'scenc': 1,
'kid': 2375,
'successmani': 1,
'dealt': 18,
'figur': 483,
'angel': 71,
'clarenc': 1,
'life': 1075,
'brad': 4,
'pitt': 3,
'abysm': 4,
'joe': 276,
'black': 2233,
'howev': 2360,
'present': 381,
'interestng': 1,
'town': 196,
'die': 252,
'oddest': 1,
'return': 516,
'thier': 37,
'selv': 5,
'confin': 11,
'premis': 9,
'send': 474,
'strang': 245,
'desert': 112,
'popul': 20,
'remain': 313,
'wouldnt': 762,
'bad': 2126,
'wasnt': 809,
'nyc': 70,
'artisan': 21,
'move': 400,
'remodel': 4,
'place': 1380,
'consult': 38,
'juno': 3,
'sylvia': 19,
'sidney': 5,
'hire': 15,
'betelgues': 3,
'pronouc': 2,
'henc': 80,
'titl': 79,
'unfortun': 548,
'po': 1,
'demon': 7,
'refus': 155,
'underworld': 1,
'visual': 39,
'stun': 31,
'micheal': 9,
'give': 4357,
'blast': 62,
'beer': 273,
'swill': 13,
'cuss': 1,
'worst': 286,
'nightmar': 23,
'superpow': 1,
'proud': 72,
'martha': 9,
'stewart': 19,

'style': 395,
'cake': 1425,
'cookidont': 2,
'liquid': 585,
'food': 8617,
'intens': 283,
'shade': 49,
'ice': 1676,
'season': 1340,
'cooki': 3068,
'excit': 470,
'tabl': 357,
'dessert': 466,
'parti': 445,
'tea': 18516,
'event': 78,
'especi': 1651,
'contest': 19,
'birthday': 287,
'christma': 825,
'raini': 23,
'afternoon': 359,
'casper': 4,
'ghostmichael': 1,
'bring': 767,
'distinguish': 49,
'characterist': 37,
'ghoul': 3,
'mere': 69,
'script': 8,
'cinematographi': 1,
'focus': 92,
'snack': 4107,
'popcorn': 1010,
'terribl': 396,
'dvdcontinu': 1,
'amaz': 1292,
'shoddi': 3,
'treatment': 58,
'releas': 115,
'simpli': 961,
'disgrac': 4,
'exampl': 167,
'energi': 977,
'extra': 1471,
'mention': 519,
'delet': 16,
'scene': 42,
'featurett': 2,
'lousi': 41,
'commentari': 18,
'wors': 205,
'cut': 1012,
'less': 2500,
'theater': 105,
'advic': 92,
'money': 1227,
'somebodi': 42,
'capabl': 17,
'medium': 478,
'provid': 713,
'materi': 103,
'greatmovi': 1,
'cheat': 38,
'dvdwarn': 1,
'trick': 191,
'format': 27,
'compani': 1927,
'mistak': 207,
'full': 1619,
'compar': 1356,
'comtain': 1,
'pictur': 372,
'top': 1949,
'bottom': 482,
'mean': 882,
'take': 3237,

'slight': 1256,
'expect': 1457,
'care': 1265,
'rip': 103,
'peopl': 2229,
'nice': 4101,
'bright': 169,
'colorbought': 3,
'dia': 3,
'los': 26,
'muerto': 1,
'skull': 8,
'ateco': 4,
'gave': 1020,
'mix': 5873,
'total': 886,
'beatlejuicwinona': 1,
'hot': 3800,
'gothic': 8,
'princess': 16,
'doom': 7,
'eldest': 6,
'brother': 153,
'ive': 5307,
'zenith': 1,
'goe': 906,
'miss': 656,
'cours': 865,
'fan': 995,
'fav': 45,
'part': 1025,
'chum': 2,
'charact': 98,
'wear': 67,
'type': 1245,
'cloth': 71,
'deal': 1242,
'witti': 4,
'clever': 38,
'charm': 81,
'spit': 135,
'hit': 702,
'gina': 4,
'favoritfavorit': 8,
'mine': 631,
'girl': 204,
'read': 1192,
'mom': 510,
'bedtim': 97,
'copi': 41,
'either': 1108,
'lost': 337,
'given': 619,
'away': 1443,
'found': 4435,
'garag': 24,
'sale': 553,
'condit': 469,
'around': 1744,
'somewher': 134,
'attic': 1,
'niec': 36,
'piec': 1335,
'heavenyear': 2,
'simul': 10,
'truli': 663,
'italian': 590,
'espresso': 933,
'cappuccino': 111,
'without': 3257,
'sever': 1798,
'arriv': 1643,
'sent': 592,
'briel': 4,
'cadiz': 1,
'machin': 906,
'russian': 37,

'china': 327,
'cup': 4502,
'case': 2065,
'espression': 14,
'pod': 1309,
'kindest': 1,
'qualiti': 2900,
'eas': 162,
'exquisit': 76,
'prepar': 928,
'gift': 1956,
'itali': 212,
'heaven': 391,
'earth': 246,
'everyonvacat': 1,
'meet': 193,
'demis': 4,
'kill': 165,
'local': 2659,
'worri': 547,
'head': 242,
'transport': 58,
'mysteri': 85,
'journey': 26,
'roller': 4,
'coaster': 4,
'score': 58,
'til': 31,
'cri': 68,
'odd': 207,
'fantasi': 7,
'comeditwo': 1,
'accid': 67,
'aliv': 58,
'ghost': 43,
'stuck': 219,
'etern': 6,
'month': 2042,
'york': 125,
'jefferi': 3,
'connecticut': 5,
'farmhous': 7,
'chang': 1111,
'aesthet': 19,
'control': 373,
'terrif': 336,
'box': 5507,
'offic': 413,
'summer': 420,
'plenti': 337,
'humor': 25,
'win': 112,
'oscar': 20,
'likeabl': 4,
'age': 393,
'gross': 165,
'funmichael': 1,
'alreadi': 642,
'major': 304,
'solidifi': 9,
'status': 31,
'hollywood': 12,
'brightest': 2,
'generat': 72,
'belli': 339,
'wife': 620,
'surpris': 1373,
'centerpiec': 6,
'gag': 78,
'defin': 243,
'begin': 290,
'ad': 2652,
'hilarihilari': 1,
'favorit': 4289,
'dine': 54,
'room': 364,
'danc': 54,

