Process:
1:Load, clean the data and tokenize
2:Encode the words
3:Word embedding
4:Build rnn model (Create embdding and LSTM layers)
5:Run and test

In [1]:

```
import numpy as np
import pandas as pd
import tensorflow as tf
import os.path
from sklearn.model_selection import train_test_split
from nltk.corpus import stopwords
import nltk
from keras.preprocessing.text import Tokenizer
from bs4 import BeautifulSoup
import re
import string
from keras.preprocessing.sequence import pad_sequences
import warnings
warnings.filterwarnings('ignore')
```

```
Using TensorFlow backend.
```

Load and clean the messages as well as encoding the lables

In [2]:

```
def load_clean(filepath):
    '''Load & clean the data'''

    #Load Data
    data = pd.read_csv(filepath)
    #rows_number=data.shape[0]
    messages=[]
    for message in data['v2']:
        #Extra celaning of text before Keras tokenization
        #Removing stopwords
        nltk.download("stopwords")
        stop_words = set(stopwords.words('english'))
        message=' '.join(i for i in message.split() if i not in stop_words)
        #Here, BeauifulSoup is used to encode not completely deccoded text(decoded from html code)
to html code again
        message = BeautifulSoup(message, 'lxml')
        #Later we strip away tags in the html encodings and decode them to text
        message=message.get_text()
        messages.append(message)

    #Encode labels
    labels=[]
    [labels.append(0) if label=="spam" else labels.append(1) for label in data['v1']]
    labels = np.asarray(labels)
    return messages,labels
```

Tokenize sentences and encode their words to integers

In [3]:

```
def encode_words(sentences):
    '''Convert words to numbers'''

    #Since we read from csv, we need to do some encoding
    #Remove u'
    sentences=[x.encode('utf-8') for x in sentences]
    #Remove \xHH characters
    sentences=[re.sub(r'[^\x00-\x7f]',r'', x) for x in sentences]
```

```
    #Keras tokenization (punctualtion removal, normalization and split by white space)
    tokenize = Tokenizer()
    #Fit tokenizer to the whole data
    tokenize.fit_on_texts(sentences)
    data_seq=tokenize.texts_to_sequences(sentences)
    word_index = tokenize.word_index
    #Choose the maximum number of tokens in all sequences
    num_tokens = [len(tokens) for tokens in data_seq]
    max_seq_length=np.max(num_tokens)
    #Make sequences to have the same length(add extra zeros to the beginnings)
    data_seq = pad_sequences(data_seq, maxlen = max_seq_length,
                                padding='pre', truncating='pre')
    #print(data_seq)
    return data_seq,word_index
```

In [4]:

```
def create_model_inputs():
    '''Define model inputs'''

    #Resert the default graph
    tf.reset_default_graph()
    #Model's placeholders for inputs
    inputs = tf.placeholder(tf.int32, [None, None], name='inputs')
    targets = tf.placeholder(tf.int32, [None, None], name='targets')
    keep_prob = tf.placeholder(tf.float32, name='keep_prob')

    return inputs,targets,keep_prob
```

Create the RNN model with 2 layer LSTM

In [5]:

```
def
build_RNN(vocabulary_size,embedding_size,inputs,num_hidden,lstm_layer_numbers,keep_prob,batch_size
):
    '''Build RNN'''

    #Embedding Layer
    '''Intialize embeddings for the words. Embedding layer connects the words to the LSTM layers (
words were respresnted in one hot vectors before the embedding and now they are embedded to the em
bedding_size vectors instead of vocabulary size vectors)'''
    embedding = tf.Variable(tf.random_uniform((vocabulary_size, embedding_size), -1, 1))
    embed = tf.nn.embedding_lookup(embedding, inputs)
    #Define LSTM layers
    lstms=[]
    for i in range(lstm_layer_numbers):
        lstms.append(tf.contrib.rnn.BasicLSTMCell(num_hidden))
    # Add regularization dropout to the LSTM cells
    drops = [tf.contrib.rnn.DropoutWrapper(lstm, output_keep_prob=keep_prob) for lstm in lstms]
    # Stack up multiple LSTM layers
    stacked_lstm = tf.contrib.rnn.MultiRNNCell(drops)
    # Getting the initial state
    initial_state = stacked_lstm.zero_state(batch_size, tf.float32)
    outputs, final_state = tf.nn.dynamic_rnn(stacked_lstm, embed, initial_state=initial_state)

    return initial_state, outputs, final_state
```

In [6]:

```
def get_batches(x, y, batch_size):
    '''Using generator to return batches for train, validation and test data'''

    n_batches = len(x)//batch_size
    x, y = x[:n_batches*batch_size], y[:n_batches*batch_size]
    for ii in range(0, len(x), batch_size):
        yield x[ii:ii+batch_size], y[ii:ii+batch_size]
```

In [7]:

```
#Input data
emaildata_file="./spam.csv"
```

In [ ]:

```
#Load and clean data; return clean messages and labels
text_messages,labels=load_clean(emaildata_file)
```

In [9]:

```
print(labels)
```

```
[1 1 0 ..., 1 1 1]
```

In [10]:

```
#Words to int
data_sequences,word_index=encode_words(text_messages)
```

