SDL EXPERIENTIAL LEARNING REPORT

Submitted by

Nisarg Brahmbhatt - 17070122015
Ishansh Sahni - 17070122024
Jack Kalavadia - 17070122025
Dhir Thacker - 17070122019

Under the guidance of

Seema wazarkar
(Department of CS-IT)

Department of Computer Science and Information Technology

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

August 2020- January 2021

Problem Statement:

Twitter Sentimental Analysis using python

What is sentimental analysis?

- Sentiment Analysis is the process of 'Computationally' determining whether a piece of Writing is positive, negative, or neutral.
- It's also known as **opinion mining**, deriving the opinion or attitude of a speaker.

Why sentiment analysis?

- **Business:** In the marketing field, companies use it to develop their strategies, to understand customers' feelings towards products or brands, how people respond to their campaigns or product launches and why consumers don't buy some products.
- **Politics:** In the political field, it is used to keep track of political views, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!
- **Public Actions:** Sentiment analysis is also used to monitor and analyze social phenomena, spot potentially dangerous situations, and determine the general mood of the blogosphere.

Steps to be followed / Algorithm:

Step1: Tokenization (dividing para into different sentences and sentences into words)
Step 2: Cleaning Data (Removes all these unique characters and words which do not add any value to the analytics part)
Step 3: Removing Stopwords (like The, was, he/she) that do not add value to the analytics part.
Step 4: Classification: To classify them as to whether it is a positive/negative/neutral word.

Positive word: We give a score of +1
Negative word: We give a score of -1
Neutral: 0

Step 5: Apply Supervised Algorithm of classification (machine learning)

You train you model with Bag of words or lexicons and test it on the analyzing statement.

Lexicons – Dictionary of a pre-classified set of words.
Once the model is trained, we can perform the test on the analysis statement. An accurate score will be the classification.
If your model is too accurate, then YES (it will be an excellent classification)
Step 6: Calculation (final sentiment score of the statement based on polarity)

Conclusion:

With this assignment, we learned how to perform the sentimental analysis on tweets using twitter's tweepy api. We used the python language and its libraries to perform the analysis part. For this experiment, we used Twitter. Still, this same analysis can also be served on some company's feedback databases and other social networks, which can help to know the public reviews about some particular topic.

Code and output:

Install and import required libraries

```
[35] # Install Libraries
     !pip install textblob
     !pip install tweepy
     !pip install pycountry
     !pip install langdetect

     Requirement already satisfied: textblob in /usr/local/lib/python3.7/dist-packages (0.15.3)
     Requirement already satisfied: nltk>=3.1 in /usr/local/lib/python3.7/dist-packages (from textblob) (3.2.5)
     Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk>=3.1->textblob) (1.15.0)
     Requirement already satisfied: tweepy in /usr/local/lib/python3.7/dist-packages (3.10.0)
     Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.7/dist-packages (from tweepy) (1.15.0)
     Requirement already satisfied: requests[socks]>=2.11.1 in /usr/local/lib/python3.7/dist-packages (from tweepy) (2.23.0)
     Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from tweepy) (1.3.0)
     Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy) (3.0.4)
     Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy) (1.24.3)
     Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy) (2.10)
     Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy) (2020.12.5)
     Requirement already satisfied: PySocks!=1.5.7,>=1.5.6; extra == "socks" in /usr/local/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy) (1.7.1)
     Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->tweepy) (3.1.0)
     Requirement already satisfied: pycountry in /usr/local/lib/python3.7/dist-packages (20.7.3)
     Requirement already satisfied: langdetect in /usr/local/lib/python3.7/dist-packages (1.0.8)
     Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from langdetect) (1.15.0)
```

```
# Import Libraries

from textblob import TextBlob
import sys
import tweepy
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import os
import nltk
import pycountry
import re
import string

from wordcloud import WordCloud, STOPWORDS
from PIL import Image
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from langdetect import detect
from nltk.stem import SnowballStemmer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from sklearn.feature_extraction.text import CountVectorizer
```

```
[37] nltk.download('vader_lexicon')
     nltk.download('stopwords')

     [nltk_data] Downloading package vader_lexicon to /root/nltk_data...
     [nltk_data]   Package vader_lexicon is already up-to-date!
     [nltk_data] Downloading package stopwords to /root/nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!
     True
```