'rider': 5,
'spectacular': 53,
'couldnt': 914,
'stop': 1093,
'keep': 3491,
'stitch': 6,
'night': 833,
'cartoon': 26,
'shown': 109,
'decid': 1037,
'fact': 1313,
'hey': 126,
'much': 7109,
'thing': 3863,
'wrong': 739,
'short': 420,
'mayb': 1010,
'efect': 1,
'older': 166,
'middl': 229,
'nameset': 1,
'england': 200,
'follow': 639,
'maitlin': 1,
'owner': 193,
'larg': 1318,
'hardwar': 9,
'vacat': 128,
'drown': 23,
'mishap': 2,
'taken': 245,
'boister': 4,
'deet': 4,
'prospect': 11,
'poltergeist': 1,
'featur': 159,
'cast': 36,
'young': 237,
'storylin': 3,
'boast': 32,
'believ': 1045,
'prop': 8,
'costum': 3,
'absolut': 1509,
'liter': 200,
'diebeetlejuic': 1,
'funiest': 1,
'realiti': 33,
'romanc': 4,
'deceas': 12,
'handbook': 6,
'break': 547,
'cover': 632,
'job': 364,
'monthbook': 1,
'poetri': 12,
'cute': 196,
'poem': 12,
'author': 77,
'purpos': 222,
'write': 311,
'rhythm': 8,
'thingwine': 1,
'saver': 153,
'obvious': 276,
'open': 1875,
'bottl': 1919,
'wine': 400,
'stay': 815,
'glass': 611,
'dinner': 672,
'let': 1341,
'expens': 1659,
'sinc': 3493,
'wont': 1659,
'wast': 618,
'rest': 584

    'rest' : 504,
    'interest': 710,
    'undrunk': 1,
    'portion': 427,
    'next': 1133,
    'put': 2655,
    'fridg': 346,
    'drink': 6390,
    'pull': 254,
    'red': 1445,
    'guynever': 1,
    'dissapoint': 84,
    'gadget': 23,
    'concerto': 4,
    'white': 1759,
    'stopper': 14,
    'handi': 281,
    'click': 60,
    'sound': 418,
    'vacuum': 80,
    'correct': 245,
    'neat': 85,
    'beetlejuicsay': 1,
    'actual': 2253,
    'memor': 24,
    'drama': 7,
    'pervert': 3,
    'caus': 566,
    'quit': 2288,
    'chao': 5,
    'newli': 39,
    'couldv': 11,
    'better': 6190,
    'disapoint': 29,
    'beetljuic': 1,
    'footag': 2,
    'havent': 804,
    'lover': 605,
    'starbuck': 597,
    'poduse': 6,
    'chamonix': 3,
    'amazon': 7130,
    'produc': 564,
    'delici': 4775,
    'latt': 271,
    'smell': 1794,
    'regular': 2648,
    'decaf': 643,
    'dark': 1624,
    'rich': 1494,
    'roast': 1525,
    'fyi': 36,
    'paper': 327,
    'tab': 56,
    'burn': 297,
    'finger': 218,
    'individu': 690,
    'wrap': 555,
    'space': 140,
    'your': 1494,
    'stamp': 43,
    'date': 731,
    'ghostsimpli': 1,
    'funniest': 9,
    'ghostbust': 1,
    'mischief': 2,
    'rid': 120,
    'decemb': 62,
    'snowman': 4,
    'anniversaridaughter': 1,
    'rosi': 11,
    'carol': 12,
    'king': 92,
    'avail': 1806,
    'far': 2161,
    'johnni': 11,
    'allig': 5,
    'chicken': 2332

```
'soup': 2004,
'rice': 2901,
'mauric': 4,
'sendak': 11,
'plus': 1140,
'cheap': 459,
'movimovi': 2,
'doesnt': 2562,
'lydia': 22,
'deitz': 1,
'sad': 221,
'depress': 37,
'mother': 462,
'redecor': 1,
'ohara': 18,
'other': 1217,
'march': 51,
'blooper': 1,
'workskeptic': 2,
'item': 2090,
'prior': 120,
'vacuvin': 6,
'last': 2416,
'maximum': 30,
'immedi': 474,
'longer': 1091,
'refriger': 507,
'week': 1581,
'devic': 31,
'pump': 246,
'plastic': 676,
'cork': 5,
'pressur': 141,
'rins': 170,
'perfect': 3668,
'singl': 687,
'sit': 703,
'switch': 969,
'betterive': 5,
'vacu': 12,
'vin': 13,
'previous': 375,
...}
```

In [11]:

```python
type(word_dist)
```

Out[11]:

```
dict
```

In [12]:

```python
import operator
word_dist = sorted(word_dist.items(), key=operator.itemgetter(1),reverse = True)
word_dist
```