In [11]:

```
word_index
```

Out[11]:


```
{'raining': 1592,
 'yellow': 4011,
 'four': 2311,
 'prices': 6549,
 'woods': 6739,
 "friend's": 2388,
 'hanging': 1973,
 'looking': 396,
 'electricity': 3752,
 'scold': 3754,
 'lord': 6828,
 'rp176781': 4605,
 'callin': 2480,
 'ew': 6749,
 'hearin': 8343,
 'screaming': 1608,
 'disturb': 1093,
 'prize': 107,
 'andre': 8476,
 'smsing': 7980,
 'wednesday': 1274,
 'oooh': 3205,
 'specially': 1072,
 'nigh': 7532,
 'tired': 809,
 'snuggles': 8160,
 "'wnevr": 6818,
 'second': 621,
 'attended': 7746,
 'txtno': 3131,
 'available': 616,
 'scraped': 8165,
 '2kbsubject': 4899,
 'scallies': 7419,
 'errors': 5266,
 'cooking': 2231,
 'fingers': 1223,
 'maraikara': 6845,
 'hero': 5162,
 "how've": 6751,
 'y87': 6931,
 'here': 233,
 'specialise': 5727,
 '47': 7730,
 'china': 2793,
 'dogwood': 7964,
 'dorm': 3261,
```

```
'08718711108': 4829,
'previews': 5968,
'84122': 5275,
'w111wx': 2211,
'kids': 1035,
'84128': 2631,
'eastenders': 3427,
'09058091870': 7880,
"i'd": 854,
"i'm": 6,
'spotty': 6044,
'golden': 6590,
"ta's": 4419,
"dat's": 3171,
'replace': 4343,
'brought': 2172,
'sterm': 8621,
'000pes': 5744,
'txt': 28,
'univ': 7383,
'9t': 3161,
'cheating': 4109,
'spoke': 1477,
'ec2a': 1668,
'browse': 5573,
'dnt': 818,
'music': 485,
'passport': 7259,
'strike': 2873,
'until': 7549,
'paperwork': 3094,
'holy': 3815,
'relax': 1506,
'successful': 2544,
'brings': 1106,
'premarica': 6284,
'hols': 1913,
'yahoo': 1209,
'hurt': 528,
'99': 3455,
'glass': 5361,
'47per': 5980,
'hole': 5020,
'hold': 776,
'95': 5385,
'up4': 3704,
'tirupur': 2475,
'itself': 1480,
'wana': 943,
'drvgsto': 4901,
'pints': 7613,
'smth': 681,
'want': 26,
'organizer': 5614,
'preferably': 2167,
'hon': 3957,
'hoo': 7609,
'travel': 1772,
'how': 31,
'hot': 526,
'hor': 2937,
'hos': 8379,
'hop': 1311,
'significance': 4694,
'1172': 8208,
'beauty': 4255,
'yun': 1794,
'wan2': 4127,
'plyr': 4966,
'wrong': 748,
'lololo': 6073,
'bsnl': 6320,
'types': 5465,
'ibored': 8519,
'aroundn': 6954,
'wins': 1685,
'yunny': 3934,
```

```
'alian': 6227,
'age16': 1119,
'tulip': 3430,
'areyouunique': 4496,
'keeps': 3683,
'lambda': 5889,
'wind': 3459,
'wine': 932,
'wc1n': 8783,
'afterwards': 8455,
"dramastorm's": 6465,
'vary': 1513,
'kickoff': 3422,
'82050': 3720,
'87575': 1894,
'welcomes': 4028,
'lovingly': 3637,
'fit': 2374,
'bringing': 2011,
'fix': 1686,
'max10mins': 1621,
'4eva': 4334,
'matured': 2014,
'09095350301': 8642,
'wales': 2276,
'hidden': 7726,
'nokia6600': 3712,
'easier': 2126,
'duvet': 8052,
'vouchers': 550,
'effects': 3234,
'schools': 3287,
'go2sri': 7321,
'silver': 3567,
'rumour': 3624,
'fetching': 4812,
'dload': 1790,
'nattil': 7273,
'arrow': 6192,
'addicted': 2090,
'burial': 4981,
'financial': 6226,
'fgkslpopw': 8247,
'series': 1905,
'allah': 1700,
'spider': 4359,
'bowls': 6694,
'strips': 5240,
"we'd": 1938,
'2day': 1099,
'ring': 576,
'rt': 5660,
'ru': 1868,
'rv': 4982,
'forwarding': 5961,
'rr': 6330,
'rs': 726,
'ha': 680,
'help08714742804': 5520,
'sms': 193,
'rd': 1838,
're': 1404,
'noice': 7627,
'09061701851': 7438,
'ofcourse': 8750,
'dracula': 3654,
'toking': 7994,
'sheet': 6317,
'ate': 1573,
'shelves': 6135,
'atm': 1679,
'ups': 4525,
'shipped': 3115,
"today's": 1354,
'clothes': 4247,
'veggie': 5286,
'kfc': 7742,
```