Authentication for Twitter API

```
# Authentication
consumerKey = "TOQQWA6qH9dNKWJ2aIBNUacd3"
consumerSecret = "qsNTNlp20bCSe8sAwW3vdMkxKDgGs2SIhFf9t2Sr5juerIFWpF"
accessToken = "1131902371938750465-ElmXiQ6JBP1APEDf70c4en02P3XYlQ"
accessTokenSecret = "xzVT0HUgZs2Oxf2ctLPRRMPG2NUuGuEeJ82MpO6Gow1vC"

auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
auth.set_access_token(accessToken, accessTokenSecret)
api = tweepy.API(auth)
```

Getting tweets with hashtag or keyword

```
#Sentiment Analysis

def percentage(part,whole):
    return 100 * float(part)/float(whole)

keyword = input("Please enter keyword or hashtag to search: ")
noOfTweet = int(input ("Please enter how many tweets to analyze: "))


tweets = tweepy.Cursor(api.search, q=keyword).items(noOfTweet)
positive = 0
negative = 0
neutral = 0
polarity = 0
tweet_list = []
neutral_list = []
negative_list = []
positive_list = []

for tweet in tweets:

    #print(tweet.text)
    tweet_list.append(tweet.text)
    analysis = TextBlob(tweet.text)
    score = SentimentIntensityAnalyzer().polarity_scores(tweet.text)
    neg = score['neg']
    neu = score['neu']
    pos = score['pos']
    comp = score['compound']
    polarity += analysis.sentiment.polarity

    if neg > pos:
        negative_list.append(tweet.text)
        negative += 1

    elif pos > neg:
        positive_list.append(tweet.text)
        positive += 1

    elif pos == neg:
        neutral_list.append(tweet.text)
        neutral += 1

positive = percentage(positive, noOfTweet)
negative = percentage(negative, noOfTweet)
neutral = percentage(neutral, noOfTweet)
polarity = percentage(polarity, noOfTweet)
positive = format(positive, '.1f')
negative = format(negative, '.1f')
neutral = format(neutral, '.1f')
```

```
Please enter keyword or hashtag to search: symbiosis
Please enter how many tweets to analyze: 200
```

Number of tweets(Positive, Negative, Neutral)

```
[6] #Number of Tweets (Total, Positive, Negative, Neutral)
    tweet_list = pd.DataFrame(tweet_list)
    neutral_list = pd.DataFrame(neutral_list)
    negative_list = pd.DataFrame(negative_list)
    positive_list = pd.DataFrame(positive_list)
    print("total number: ",len(tweet_list))
    print("positive number: ",len(positive_list))
    print("negative number: ", len(negative_list))
    print("neutral number: ",len(neutral_list))
```

```
total number:  1000
positive number:  415
negative number:  345
neutral number:  240
```

Tweet list

```
tweet_list
```

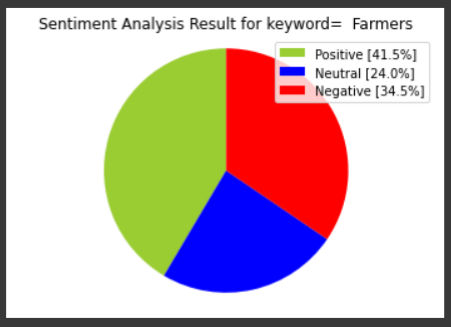
	0
0	RT @LeviSquad_SK: @VarshaEGaikwad @INCIndia @I...
1	RT @MdLal67739928: Central Govt gives a befitt...
2	@FishforBrains3 @TalkSalmon So you aren't admi...
3	NHAI suffers Rs. 814 crore toll revenue loss i...
4	Farmers had a long due demand of open market a...
...	...
995	RT @MdLal67739928: And the UK parliament wante...
996	#farmerprotesthijacked by the anti-national el...
997	RT @NavdeepSingh_7: Stand with what is right, ...
998	RT @vippaluri: @vivekvenkatswam •This is the h...
999	RT @LeviSquad_SK: @VarshaEGaikwad @INCIndia @I...