Out[12]:

```
[('tast', 22261),
 ('like', 21038),
 ('great', 20108),
 ('good', 19175),
 ('tea', 18516),
 ('flavor', 17186),
 ('one', 14636),
 ('product', 14559),
 ('use', 13775),
 ('love', 13563),
 ('tri', 12326),
 ('make', 11801),
 ('get', 10526),
 ('best', 9955),
```

```
('coffe', 9101),
('food', 8617),
('eat', 8302),
('would', 8177),
('time', 7991),
('buy', 7941),
('find', 7437),
('also', 7419),
('realli', 7276),
('dont', 7132),
('amazon', 7130),
('much', 7109),
('littl', 6732),
('even', 6484),
('order', 6443),
('drink', 6390),
('price', 6195),
('better', 6190),
('chocol', 6159),
('bag', 6102),
('well', 6066),
('mix', 5873),
('sugar', 5827),
('store', 5725),
('box', 5507),
('sweet', 5473),
('dog', 5332),
('ive', 5307),
('recommend', 5101),
('year', 4976),
('high', 4946),
('day', 4929),
('delici', 4775),
('treat', 4696),
('cup', 4502),
('found', 4435),
('want', 4400),
('give', 4357),
('enjoy', 4352),
('favorit', 4289),
('way', 4247),
('add', 4192),
('water', 4162),
('look', 4148),
('snack', 4107),
('nice', 4101),
('packag', 4077),
('made', 4037),
('brand', 4006),
('think', 3996),
('wonder', 3933),
('work', 3881),
('thing', 3863),
('milk', 3854),
('hot', 3800),
('first', 3756),
('need', 3732),
('organ', 3678),
('perfect', 3668),
('excel', 3647),
('free', 3640),
('two', 3623),
('know', 3610),
('mani', 3609),
('healthi', 3572),
('differ', 3570),
('come', 3563),
('cat', 3559),
('natur', 3545),
('ever', 3501),
('sinc', 3493),
('keep', 3491),
('bit', 3489),
('ship', 3475),
('fresh', 3457),
('could', 3430),
('pack', 3380),
```

```
('lot', 3340),
('sauc', 3329),
('easi', 3291),
('cant', 3287),
('say', 3259),
('without', 3257),
('take', 3237),
('still', 3215),
('bar', 3201),
('review', 3128),
('purchas', 3121),
('green', 3103),
('cook', 3103),
('cooki', 3068),
('tasti', 3066),
('stuff', 3064),
('never', 3063),
('ingredi', 3041),
('everi', 2989),
('rice', 2901),
('qualiti', 2900),
('alway', 2882),
('oil', 2847),
('candi', 2847),
('right', 2837),
('whole', 2831),
('contain', 2801),
('cereal', 2793),
('enough', 2781),
('juic', 2760),
('hard', 2755),
('someth', 2730),
('salt', 2705),
('seem', 2674),
('local', 2659),
('fruit', 2655),
('put', 2655),
('ad', 2652),
('regular', 2648),
('serv', 2602),
('bought', 2571),
('doesnt', 2562),
('quick', 2550),
('dri', 2533),
('back', 2525),
('less', 2500),
('calori', 2498),
('chip', 2484),
('small', 2476),
('got', 2474),
('real', 2442),
('last', 2416),
('chees', 2411),
('long', 2381),
('kid', 2375),
('howev', 2360),
('chicken', 2332),
('though', 2308),
('quit', 2288),
('groceri', 2288),
('size', 2280),
('famili', 2271),
('strong', 2263),
('actual', 2253),
('help', 2242),
('feel', 2242),
('thank', 2239),
('black', 2233),
('peopl', 2229),
('bread', 2214),
('fat', 2206),
('old', 2204),
('see', 2190),
('textur', 2187),
('blend', 2168),
('didnt', 2163),
('far', 2161),
```

```
('sure', 2130),
('bad', 2126),
('diet', 2119),
('low', 2096),
('definit', 2091),
('item', 2090),
('case', 2065),
('big', 2059),
('friend', 2049),
('month', 2042),
('bean', 2026),
('problem', 2007),
('soup', 2004),
('can', 1999),
('anoth', 1992),
('new', 1984),
('light', 1979),
('worth', 1974),
('gluten', 1959),
('gift', 1956),
('top', 1949),
('compani', 1927),
('butter', 1927),
('bottl', 1919),
('start', 1915),
('pretti', 1911),
('may', 1889),
('almost', 1885),
('open', 1875),
('pasta', 1859),
('bake', 1854),
('theyr', 1853),
('varieti', 1840),
('recip', 1834),
('avail', 1806),
('amount', 1805),
('spice', 1805),
('sever', 1798),
('usual', 1797),
('smell', 1794),
('anyth', 1782),
('thought', 1767),
('minut', 1764),
('happi', 1760),
('white', 1759),
('health', 1755),
('around', 1744),
('meal', 1743),
('per', 1741),
('go', 1727),
('chew', 1716),
('breakfast', 1699),
('star', 1698),
('ice', 1676),
('that', 1674),
('kind', 1668),
('expens', 1659),
('wont', 1659),
('especi', 1651),
('home', 1648),
('arriv', 1643),
('morn', 1628),
('dark', 1624),
('nut', 1621),
('soda', 1621),
('full', 1619),
('protein', 1603),
('wheat', 1599),
('prefer', 1595),
('week', 1581),
('leav', 1575),
('vanilla', 1574),
('corn', 1533),
('powder', 1528),
('roast', 1525),
('fill', 1517),
('hand', 1514),
```

```
('absolut', 1509),
('reason', 1503),
('rich', 1494),
('your', 1494),
('yummi', 1487),
('pleas', 1482),
('might', 1481),
('extra', 1471),
('bitter', 1466),
('syrup', 1465),
('probabl', 1458),
('expect', 1457),
('abl', 1449),
('red', 1445),
('away', 1443),
('carri', 1435),
('half', 1434),
('cream', 1430),
('save', 1428),
('instead', 1426),
('cake', 1425),
('peanut', 1424),
('honey', 1424),
('disappoint', 1423),
('came', 1422),
('cracker', 1407),
('cost', 1405),
('live', 1401),
('market', 1393),
('noth', 1388),
('receiv', 1384),
('satisfi', 1382),
('place', 1380),
('call', 1379),
('surpris', 1373),
('altern', 1370),
('youll', 1369),
('brew', 1367),
('compar', 1356),
('said', 1353),
('sweeten', 1346),
('let', 1341),
('season', 1340),
('piec', 1335),
('fine', 1331),
('meat', 1322),
('larg', 1318),
('cherri', 1317),
('three', 1316),
('husband', 1315),
('fact', 1313),
('pod', 1309),
('orang', 1303),
('ill', 1302),
('smooth', 1296),
('amaz', 1292),
('pepper', 1282),
('appl', 1274),
('care', 1265),
('fast', 1264),
('slight', 1256),
('salad', 1253),
('type', 1245),
('deal', 1242),
('spici', 1237),
('yet', 1230),
('anyon', 1229),
('money', 1227),
('other', 1217),
('least', 1217),
('addit', 1205),
('isnt', 1201),
('coconut', 1200),
('fiber', 1194),
('read', 1192),
('must', 1183),
('coupl', 1181),
```

```
('hope', 1171),
('includ', 1165),
('choic', 1165),
('almond', 1164),
('ago', 1163),
('everyon', 1155),
('wish', 1152),
('person', 1143),
('oatmeal', 1143),
('hous', 1141),
('plus', 1140),
('oliv', 1136),
('next', 1133),
('although', 1123),
('dish', 1121),
('crunchi', 1121),
('color', 1119),
('end', 1116),
('offer', 1116),
('chang', 1111),
('etc', 1109),
('cold', 1109),
('either', 1108),
('lemon', 1107),
('valu', 1107),
('jar', 1105),
('version', 1100),
('tell', 1099),
('special', 1098),
('turn', 1096),
('everyth', 1096),
('stop', 1093),
('longer', 1091),
('second', 1090),
('els', 1085),
('cinnamon', 1083),
('noodl', 1083),
('soft', 1077),
('life', 1075),
('soy', 1071),
('side', 1070),
('tomato', 1061),
('mouth', 1058),
('origin', 1051),
('flour', 1047),
('list', 1046),
('believ', 1045),
('gum', 1043),
('decid', 1037),
('son', 1036),
('run', 1034),
('part', 1025),
('experi', 1022),
('gave', 1020),
('brown', 1016),
('addict', 1013),
('cut', 1012),
('popcorn', 1010),
('mayb', 1010),
('nutrit', 1008),
('stock', 1006),
('went', 1005),
('conveni', 1005),
('seed', 1003),
('glad', 997),
('fan', 995),
('sometim', 993),
('sell', 988),
('awesom', 987),
('packet', 987),
('past', 980),
('energi', 977),
('stick', 976),
('hour', 970),
('ounc', 970),
('switch', 969),
('strawberri', 968),
```

```
('weight', 968),
('vitamin', 967),
('bowl', 962),
('simpli', 961),
('notic', 961),
('heat', 948),
('rather', 947),
('potato', 943),
('plain', 940),
('carb', 940),
('espresso', 933),
('grain', 931),
('pop', 930),
('prepar', 928),
('often', 923),
('mild', 918),
('cheaper', 915),
('couldnt', 914),
('ginger', 913),
('goe', 906),
('machin', 906),
('servic', 906),
('pill', 904),
('artifici', 902),
('normal', 898),
('onlin', 897),
('microwav', 896),
('pound', 895),
('refresh', 893),
('cocoa', 889),
('instant', 888),
('bite', 887),
('total', 886),
('fantast', 885),
('gummi', 885),
('base', 883),
('mean', 882),
('chai', 878),
('ask', 874),
('carbon', 874),
('consist', 874),
('took', 873),
('egg', 873),
('final', 870),
('easili', 867),
('cours', 865),
('super', 865),
('world', 864),
('herb', 864),
('close', 858),
('substitut', 855),
('exact', 851),
('set', 850),
('beef', 848),
('pay', 847),
('lunch', 844),
('salti', 843),
('pot', 838),
('label', 837),
('caffein', 836),
('direct', 835),
('night', 833),
('christma', 825),
('wait', 823),
('aroma', 823),
('result', 819),
('stay', 815),
('fun', 812),
('process', 812),
('wasnt', 809),
('replac', 808),
('berri', 807),
('clean', 806),
('havent', 804),
('sodium', 802),
('benefit', 797),
('ground', 797),
```

```
('delight', 796),
('particular', 794),
('combin', 793),
('daughter', 792),
('bodi', 791),
('cannot', 789),
('pleasant', 787),
('near', 787),
('name', 784),
('licoric', 784),
('becom', 778),
('effect', 777),
('fair', 775),
('line', 773),
('feed', 771),
('consid', 769),
('bring', 767),
('wouldnt', 762),
('suggest', 760),
('state', 758),
('rate', 756),
('veget', 754),
('dress', 753),
('fish', 751),
('beverag', 749),
('similar', 749),
('search', 747),
('certain', 745),
('babi', 741),
('grey', 740),
('wrong', 739),
('pancak', 738),
('daili', 738),
('chewi', 735),
('complet', 732),
('date', 731),
('four', 728),
('bulk', 728),
('except', 725),
('com', 725),
('recent', 724),
('bold', 721),
('idea', 720),
('extrem', 717),
('content', 717),
('custom', 717),
('mint', 717),
('check', 717),
('pick', 714),
('provid', 713),
('interest', 710),
('rememb', 709),
('roll', 707),
('true', 707),
('pet', 706),
('toy', 704),
('sit', 703),
('hit', 702),
('along', 700),
('formula', 700),
('there', 699),
('preserv', 699),
('shop', 697),
('gram', 695),
('pure', 694),
('chili', 691),
('individu', 690),
('singl', 687),
('deliveri', 684),
('smaller', 682),
('oat', 682),
('deliv', 681),
('teeth', 678),
('tradit', 676),
('plastic', 676),
('finish', 676),
('eaten', 675),
```

```
('watch', 674),
('coat', 673),
('dinner', 672),
('entir', 672),
('senseo', 670),
('creami', 666),
('earl', 666),
('someon', 665),
('grow', 664),
('truli', 663),
('healthier', 662),
('continu', 662),
('beat', 661),
('garlic', 659),
('note', 659),
('crunch', 658),
('beauti', 657),
('miss', 656),
('french', 650),
('acid', 650),
('point', 647),
('guess', 646),
('area', 644),
('decaf', 643),
('jerki', 643),
('overal', 643),
('alreadi', 642),
('supermarket', 642),
('follow', 639),
('insid', 638),
('flower', 636),
('sourc', 634),
('raw', 633),
('hint', 633),
('cover', 632),
('share', 632),
('mine', 631),
('vegan', 631),
('simpl', 630),
('later', 626),
('five', 626),
('togeth', 625),
('restaur', 623),
('plant', 623),
('huge', 622),
('wife', 620),
('given', 619),
('wast', 618),
('loos', 618),
('glass', 611),
('shape', 611),
('boil', 610),
('soon', 609),
('larger', 609),
('consum', 608),
('pour', 607),
('raisin', 606),
('gone', 605),
('lover', 605),
('slice', 603),
('left', 603),
('hold', 602),
('dip', 601),
('sold', 598),
('veggi', 598),
('starbuck', 597),
('import', 596),
('yes', 595),
('allergi', 594),
('general', 593),
('melt', 593),
('done', 592),
('sent', 592),
('gourmet', 592),
('ate', 591),
('sour', 591),
('italian', 590),
```

```
('mind', 590),
('aftertast', 590),
('granola', 588),
('discov', 587),
('arent', 587),
('impress', 586),
('liquid', 585),
('yum', 585),
('rest', 584),
('balanc', 582),
('children', 581),
('warm', 581),
('uniqu', 578),
('bear', 578),
('anyway', 577),
('within', 574),
('difficult', 574),
('thick', 571),
('toast', 568),
('caus', 566),
('thin', 566),
('fri', 565),
('produc', 564),
('onion', 561),
('easier', 560),
('due', 560),
('crave', 559),
('throw', 558),
('busi', 556),
('lower', 556),
('wrap', 555),
('sale', 553),
('picki', 553),
('maker', 551),
('unfortun', 548),
('worri', 547),
('break', 547),
('cool', 543),
('option', 542),
('kitchen', 540),
('pie', 540),
('clear', 538),
('stomach', 538),
('seen', 537),
('tuna', 537),
('vet', 535),
('cashew', 535),
('unlik', 533),
('drop', 532),
('told', 530),
('site', 527),
('saw', 526),
('crisp', 526),
('blue', 525),
('weve', 525),
('diabet', 522),
('english', 520),
('allow', 520),
('mention', 519),
('stir', 519),
('sparkl', 518),
('plan', 517),
('possibl', 516),
('return', 516),
('tart', 515),
('requir', 514),
('fit', 514),
('level', 512),
('websit', 512),
('anywher', 511),
('mom', 510),
('browni', 510),
('refriger', 507),
('avoid', 506),
('today', 505),
('suppli', 504),
('drinker', 504),
```