```
'hear': 349,
'ente': 5168,
"b'tooth": 5844,
'basketball': 7288,
'service': 177,
'09061743386': 4219,
'engagement': 8080,
'xin': 7300,
'needed': 1921,
'listed': 7291,
'loosu': 7777,
'hiya': 1135,
'listen': 787,
'clubmoby': 8125,
'wisdom': 2906,
'termsapply': 8175,
'trek': 4870,
'peril': 7090,
'showed': 7262,
'saeed': 5998,
'tree': 2421,
'likely': 2478,
'project': 779,
'percentages': 7083,
'bridgwater': 4796,
'feeling': 531,
'boston': 1658,
'09061749602': 5093,
'selflessness': 6780,
'9755': 4709,
'9758': 8158,
'affairs': 2922,
'escalator': 5149,
'flippin': 5294,
'responsible': 5679,
'witot': 5451,
'andros': 4056,
'okie': 605,
'causing': 4158,
'doors': 2697,
'hum': 6209,
'shall': 357,
'doin': 1057,
'victoria': 8684,
'doit': 5023,
'swiss': 3914,
'laxinorficated': 7144,
'mouth': 3019,
'daywith': 6776,
'letter': 1900,
'thriller': 6909,
'cops': 8710,
'marsms': 6060,
'camp': 7690,
'passes': 7392,
'everythin': 5356,
'41685': 3342,
'tech': 3392,
'84484': 7572,
'scream': 1531,
'came': 427,
'marvel': 8753,
'saying': 584,
'bomb': 8270,
'prin': 7671,
'insects': 5372,
'advisors': 8363,
'teresa': 5814,
'prix': 4782,
'gauge': 5430,
'buzzzz': 6696,
'participate': 6803,
'lessons': 1087,
'busy': 544,
"u'll": 1399,
'menu': 1309,
'appreciated': 3485,
```

```
'appreciated': 3109,
'cougar': 5177,
'touched': 2348,
'rich': 2784,
'rice': 3385,
'pocked': 7810,
'plate': 6968,
'0871277810810': 3020,
'platt': 6042,
'uworld': 8579,
'tips': 7657,
'lmao': 1422,
'bus8': 5136,
'kittum': 7274,
'asssssholeeee': 7566,
'piggy': 4843,
'respond': 1719,
'disaster': 8376,
'fair': 2286,
'rupaul': 5949,
'goodnight': 1089,
'result': 2004,
'bleh': 3374,
'best': 241,
'lotz': 8212,
'ctargg': 5041,
'lots': 653,
'lotr': 2686,
'wikipedia': 4826,
'80122300p': 6401,
'stamps': 2592,
'score': 2624,
'glasgow': 5570,
'men': 1070,
'nationwide': 7573,
'nature': 1605,
'rolled': 6416,
'rajini': 5517,
'icicibank': 3559,
'wtc': 6599,
'wtf': 1987,
'wth': 7375,
'roller': 8098,
'pity': 8816,
'accident': 2905,
'brown': 6002,
'country': 1302,
'macedonia': 4371,
'planned': 1449,
'lookin': 1611,
'tomarrow': 2346,
'machan': 1908,
'login': 1372,
'argue': 1957,
'asked': 415,
'30th': 4495,
'itried2tell': 5791,
'2nd': 382,
'happenin': 5396,
'darlin': 725,
'sk38xh': 1396,
'250': 634,
'255': 8153,
'9ja': 2453,
'billing': 7901,
'shouting': 5583,
'fri': 742,
'fro': 5085,
'frm': 784,
'much': 74,
'wuld': 4031,
'stadium': 6568,
'parents': 720,
'obese': 5207,
'life': 119,
'dave': 2450,
'lift': 1345,
'chile': 7463,
```

```
'child': 2280,
'25p': 1235,
'spin': 7802,
'bridal': 7134,
'chill': 2647,
'unsold': 1774,
'3680': 3072,
'09058091854': 3435,
'2wks': 3058,
'selfindependence': 7544,
'meetin': 2059,
'k': 52,
'42810': 8263,
'a21': 5702,
'congratulation': 8350,
'played': 2086,
'078': 8513,
'player': 719,
'feed': 8682,
'things': 204,
'honi': 7599,
'tgxxrz': 4704,
'dha': 5971,
'hont': 7711,
'split': 8091,
'babies': 3564,
'4fil': 3471,
'08718727870150ppm': 8089,
'tops': 8008,
'ppm150': 5090,
'tune': 4942,
'academic': 4278,
'nachos': 7576,
'xxxxxxxxxxxxxx': 7177,
'opinions': 3869,
'gigolo': 7674,
'08701752560': 7225,
'dosomething': 6448,
'sleepy': 3991,
'nydc': 3511,
'87121': 1888,
'credited': 2121,
'waht': 3601,
'rushing': 8387,
'previous': 2929,
'hai': 1391,
'enters': 4675,