1000 rows × 1 columns

Pie-cart

```
[8] #Creating PieCart

    labels = ['Positive ['+str(positive)+'%]' , 'Neutral ['+str(neutral)+'%]','Negative ['+str(negative)+'%]']
    sizes = [positive, neutral, negative]
    colors = ['yellowgreen', 'blue','red']
    patches, texts = plt.pie(sizes,colors=colors, startangle=90)
    plt.style.use('default')
    plt.legend(labels)
    plt.title("Sentiment Analysis Result for keyword=  "+keyword+"" )
    plt.axis('equal')
    plt.show()
```



Cleaning Tweets to Analyze statement

```
[12] #Cleaning Text (RT, Punctuation etc)

     #Creating new dataframe and new features
     tw_list = pd.DataFrame(tweet_list)
     tw_list["text"] = tw_list[0]

     #Removing RT, Punctuation etc
     remove_rt = lambda x: re.sub('RT @\w+: '," ",x)
     rt = lambda x: re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",x)
     tw_list["text"] = tw_list.text.map(remove_rt).map(rt)
     tw_list["text"] = tw_list.text.str.lower()
     tw_list.head(10)
```

	0	text
0	RT @LeviSquad_SK: @VarshaEGaikwad @INCIndia @I...	hey can you tell
1	RT @MdLal67739928: Central Govt gives a befitt...	central govt gives a befitting reply to rahul...
2	@FishforBrains3 @TalkSalmon So you aren't admi...	so you aren t admitting that the figures f...
3	NHAI suffers Rs. 814 crore toll revenue loss i...	nhai suffers rs 814 crore toll revenue loss i...
4	Farmers had a long due demand of open market a...	farmers had a long due demand of open market a...
5	RT @Ruptly: Hundreds of farmers marched throug...	hundreds of farmers marched through asuncion...
6	RT @RobTheRich001: Massive direct action takin...	massive direct action taking place today by f...
7	RT @jatt_sandhusaab: Please see Point No 6 in ...	please see point no 6 in demand charter place...
8	RT @MeghUpdates: FIR registered against Allege...	fir registered against alleged farmer leader ...
10	•The innocent farmers and youngsters are misle...	the innocent farmers and youngsters are misle...

Sentiment Analyse

```
[13] #Calculating Negative, Positive, Neutral and Compound values

     tw_list[['polarity', 'subjectivity']] = tw_list['text'].apply(lambda Text: pd.Series(TextBlob(Text).sentiment))
     for index, row in tw_list['text'].iteritems():
         score = SentimentIntensityAnalyzer().polarity_scores(row)
         neg = score['neg']
         neu = score['neu']
         pos = score['pos']
         comp = score['compound']
         if neg > pos:
             tw_list.loc[index, 'sentiment'] = "negative"
         elif pos > neg:
             tw_list.loc[index, 'sentiment'] = "positive"
         else:
             tw_list.loc[index, 'sentiment'] = "neutral"
         tw_list.loc[index, 'neg'] = neg
         tw_list.loc[index, 'neu'] = neu
         tw_list.loc[index, 'pos'] = pos
         tw_list.loc[index, 'compound'] = comp

     tw_list.head(10)
```