('yogurt', 504),
('hook', 500),
('incred', 498),
('american', 495),
('sandwich', 495),
('sea', 495),
('bone', 493),
('pocket', 493),
('izz', 492),
('hate', 490),
('tin', 490),
('opinion', 489),
('basic', 489),
('raspberri', 489),
('suppos', 488),
('issu', 488),
('form', 487),
('happen', 487),
('appreci', 485),
('twice', 485),
('unless', 485),
('digest', 485),
('none', 484),
('splenda', 484),
('figur', 483),
('concentr', 483),
('banana', 483),
('bottom', 482),
('moist', 482),
('youv', 479),
('leaf', 479),
('medium', 478),
('word', 474),
('send', 474),
('immedi', 474),
('shake', 474),
('manufactur', 473),
('herbal', 473),
('tend', 472),
('excit', 470),
('condit', 469),
('number', 469),
('six', 468),
('grape', 468),
('mill', 467),
('dessert', 466),
('homemad', 466),
('muffin', 466),
('batch', 466),
('ball', 463),
('describ', 463),
('mother', 462),
('subtl', 462),
('count', 461),
('blueberri', 461),
('touch', 460),
('vegetarian', 460),
('matter', 459),
('cheap', 459),
('imagin', 457),
('test', 457),
('trip', 456),
('agre', 455),
('remind', 455),
('stevia', 451),
('steep', 450),
('seal', 450),
('standard', 449),
('sampl', 449),
('concern', 449),
('instruct', 449),
('wow', 448),
('sort', 448),
('tree', 448),
('anymor', 448),
('chemic', 447),
('improv', 446),

```
('improv', 446),
('train', 446),
('parti', 445),
('show', 444),
('sprinkl', 444),
('stale', 444),
('expir', 439),
('extract', 439),
('main', 439),
('stand', 437),
('jasmin', 437),
('blood', 436),
('quinoa', 436),
('perhap', 432),
('readi', 432),
('kick', 432),
('system', 431),
('portion', 427),
('jelli', 426),
('vinegar', 426),
('pretzel', 426),
('book', 424),
('complaint', 424),
('premium', 422),
('summer', 420),
('short', 420),
('broth', 420),
('fabul', 418),
('sound', 418),
('broken', 416),
('tini', 414),
('bland', 414),
('offic', 413),
('mustard', 413),
('typic', 411),
('biscuit', 411),
('travel', 410),
('whatev', 409),
('otherwis', 409),
('flake', 409),
('doubl', 406),
('over', 405),
('movi', 402),
('caramel', 402),
('overpow', 401),
('move', 400),
('wine', 400),
('shelf', 400),
('visit', 400),
('lose', 398),
('twine', 398),
('power', 397),
('puppi', 397),
('heart', 397),
('terribl', 396),
('learn', 395),
('style', 395),
('garden', 394),
('age', 393),
('heavi', 392),
('child', 391),
('heaven', 391),
('shipment', 390),
('spread', 389),
('salmon', 389),
('seller', 389),
('creat', 388),
('most', 388),
('crazi', 387),
('remov', 387),
('mac', 386),
('crispi', 385),
('brought', 383),
('school', 382),
('smoke', 382),
('present', 381),
('tablespoon', 381),
('inform', 381)
```