'ham': 2312,
'duffer': 5893,
'1lemon': 7852,
'had': 405,
'haf': 497,
'obedient': 6497,
'innocent': 3952,
'east': 4094,
'hat': 8795,
'hav': 447,
"t's": 1229,
'fromm': 2252,
'possible': 1127,
'twinks': 7418,
'possibly': 6734,
'birth': 2770,
'vday': 3744,
'shadow': 5996,
'unique': 1945,
'stylist': 5625,
'remind': 1863,
'steps': 7512,
'9280114': 8148,
'ola': 3512,
'right': 110,
'old': 568,
'crowd': 7909,
'people': 225,
'weds': 6114,
```

```
'easy': 373,
'feel': 162,
'fuuuuck': 8557,
'creep': 4002,
'enemies': 7488,
'08718725756': 6901,
'for': 172,
'bottom': 2883,
'fox': 6788,
'creative': 7433,
'treadmill': 8590,
'muhommad': 7660,
'wocay': 8686,
'suitemates': 7578,
'dental': 8808,
"hubby's": 8032,
'colleg': 7373,
'starring': 6452,
'losing': 2021,
'memorable': 4431,
'quiteamuzing': 8073,
'dollars': 1720,
'careabout': 5793,
'o': 838,
'suggestions': 8817,
'slightly': 3118,
'raised': 6170,
'statements': 6370,
'honeymoon': 5645,
'sol': 1434,
'soo': 3325,
'sos': 7536,
'69696': 3971,
'69698': 3065,
'soz': 8143,
'janx': 5401,
'4742': 1942,
'support': 845,
'constantly': 2755,
'halla': 5528,
'greatness': 5346,
'jane': 2640,
'happy': 84,
'b4280703': 3169,
'offer': 330,
'6wu': 3438,
'paypal': 5099,
'notifications': 4968,
'talents': 5671,
'fiting': 7662,
'congratulations': 731,
'inside': 1426,
'pest': 5621,
'lays': 3247,
'smashed': 3805,
'151': 8651,
'150': 637,
'153': 3150,
'half8th': 4893,
'textbook': 5783,
"''": 545,
'exist': 4467,
'accounting': 7498,
'ericsson': 3037,
'dealer': 7737,
'norm150p': 1558,
'80160': 5262,
'floor': 2327,
'actor': 3077,
'uttered': 7625,
'flood': 7423,
'role': 1627,
'ambitious': 6613,
'smell': 5050,
'truffles': 3145,
"'t": 2091,
'intend': 5788
```

```
 'intend': 5788,
'fathima': 2739,
'07742676969': 3002,
'outage': 7535,
'mre': 8322,
'hollalater': 5070,
'jewelry': 7784,
'nxt': 1400,
'loveme': 3308,
'preponed': 8426,
'cuddled': 6544,
'07732584351': 4390,
'broadband': 7211,
'time': 22,
'push': 3909,
'timi': 7403,
'6230': 6371,
'sday': 8525,
'chain': 2650,
'saibaba': 8770,
'cudnt': 5166,
'3ss': 3241,
'boltblue': 4698,
'oso': 500,
'baller': 8392,
"when's": 2859,
'overdid': 8567,
'lara': 6034,
'macha': 4885,
'comuk': 1130,
'followin': 4622,
'macho': 2913,
'machi': 4677,
'jeri': 4897,
'k718': 6138,
'prepaid': 3035,
'doke': 6266,
'minuts': 1860,
'cheap': 1034,
'maretare': 7017,
"tyler's": 8296,
'choice': 1811,
'onwords': 8438,
'pleasssssseeeeee': 4521,
'5min': 2355,
'exact': 2131,
'28days': 3449,
'minute': 600,
'tear': 1440,
'leave': 191,
'solved': 4032,
'settle': 2835,
'team': 1283,
'loads': 925,
'prevent': 4866,
'spiritual': 8798,
'rents': 3693,
'videochat': 1788,
'sigh': 4060,
'prediction': 4508,
'sign': 1191,
'08712402972': 7077,
'erotic': 8643,
'shirts': 2864,
'rentl': 1792,
'workand': 6223,
'headset': 7930,
"'hw": 3879,
'celebrated': 7119,
'melt': 4217,
'current': 1708,
'300': 1882,
'axel': 8194,
'falling': 3630,
'ground': 2610,
'boost': 1852,
'unintentionally': 8538,
'funeral': 4197,
```

```
'funeral': 4197,
'understanding': 2122,
'yards': 8329,
'address': 519,
'alone': 675,
'along': 2159,
'neville': 6857,
'brilliant': 1867,
'300603': 3138,
'wherever': 1574,
"anybody's": 5975,
'bw': 8155,
'fassyole': 8498,
'studies': 8588,
'influx': 8801,
'love': 24,
'prefer': 2795,
'bloody': 1937,
'fake': 3722,
'4jx': 4957,
'gotbabes': 6800,
'sky': 1442,
'crammed': 5978,
'working': 399,
'positive': 8735,
'angry': 630,
'tightly': 6817,
'wicket': 8620,
'opposed': 7067,
'wondering': 1140,
'films': 2894,
"cann't": 2907,
'trishul': 7607,
'loving': 674,
'09065394973': 5930,
'afford': 6364,
'oooooh': 8668,
'appendix': 6793,
'everywhere': 3017,
'ip4': 1164,
'scratches': 5179,
'easiest': 6334,
'behalf': 7970,
'logos': 3516,
'valued': 796,
'hussey': 5034,
'pretend': 7115,
'lttrs': 3873,
'printing': 6526,
'values': 7121,
'following': 1293,
'logon': 8067,
'mesages': 4268,
'muah': 4333,
'awesome': 599,
'weasels': 6685,
'parachute': 5887,
'88066': 2812,
'hides': 8191,
'admirer': 1003,
'offense': 8183,
'dooms': 7189,
'poking': 6923,
'meive': 4775,
'62220cncl': 7475,
'fps': 6280,
'elephant': 6911,
'69200': 6756,
'lido': 2307,
'laundry': 3735,
'landmark': 7893,
'23f': 6007,
'23g': 6008,
'spot': 7453,
'dats': 4505,
'suntec': 2651,
'unclaimed': 4919,
'data': 529,
```

```
'date': 538,
'such': 6659,
'data': 5190,
'brainless': 7051,
'disagreeable': 8454,
'stress': 2509,
'surfing': 1698,
'natural': 2772,
'sp': 1183,
'st': 887,
'complaining': 5745,
'si': 2001,
'sh': 2713,
'so': 33,
'sn': 2553,
'swollen': 7836,
'sc': 5726,
'misplaced': 6377,
'sg': 6832,
'hol': 3313,
'se': 2750,
'sd': 5143,
'drunken': 4114,
'bootydelious': 3021,
'differences': 8294,
'speedchat': 3464,
'years': 464,
'professors': 3257,
'course': 771,
'studentfinancial': 7091,
'konw': 3600,
'disconnect': 4172,
'jia': 3618,
'avin': 8065,
'attraction': 5221,
'jiu': 1711,
'930': 3276,
'decades': 8129,
'instantly': 2823,
'conveying': 7720,
'matches': 1236,
'smarter': 4399,
'n9dx': 3106,
'feelin': 3663,
'records': 1770,
'subscribers': 7926,
'sorted': 2637,
'twilight': 3659,
'maintaining': 8660,
'matched': 5519,
'pokkiri': 5220,
'shouted': 2513,
'83435': 6424,
'blacko': 8499,
'othrwise': 7885,
'quarter': 8228,
'ovr': 5531,
'retrieve': 3846,
'padhe': 6305,
'receipt': 1802,
'disasters': 6955,
'pataistha': 7353,
'joker': 8286,
'83332': 7069,
'blu': 4205,
'alwa': 6827,
'83383': 3935,
'wiskey': 3890,
'trauma': 4991,
'internet': 1308,
'hcl': 6357,
'flurries': 7522,
'ppt150x3': 5078,
'sheffield': 3407,
'1131': 6985,
'million': 7804,
'possibility': 7552,
```

```
'quite': 323,
'grandma': 8009,
'vijaykanth': 8122,
'raed': 3610,
'training': 1249,
'thankyou': 6115,
'rael': 3602,
'dunno': 377,
'swtheart': 2384,
'initiate': 6781,
'massive': 4271,
'dinero': 5743,
'neglect': 4903,
'20m12aq': 6540,
'emotion': 7883,
'oni': 1885,
'frwd': 6233,
'spoken': 2056,
'potter': 3523,
'one': 29,
'spanish': 2328,
'vava': 2313,
'69911': 5280,
'open': 652,
'city': 1761,
'sozi': 7975,
'bite': 2483,
'uks': 2866,
'indicate': 4243,
'fml': 1912,
'2': 4,
'stifled': 6065,
'stuffed': 5097,
'definitly': 7991,
'bits': 4979,
'coccooning': 6924,
'floppy': 7951,
'snatch': 5329,
'fooled': 5676,
'boyfriend': 3626,
'remembr': 3522,
'depressed': 3458,
'scrumptious': 4891,
'cutefrnd': 2382,
'arun': 2732,
'arul': 5941,
'attracts': 7332,
'illness': 4346,
'sao': 4449,
'sam': 1683,
'sac': 4104,
'turned': 4980,
'argument': 1684,
'sae': 557,
'sad': 563,
'woah': 8188,
'say': 109,
'sar': 3469,
'saw': 475,
'sat': 360,
'1cup': 7853,
'zoe': 3408,
'babysit': 7037,
'aproach': 3875,
'15pm': 7982,
'note': 2243,
'taka': 8311,