('pan', 380),
('tazo', 380),
('root', 379),
('quantiti', 378),
('thrill', 377),
('spend', 376),
('previous', 375),
('adult', 375),
('select', 374),
('kraft', 374),
('numi', 374),
('control', 373),
('mountain', 373),
('mess', 372),
('pictur', 372),
('chop', 372),
('decent', 372),
('holiday', 372),
('delic', 372),
('lack', 371),
('poor', 371),
('salsa', 371),
('higher', 370),
('serious', 369),
('somewhat', 369),
('kept', 368),
('steak', 367),
('frost', 367),
('sick', 367),
('fall', 367),
('pantri', 367),
('weak', 367),
('straight', 366),
('forward', 366),
('room', 364),
('job', 364),
('claim', 363),
('cholesterol', 363),
('skin', 363),
('alon', 362),
('horribl', 362),
('reduc', 362),
('nutti', 361),
('dollar', 361),
('yeast', 361),
('hazelnut', 360),
('curri', 360),
('afternoon', 359),
('realiz', 358),
('tip', 358),
('tabl', 357),
('countri', 357),
('web', 357),
('nutriti', 357),
('comment', 356),
('known', 355),
('play', 355),
('spoon', 355),
('celiac', 354),
('pass', 352),
('grill', 352),
('winter', 352),
('everyday', 351),
('knew', 350),
('charg', 350),
('peach', 350),
('pizza', 349),
('across', 348),
('pricey', 348),
('outstand', 347),
('fridg', 346),
('safe', 346),
('stronger', 344),
('cheddar', 344),
('gold', 343),
('sip', 342),
('bes', 341)

```
( 'ies , 541),
('frozen', 341),
('belli', 339),
('wild', 339),
('popular', 339),
('mapl', 339),
('felt', 338),
('kosher', 338),
('lost', 337),
('plenti', 337),
('teaspoon', 337),
('superior', 337),
('terrif', 336),
('indian', 336),
('lipton', 336),
('futur', 336),
('lime', 335),
('appear', 333),
('relat', 333),
('chanc', 332),
('chunk', 332),
('lid', 332),
('agav', 332),
('dairi', 331),
('haribo', 331),
('various', 330),
('bud', 330),
('prompt', 329),
('outsid', 329),
('shell', 329),
('coke', 329),
('kashi', 329),
('classic', 328),
('fix', 328),
('china', 327),
('paper', 327),
('troubl', 327),
('late', 326),
('boy', 326),
('aw', 326),
('crust', 325),
('kit', 324),
('descript', 324),
('door', 324),
('grown', 323),
('assort', 323),
('chines', 323),
('heard', 323),
('thai', 322),
('handl', 322),
('eater', 322),
('guest', 321),
('allerg', 321),
('warn', 320),
('sticki', 320),
('increas', 320),
('fruiti', 320),
('freez', 320),
('bob', 319),
('introduc', 318),
('gravi', 318),
('keurig', 317),
('remain', 313),
 ...]
```