'algarve': 3005,
'take': 54,
'wanting': 2087,
'ericson': 6507,
'handing': 6527,
'printer': 6642,
'opposite': 8219,
'knew': 905,
'buffet': 2935,
'printed': 3047,
```

```
'pages': 2218,
'countinlots': 6893,
'02085076972': 7266,
'phil': 6856,
'infections': 4316,
'drive': 585,
'werethe': 4652,
'salt': 8552,
'annoncement': 4471,
'walking': 1418,
'5wq': 6494,
'inclu': 8787,
'bright': 2148,
"joke's": 7724,
'5wb': 1417,
'5we': 1165,
'applyed': 8093,
'slow': 953,
'farting': 8360,
'robs': 6622,
'coaxing': 5173,
'foward': 8595,
'jaykwon': 4680,
'going': 30,
'actin': 4380,
'hockey': 2967,
'slob': 6794,
'caroline': 3095,
'carolina': 8267,
'b4u': 6059,
'psychiatrist': 5623,
'4882': 3229,
'freezing': 2703,
'murali': 5836,
'compliments': 8778,
'awaiting': 1260,
'settings': 2186,
'getstop': 3300,
'borrow': 3306,
'tenerife': 1349,
'worried': 983,
'racal': 5057,
'priest': 6847,
'roger': 2229,
'worries': 1095,
'tortilla': 3001,
'where': 203,
'xmas': 435,
'busetop': 4636,
'persian': 8254,
'anyways': 1939,
'pisces': 8616,
"alex's": 4864,
'cttargg': 5040,
'8883': 8068,
'dormitory': 8466,
'x29': 6935,
'refreshed': 4602,
'availa': 4951,
'jobs': 3916,
'screen': 2653,
"employer's": 4578,
'concentrate': 2596,
'spare': 3285,
'amore': 4361,
'spark': 5640,
'listening2the': 5370,
'many': 181,
's': 207,
'residency': 6382,
'7876150ppm': 4008,
'expression': 3135,
"can't": 173,
'stream': 6246,
'conected': 5230,
'call2optout': 888,
'anti': 2628,
```

```
'3000': 7562,
'mapquest': 7963,
'boat': 2763,
'cramps': 2714,
'swashbuckling': 8282,
'stretch': 3556,
'west': 3983,
'breath': 3443,
'reflex': 5920,
'wants': 561,
'gist': 3279,
'hlday': 7689,
'coughing': 7214,
'09111032124': 4641,
'820554ad0a1705572711': 8731,
'photos': 1782,
'300p': 3829,
'09058097218': 5189,
'naseeb': 6203,
'single': 1222,
'squeezed': 8659,
'situation': 1066,
'3uz': 2387,
'ive': 940,
'wire3': 7230,
'purse': 3494,
'bros': 2858,
'blah': 2424,
'limping': 8230,
'verified': 4175,
'0125698789': 4531,
'thinkin': 1147,
'cost1': 1620,
'cost3': 7450,
'thirtyeight': 4519,
'downs': 6691,
'sterling': 6046,
'askin': 1514,
'sickness': 7651,
'mtnl': 8439,
'cheers': 1159,
'callon': 6652,
'cheery': 5381,
'italian': 1805,
'defo': 5182,
'88888': 4013,
'natalie2k9': 8428,
'costa': 1433,
'volcanoes': 6949,
'nothin': 3848,
"shahjahan's": 7248,
'costs': 1834,
'1winaweek': 2776,
'grumpy': 4435,
'hubby': 3588,
'dimension': 5698,
'summer': 1128,
'being': 8636,
'150p16': 3433,
'forevr': 2917,
'sum1': 2306,
'08714712412': 8409,
'88088': 3867,
'6089': 8658,
'lolnice': 4692,
'ghodbandar': 5724,
'weekly': 486,
'81151': 2052,
'310303': 6900,
'kerala': 1578,
'adsense': 8420,
'drugdealer': 6005,
'f4q': 4604,
'starving': 6970,
'proze': 8011,
'aspects': 6427,
'around': 167,
```

```
 'lnly': 5681,
 'pos': 6486,
 'dark': 2870,
 'traffic': 3678,
 'pop': 2495,
 '2geva': 4030,
 'world': 303,
 'postal': 5241,
 'vague': 7497,
 'dare': 2239,
 'stranger': 2398,
 'poo': 7394,
 'quizclub': 6400,
 'alaipayuthe': 3696,
 'gimme': 2517,
 'clas': 8278,
 'gimmi': 7460,
 'slovely': 5725,
 'masteriastering': 7149,
 'ortxt': 8361,
 '4few': 5229,
 'monthlysubscription': 4720,
 'playin': 7537,
 '5wkg': 5142,
 'thinks': 773,
 "there'll": 8176,
 'strewn': 5857,
 'memories': 8213,
 'noon': 1121,
 "''ok''": 1859,
 ...}
```