In [13]:

```
type(word_dist)
```

Out[13]:

```
list
```

In [14]:

```
word dist[:15]
```

```
word_dist[:15]
```

```
[('tast', 22261),
 ('like', 21038),
 ('great', 20108),
 ('good', 19175),
 ('tea', 18516),
 ('flavor', 17186),
 ('one', 14636),
 ('product', 14559),
 ('use', 13775),
 ('love', 13563),
 ('tri', 12326),
 ('make', 11801),
 ('get', 10526),
 ('best', 9955),
 ('coffe', 9101)]
```

In [0]:

```python
#https://stackoverflow.com/questions/3071415/efficient-method-to-calculate-the-rank-vector-of-a-list-in-python


'''a={}
  rank=1
  for num in sorted(vector):
    if num not in a:
      a[num]=rank
      rank=rank+1 '''

a = {}
rank = 1
for num in range(len(word_dist)):
    i = word_dist[num][0]
    a[i] = rank
    rank+=1
```

In [0]:

```python
X = []
for sent in z:
    rows = []
    for word in sent.split():
        rows.append(a[word])
    X.append(rows)
```

In [0]:

```python
#X[:20]
```

In [0]:

```python
Y = final['Score']
```

In [0]:

```python
from sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split(X,Y,test_size = 0.2,random_state = 0,shuffle = False)
```

In [0]:

```python
%matplotlib notebook
%matplotlib inline
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 [0]:

```
import warnings
plt.style.use('fivethirtyeight')
plt.rcParams['figure.figsize'] = [10, 5]
warnings.filterwarnings("ignore", category=FutureWarning)
%config InlineBackend.figure_format = 'retina'
```

In [22]:

```
# truncate and/or pad input sequences
max_review_length = 600
X_train = sequence.pad_sequences(X_train, maxlen=max_review_length)
X_test = sequence.pad_sequences(X_test, maxlen=max_review_length)

print(X_train.shape)
print(X_train[1])
```

```
(40000, 600)
[    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0    0    0
```

```
       0        0        0        0        0        0        0        0        0        0        0        0
     117     2782        3       55      867       19    21192      164      787     1098     5421       44
     327      844     1229      593      132       46     3471      123        9     2782      805     3568
    1978     2969     4566      352      117      857      102      535      805      641      550      273]
```

In [0]:

```
top_words = 5000
epochs = 10
batch_size = 64
```

RELU activation with less layers

In [30]:

```
# create the model
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length, input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='relu'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
#Refer: https://datascience.stackexchange.com/questions/10615/number-of-parameters-in-an-lstm-mode
l
```

```
_____
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 600, 32)           160000
_____
lstm_3 (LSTM)                (None, 100)               53200
_____
dense_3 (Dense)              (None, 1)                 101
=================================================================
Total params: 213,301
Trainable params: 213,301
Non-trainable params: 0
_____
None
```

In [0]:

```
from datetime import datetime
```

In [31]:

```
start = datetime.now()


history = model.fit(X_train, y_train, epochs = epochs, batch_size = batch_size, verbose=1,validatio
n_data=(X_test, y_test))
scores = model.evaluate(X_test, y_test, verbose = 0)
print("Accuracy: %.2f%%" % (scores[1]*100))


score = model.evaluate(X_test, y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# list of epoch numbers
x = list(range(1,epochs +1))

# print(history.history.keys())
# dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
# history = model_drop.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, va
lidation_data=(X_test, Y_test))

# we will get val_loss and val_acc only when you pass the paramter validation_data
```

```
# val_loss : validation loss
# val_acc : validation accuracy

# loss : training loss
# acc : train accuracy
# for each key in histrory.histrory we will have a list of length equal to number of epochs

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)


print('Time taken to run this cell :', datetime.now() - start)
```

```
Train on 40000 samples, validate on 10000 samples
Epoch 1/10
40000/40000 [==============================] - 753s 19ms/step - loss: 14.3317 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 2/10
40000/40000 [==============================] - 751s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 3/10
40000/40000 [==============================] - 748s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 4/10
40000/40000 [==============================] - 745s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 5/10
40000/40000 [==============================] - 748s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 6/10
40000/40000 [==============================] - 750s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 7/10
40000/40000 [==============================] - 748s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 8/10
40000/40000 [==============================] - 751s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 9/10
40000/40000 [==============================] - 747s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Epoch 10/10
40000/40000 [==============================] - 748s 19ms/step - loss: 14.3370 - acc: 0.1105 - val_
loss: 14.1597 - val_acc: 0.1215
Accuracy: 12.15%
Test score: 14.159729119873047
Test accuracy: 0.1215
Time taken to run this cell : 2:09:18.053135
```
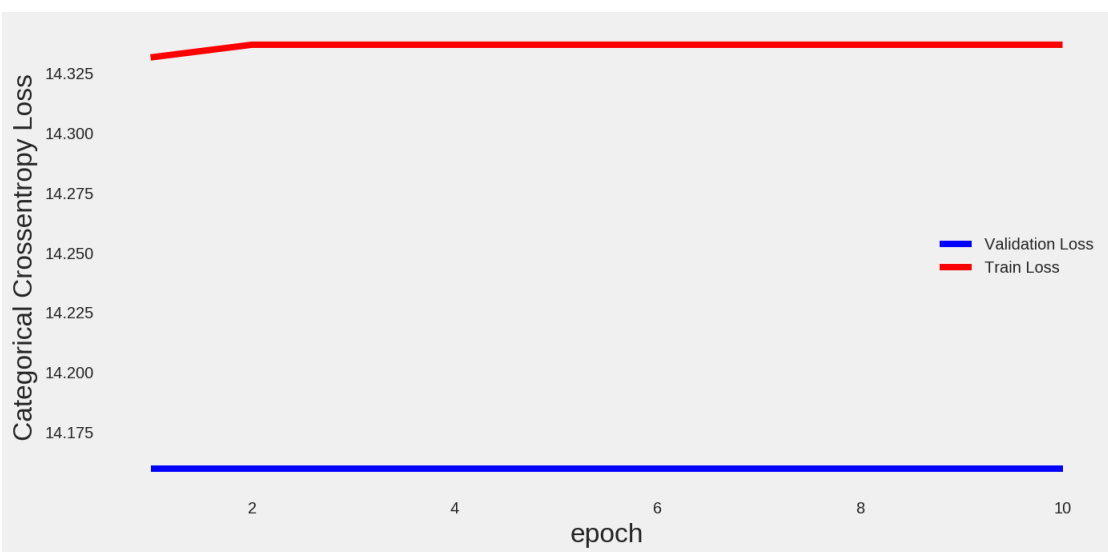

Sigmoid

In [35]:

```
# create the model
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length, input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
#Refer: https://datascience.stackexchange.com/questions/10615/number-of-parameters-in-an-lstm-mode
l
```

```
_____
Layer (type)                 Output Shape              Param #
=================================================================
embedding_6 (Embedding)      (None, 600, 32)           160000
_____
lstm_7 (LSTM)                (None, 100)               53200
_____
dense_6 (Dense)              (None, 1)                 101
=================================================================
Total params: 213,301
Trainable params: 213,301
Non-trainable params: 0
_____
None
```

In [37]:

```
start = datetime.now()


history = model.fit(X_train, y_train, epochs = epochs, batch_size = batch_size, verbose=1,validatio
n_data=(X_test, y_test))
scores = model.evaluate(X_test, y_test, verbose = 0)
print("Accuracy: %.2f%%" % (scores[1]*100))


score = model.evaluate(X_test, y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# list of epoch numbers
x = list(range(1,epochs +1))

# print(history.history.keys())
# dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
# history = model_drop.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, va
lidation_data=(X_test, Y_test))

# we will get val_loss and val_acc only when you pass the paramter validation_data
# val_loss : validation loss
# val_acc : validation accuracy

# loss : training loss
# acc : train accuracy
# for each key in histrory.histrory we will have a list of length equal to number of epochs

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)


print('Time taken to run this cell :', datetime.now() - start)
```

```
Train on 40000 samples, validate on 10000 samples
Epoch 1/10
40000/40000 [==============================] - 764s 19ms/step - loss: 0.3455 - acc: 0.8897 - val_l
oss: 0.3673 - val_acc: 0.8788
Epoch 2/10
40000/40000 [==============================] - 758s 19ms/step - loss: 0.3440 - acc: 0.8896 - val_l
oss: 0.3655 - val_acc: 0.8785
Epoch 3/10
```

```
40000/40000 [==============================] - 757s 19ms/step - loss: 0.3408 - acc: 0.8894 - val_l
oss: 0.3645 - val_acc: 0.8787
Epoch 4/10
40000/40000 [==============================] - 756s 19ms/step - loss: 0.3332 - acc: 0.8893 - val_l
oss: 0.3548 - val_acc: 0.8774
Epoch 5/10
40000/40000 [==============================] - 760s 19ms/step - loss: 0.3189 - acc: 0.8888 - val_l
oss: 0.3453 - val_acc: 0.8763
Epoch 6/10
40000/40000 [==============================] - 765s 19ms/step - loss: 0.2988 - acc: 0.8901 - val_l
oss: 0.3326 - val_acc: 0.8798
Epoch 7/10
40000/40000 [==============================] - 763s 19ms/step - loss: 0.5015 - acc: 0.8909 - val_l
oss: 1.9322 - val_acc: 0.8788
Epoch 8/10
40000/40000 [==============================] - 763s 19ms/step - loss: 1.7580 - acc: 0.8897 - val_l
oss: 1.9322 - val_acc: 0.8788
Epoch 9/10
40000/40000 [==============================] - 763s 19ms/step - loss: 1.7580 - acc: 0.8897 - val_l
oss: 1.9322 - val_acc: 0.8788
Epoch 10/10
40000/40000 [==============================] - 764s 19ms/step - loss: 1.7580 - acc: 0.8897 - val_l
oss: 1.9322 - val_acc: 0.8788
Accuracy: 87.88%
Test score: 1.9322171548843405
Test accuracy: 0.8788
Time taken to run this cell : 2:11:21.863930
```
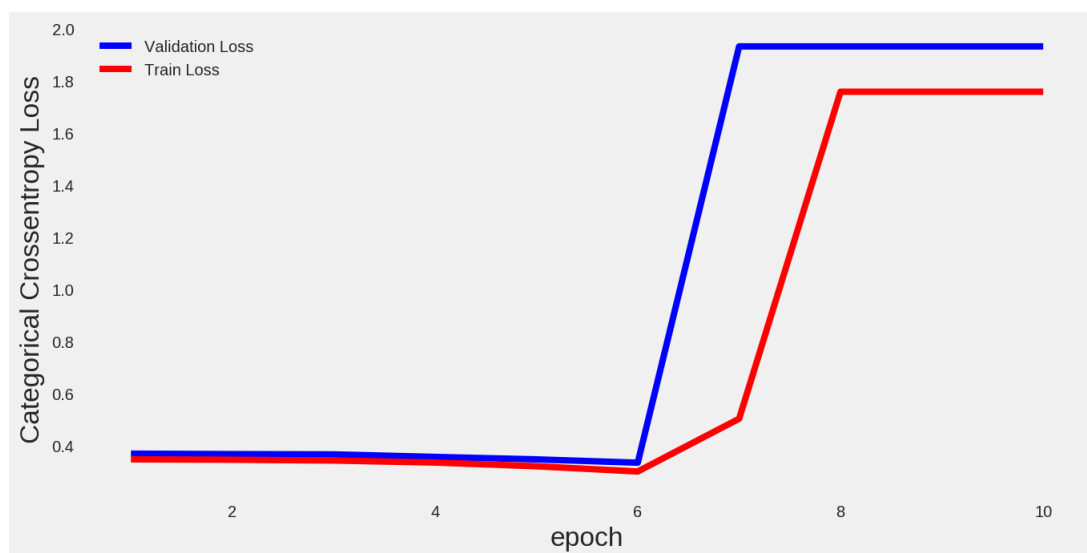

Deep **LSTMs** with batch normalizations and dropout

In [41]:

```python
from keras.layers.normalization import BatchNormalization


# create the model
embedding_vecor_length = 64
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length, input_length=max_review_length))
model.add(BatchNormalization())
model.add(LSTM(100, return_sequences=True))
model.add(Dropout(0.25))
model.add(BatchNormalization())
model.add(LSTM(100))
model.add(Dropout(0.25))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
#Refer: https://datascience.stackexchange.com/questions/10615/number-of-parameters-in-an-lstm-mode
l
```

```
_____
Layer (type)                 Output Shape              Param #
=================================================================
```

```
embedding_9 (Embedding)          (None, 600, 64)          320000
_____
batch_normalization_15 (Batc (None, 600, 64)              256
_____
lstm_21 (LSTM)                   (None, 600, 100)         66000
_____
dropout_16 (Dropout)             (None, 600, 100)         0
_____
batch_normalization_16 (Batc (None, 600, 100)             400
_____
lstm_22 (LSTM)                   (None, 100)              80400
_____
dropout_17 (Dropout)             (None, 100)              0
_____
dense_6 (Dense)                  (None, 1)                101
=================================================================
Total params: 467,157
Trainable params: 466,829
Non-trainable params: 328
_____
None
```

In [42]:

```python
start = datetime.now()

history = model.fit(X_train, y_train, epochs = epochs, batch_size = batch_size, verbose=1,validatio
n_data=(X_test, y_test))
scores = model.evaluate(X_test, y_test, verbose = 0)
print("Accuracy: %.2f%%" % (scores[1]*100))


score = model.evaluate(X_test, y_test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])

fig,ax = plt.subplots(1,1)
ax.set_xlabel('epoch') ; ax.set_ylabel('Categorical Crossentropy Loss')

# list of epoch numbers
x = list(range(1,epochs +1))

# print(history.history.keys())
# dict_keys(['val_loss', 'val_acc', 'loss', 'acc'])
# history = model_drop.fit(X_train, Y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, va
lidation_data=(X_test, Y_test))

# we will get val_loss and val_acc only when you pass the paramter validation_data
# val_loss : validation loss
# val_acc : validation accuracy

# loss : training loss
# acc : train accuracy
# for each key in histrory.histrory we will have a list of length equal to number of epochs

vy = history.history['val_loss']
ty = history.history['loss']
plt_dynamic(x, vy, ty, ax)


print('Time taken to run this cell :', datetime.now() - start)
```

```
Train on 40000 samples, validate on 10000 samples
Epoch 1/10
40000/40000 [==============================] - 1512s 38ms/step - loss: 0.2914 - acc: 0.8946 - val_
loss: 0.3778 - val_acc: 0.8801
Epoch 2/10
40000/40000 [==============================] - 1458s 36ms/step - loss: 0.2101 - acc: 0.9199 - val_
loss: 0.2296 - val_acc: 0.9091
Epoch 3/10
40000/40000 [==============================] - 1480s 37ms/step - loss: 0.1874 - acc: 0.9281 - val_
loss: 0.2228 - val_acc: 0.9139
Epoch 4/10
40000/40000 [==============================] - 1577s 39ms/step - loss: 0.1656 - acc: 0.9382 - val_
loss: 0.2142 - val_acc: 0.9142
Epoch 5/10
```

Epoch 5/10
40000/40000 [==============================] - 1612s 40ms/step - loss: 0.1459 - acc: 0.9450 - val_
loss: 0.2139 - val_acc: 0.9222
Epoch 6/10
40000/40000 [==============================] - 1639s 41ms/step - loss: 0.1291 - acc: 0.9515 - val_
loss: 0.2247 - val_acc: 0.9188
Epoch 7/10
40000/40000 [==============================] - 1693s 42ms/step - loss: 0.1119 - acc: 0.9589 - val_
loss: 0.3463 - val_acc: 0.8855
Epoch 8/10
40000/40000 [==============================] - 1679s 42ms/step - loss: 0.1014 - acc: 0.9617 - val_
loss: 0.2382 - val_acc: 0.9184
Epoch 9/10
40000/40000 [==============================] - 1669s 42ms/step - loss: 0.0858 - acc: 0.9687 - val_
loss: 0.2813 - val_acc: 0.9196
Epoch 10/10
40000/40000 [==============================] - 1693s 42ms/step - loss: 0.0758 - acc: 0.9723 - val_
loss: 0.2770 - val_acc: 0.9185
Accuracy: 91.85%
Test score: 0.2770338848128915
Test accuracy: 0.9185
Time taken to run this cell : 4:37:11.954754
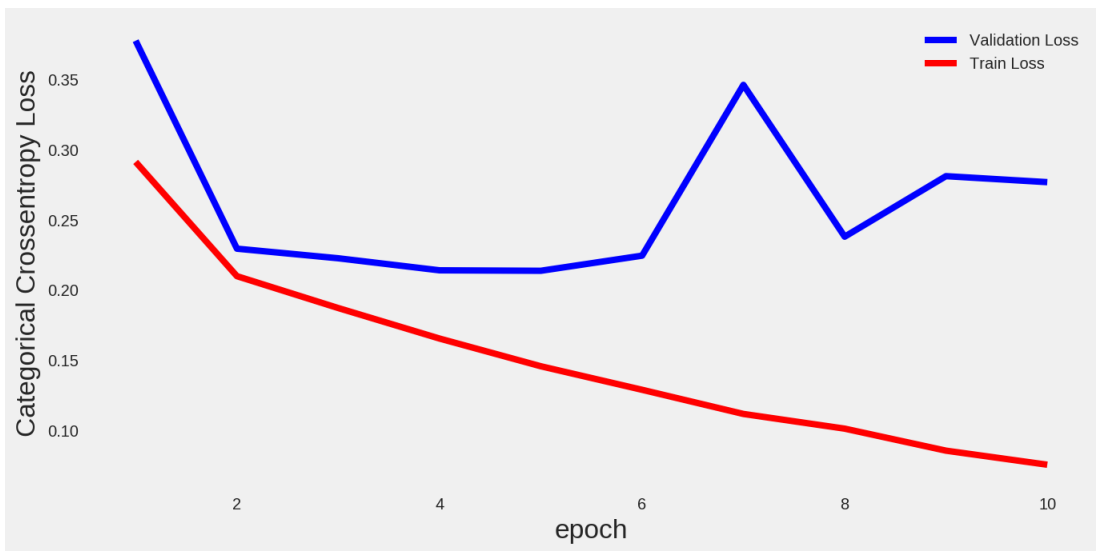
In [48]:

```python
from prettytable import PrettyTable

x = PrettyTable()

x.field_names = ["S.R",'Model',"Test loss","Accuracy",]

x.add_row([(1),"Relu", 14.159, '12.15%'])
x.add_row([(2),"Sigmoid",  1.93, '87.88%'])
x.add_row([(3),'Sigmoid with Drop outs and BN', 0.277, '91.85%'])

x.get_string(title = "-----SUMMARY-----")

print(x)
```

```
+-----+-------------------------------+-----------+----------+
| S.R |             Model             | Test loss | Accuracy |
+-----+-------------------------------+-----------+----------+
|  1  |             Relu              |  14.159   |  12.15%  |
|  2  |            Sigmoid            |   1.93    |  87.88%  |
|  3  | Sigmoid with Drop outs and BN |   0.277   |  91.85%  |
+-----+-------------------------------+-----------+----------+
```