In [12]:

```
print(data_sequences)
```

```
[[   0    0    0 ...,   20 4361   98]
 [   0    0    0 ...,  422    2 1885]
 [   0    0    0 ...,  618  343 2936]
 ...,
 [   0    0    0 ...,   33  504 8817]
 [   0    0    0 ...,  993  151   12]
 [   0    0    0 ...,   88  436  219]]
```

In [13]:

```
#Split the data into train,test and validation sets
#First split train and test parts, then split train part to train and validation parts
X_train, X_test, y_train, y_test = train_test_split(data_sequences, labels, test_size=0.2, random_s
tate=1)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=1)
```

In [14]:

```
#Define Parameters
#Vocab size plus one for 0, the int number that added for padding
n_input = len(word_index)+1
# number of units
num_hidden = 256
lstm_layer_numbers=2
embed_size=300
batch_size= 250
learning_rate=0.001
```

Build and execute the graph

In [15]:

```
inputs,targets,keep_prob=create_model_inputs()
initial_state, outputs, final_state = build_RNN(n_input,embed_size,inputs,num_hidden,lstm_layer_num
bers,keep_prob,batch_size)
```

```
# Loss and optimizer
#second parameter: one output which indicates if the input message is spam or ham
predictions = tf.contrib.layers.fully_connected(outputs[:, -1], 1, activation_fn=tf.sigmoid,
                                                weights_initializer=tf.truncated_normal_initializer
stddev=0.1),
                                                biases_initializer=tf.zeros_initializer())

loss_function = tf.losses.mean_squared_error(targets, predictions)
optimizer = tf.train.AdadeltaOptimizer(learning_rate).minimize(loss_function)
correct_pred = tf.equal(tf.cast(tf.round(predictions), tf.int32), labels)
accuracy = tf.reduce_mean(tf.cast(correct_pred, tf.float32))


#Execute the graph
sess = tf.Session()
saver = tf.train.Saver()
init_op = tf.initialize_all_variables()
sess.run(init_op)
no_of_batches_train = int(len(X_train)/batch_size)
no_of_batches_valid = int(len(X_val)/batch_size)

epochs = 35
for epoch in range(epochs):
    state = sess.run(initial_state)
    avg_cost_train = 0
    avg_acc_train= 0
    for ii, (x, y) in enumerate(get_batches(X_train, y_train, batch_size), 1):
        _, cost, acc= sess.run([optimizer, loss_function,accuracy], feed_dict={inputs: x,
                                                        targets: y[:, None],keep_prob: 0.5,initial_
ate: state})

        avg_cost_train += cost / no_of_batches_train
        avg_acc_train += acc / no_of_batches_train
    state_val = sess.run(initial_state)
    avg_cost_val = 0
    avg_acc_val = 0
    for ii, (x, y) in enumerate(get_batches(X_val, y_val, batch_size), 1):
        _, cost, acc= sess.run([optimizer, loss_function, accuracy], feed_dict={inputs: x,
                                                        targets: y[:, None],keep_prob: 1,initial_st
e: state_val})

        avg_cost_val += cost / no_of_batches_valid
        avg_acc_val += acc / no_of_batches_valid
    print("Epoch:", epoch+1, "cost_train=", avg_cost_train, "cost_val=", avg_cost_val)
    print("acc_train=", avg_acc_train, "acc_val=", avg_acc_val)
#Save the model into a file
checkpoint="./model/savedmodel.ckpt"
save_path = saver.save(sess, checkpoint)
sess.close()
```

```
WARNING:tensorflow:From /usr/local/lib/python2.7/site-
packages/tensorflow/python/util/tf_should_use.py:107: initialize_all_variables (from
tensorflow.python.ops.variables) is deprecated and will be removed after 2017-03-02.
Instructions for updating:
Use `tf.global_variables_initializer` instead.
('Epoch:', 1, 'cost_train=', 0.23654443770647046, 'cost_val=', 0.23248936732610065)
('acc_train=', 0.63759224329675945, 'acc_val=', 0.78982198238372803)
('Epoch:', 2, 'cost_train=', 0.23368214390107564, 'cost_val=', 0.22953258951505029)
('acc_train=', 0.65787559747695923, 'acc_val=', 0.80250777800877882)
('Epoch:', 3, 'cost_train=', 0.22993492228644236, 'cost_val=', 0.22661814590295154)
('acc_train=', 0.68526858942849289, 'acc_val=', 0.80738693475723267)
('Epoch:', 4, 'cost_train=', 0.2268960763301168, 'cost_val=', 0.22372841338316601)
('acc_train=', 0.70241533858435501, 'acc_val=', 0.80933860937754298)
('Epoch:', 5, 'cost_train=', 0.22416281593697412, 'cost_val=', 0.22087885936101276)
('acc_train=', 0.71579817363194043, 'acc_val=', 0.81324191888173436)
('Epoch:', 6, 'cost_train=', 0.22121161435331615, 'cost_val=', 0.21803827583789825)
('acc_train=', 0.73649974380220695, 'acc_val=', 0.82104857762654615)
('Epoch:', 7, 'cost_train=', 0.21850442034857614, 'cost_val=', 0.21523192028204596)
('acc_train=', 0.7509281081812722, 'acc_val=', 0.82202440500259399)
('Epoch:', 8, 'cost_train=', 0.21552183159760069, 'cost_val=', 0.21244945625464121)
('acc_train=', 0.75490113241331924, 'acc_val=', 0.82397605975468946)
('Epoch:', 9, 'cost_train=', 0.21421175343649729, 'cost_val=', 0.20967903236548108)
('acc_train=', 0.76305629951613285, 'acc_val=', 0.825927734375)
('Epoch:', 10, 'cost_train=', 0.21026445499488286, 'cost_val=', 0.20695790151755011)
('acc_train=', 0.77267521194049282, 'acc_val=', 0.82787938912709547)
('Epoch:', 11, 'cost_train=', 0.20714088210037773, 'cost_val=', 0.20426748692989349)
('acc_train=', 0.78752179230962482, 'acc_val=', 0.82983104387919104)
```

('Epoch:', 12, 'cost_train=', 0.20487729460000992, 'cost_val=', 0.20160784820715588)
('acc_train=', 0.79086749894278396, 'acc_val=', 0.83080687125523878)
('Epoch:', 13, 'cost_train=', 0.20227206072637013, 'cost_val=', 0.19898511469364166)
('acc_train=', 0.79233124852180492, 'acc_val=', 0.83178271849950147)
('Epoch:', 14, 'cost_train=', 0.19994997446026125, 'cost_val=', 0.1964009553194046)
('acc_train=', 0.79839534844670978, 'acc_val=', 0.83178271849950147)
('Epoch:', 15, 'cost_train=', 0.19753757438489367, 'cost_val=', 0.19385014971097309)
('acc_train=', 0.80299569453511932, 'acc_val=', 0.83373437325159694)
('Epoch:', 16, 'cost_train=', 0.19442050052540644, 'cost_val=', 0.19133926431337994)
('acc_train=', 0.80780515500477379, 'acc_val=', 0.83373437325159694)
('Epoch:', 17, 'cost_train=', 0.19210152753761836, 'cost_val=', 0.18886122604211172)
('acc_train=', 0.81177817497934601, 'acc_val=', 0.83373437325159694)
('Epoch:', 18, 'cost_train=', 0.18951418144362317, 'cost_val=', 0.18641660114129385)
('acc_train=', 0.81679673705782208, 'acc_val=', 0.83763772249221802)
('Epoch:', 19, 'cost_train=', 0.18673642831189294, 'cost_val=', 0.18402357896169028)
('acc_train=', 0.82097886289869026, 'acc_val=', 0.83763772249221802)
('Epoch:', 20, 'cost_train=', 0.18458035268953868, 'cost_val=', 0.18166783452033997)
('acc_train=', 0.82306993433407383, 'acc_val=', 0.83861356973648071)
('Epoch:', 21, 'cost_train=', 0.18268886421407973, 'cost_val=', 0.17936232189337412)
('acc_train=', 0.82746118307113647, 'acc_val=', 0.83958939711252856)
('Epoch:', 22, 'cost_train=', 0.18011025020054408, 'cost_val=', 0.17709792653719583)
('acc_train=', 0.82599742923464103, 'acc_val=', 0.83958939711252856)
('Epoch:', 23, 'cost_train=', 0.17714619530098782, 'cost_val=', 0.17487951616446176)
('acc_train=', 0.82683386547224869, 'acc_val=', 0.84056522448857629)
('Epoch:', 24, 'cost_train=', 0.175374156662396, 'cost_val=', 0.17268382509549457)
('acc_train=', 0.83101599131311699, 'acc_val=', 0.84251687924067187)
('Epoch:', 25, 'cost_train=', 0.17322369558470591, 'cost_val=', 0.17055401206016541)
('acc_train=', 0.82620654361588619, 'acc_val=', 0.84349270661671949)
('Epoch:', 26, 'cost_train=', 0.17131450985159191, 'cost_val=', 0.16845848659674326)
('acc_train=', 0.83331618138722019, 'acc_val=', 0.84544436136881507)
('Epoch:', 27, 'cost_train=', 0.16834229337317602, 'cost_val=', 0.16639579335848492)
('acc_train=', 0.83164332594190316, 'acc_val=', 0.8464201887448628)
('Epoch:', 28, 'cost_train=', 0.16638972503798347, 'cost_val=', 0.164379154642423)
('acc_train=', 0.83373438034738834, 'acc_val=', 0.8473960359891255)
('Epoch:', 29, 'cost_train=', 0.16458797880581444, 'cost_val=', 0.16240859031677246)
('acc_train=', 0.83394349898610798, 'acc_val=', 0.8473960359891255)
('Epoch:', 30, 'cost_train=', 0.16234255369220457, 'cost_val=', 0.16048128406206766)
('acc_train=', 0.83289796113967896, 'acc_val=', 0.8473960359891255)
('Epoch:', 31, 'cost_train=', 0.15995593475443978, 'cost_val=', 0.1586072345574697)
('acc_train=', 0.83958938292094654, 'acc_val=', 0.84934769074122118)
('Epoch:', 32, 'cost_train=', 0.15835819712706975, 'cost_val=', 0.15678417682647705)
('acc_train=', 0.84147133997508461, 'acc_val=', 0.85129936536153161)
('Epoch:', 33, 'cost_train=', 0.15666612450565609, 'cost_val=', 0.15499959389368692)
('acc_train=', 0.83812563759940018, 'acc_val=', 0.85129936536153161)
('Epoch:', 34, 'cost_train=', 0.15438338369131088, 'cost_val=', 0.15325050552686056)
('acc_train=', 0.84126222985131405, 'acc_val=', 0.85227519273757935)
('Epoch:', 35, 'cost_train=', 0.15253154827015739, 'cost_val=', 0.15155149002869925)
('acc_train=', 0.84021670051983421, 'acc_val=', 0.85325102011362719)

In [16]:

```
#Test the saved model
no_of_batches_test = int(len(X_test)/batch_size)
sess = tf.Session()
# Load the model
saver = tf.train.Saver()
saver.restore(sess, checkpoint)
state_test = sess.run(initial_state)
avg_cost_test = 0
avg_acc_test = 0
for ii, (x, y) in enumerate(get_batches(X_test, y_test, batch_size), 1):
    _, cost, acc = sess.run([optimizer, loss_function, accuracy], feed_dict={inputs: x,
                                                    targets: y[:, None],keep_prob: 1,initial_state:
tate_test})
    avg_cost_test += cost / no_of_batches_test
    avg_acc_test += acc / no_of_batches_test
print("Test loss",avg_cost_test)
print("Test Accuracy",avg_acc_test)
sess.close()
```

INFO:tensorflow:Restoring parameters from ./model/savedmodel.ckpt
('Test loss', 0.14462399482727051)
('Test Accuracy', 0.85788622498512268)