	0	text	polarity	subjectivity	sentiment	neg	neu	pos	compound
0	RT @LeviSquad_SK: @VarshaEGaikwad @INCIndia @I...	hey can you tell	0.000000	0.0000	neutral	0.000	1.000	0.000	0.0000
1	RT @MdLal67739928: Central Govt gives a befitt...	central govt gives a befitting reply to rahul...	0.050000	0.3250	neutral	0.000	1.000	0.000	0.0000
2	@FishforBrains3 @TalkSalmon So you aren't admi...	so you aren t admitting that the figures f...	0.000000	0.0000	positive	0.000	0.820	0.180	0.5106
3	NHAI suffers Rs. 814 crore toll revenue loss i...	nhai suffers rs 814 crore toll revenue loss i...	-0.362500	0.5375	negative	0.363	0.637	0.000	-0.7506
4	Farmers had a long due demand of open market a...	farmers had a long due demand of open market a...	-0.058333	0.4250	positive	0.063	0.753	0.184	0.4404
5	RT @Ruptly: Hundreds of farmers marched throug...	hundreds of farmers marched through asuncion...	0.000000	0.0000	negative	0.209	0.791	0.000	-0.4019
6	RT @RobTheRich001: Massive direct action takin...	massive direct action taking place today by f...	0.225000	0.6000	positive	0.066	0.833	0.101	0.2023
7	RT @jatt_sandhusaab: Please see Point No 6 in ...	please see point no 6 in demand charter place...	0.000000	0.0000	negative	0.320	0.599	0.081	-0.7003
8	RT @MeghUpdates: FIR registered against Allege...	fir registered against alleged farmer leader ...	-0.100000	0.1000	positive	0.085	0.798	0.117	0.1779
10	•The innocent farmers and youngsters are misle...	the innocent farmers and youngsters are misle...	0.200000	0.4000	positive	0.115	0.747	0.138	0.1027

```
[14] #Creating new data frames for all sentiments (positive, negative and neutral)

     tw_list_negative = tw_list[tw_list["sentiment"]=="negative"]
     tw_list_positive = tw_list[tw_list["sentiment"]=="positive"]
     tw_list_neutral = tw_list[tw_list["sentiment"]=="neutral"]
```

```
[15] #Function for count_values_in single columns

     def count_values_in_column(data,feature):
         total=data.loc[:,feature].value_counts(dropna=False)
         percentage=round(data.loc[:,feature].value_counts(dropna=False,normalize=True)*100,2)
         return pd.concat([total,percentage],axis=1,keys=['Total','Percentage'])
```

```
[16] #Count_values for sentiment
     count_values_in_column(tw_list,"sentiment")
```

	Total	Percentage
positive	320	42.50
negative	274	36.39
neutral	159	21.12

Creating Wordcloud

- Positive

```
#Function to Create Wordcloud
comment_words = ''
stopwords = set(STOPWORDS)
for val in tw_list_positive.text:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    comment_words += " ".join(tokens)+" "

wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```


farmer
farmers protest
farmers forced
farmers free
good farmers
support farmer
away prices
forced sell
throw away
sell produce
today
prices good
market help
bharat bandh
help provide
due demand
demand open
free sell
produce middlemen
pillar india
open market
long due
provide competitive
farmers long
middlemen throw
competitive price
supporting farmers
sikh community
community particular
punjab sikh
peacefully don
per market
bharatbandh
look innocent
innocent ones
don come
made self
hailed panacea
particular always
brilliant analogy
price per
free exercise
field agriculture
agriculture farmers
getting best
always strongest
protest made
exercise democratic
india shining
farmers punjab
timeforfarmreforms
baseless allegation
oppositions baseless
bjp mp
stand
mp tejasvi
innocent peaceful
producers now
farmbill rightly
shining growth
will
self goal
middlemen frustrated
strongest pillar
ones marching
frustrated original
now getting
come way
farmers poor
poor innocent
panacea field
surya parliament
parliament oppositions
higher msp
msp indian
marching peacefully
best price
let unite
forth bjp
fake farmers
original producers
peaceful farmers
put forth
rightly hailed
exactly middlemen
support
tejasvi surya
govt higher
goal govt
analogy put

- Negative

```
[18] comment_words = ''
     stopwords = set(STOPWORDS)
     for val in tw_list_negative.text:

         # typecaste each val to string
         val = str(val)

         # split the value
         tokens = val.split()

         # Converts each token into lowercase
         for i in range(len(tokens)):
             tokens[i] = tokens[i].lower()

         comment_words += " ".join(tokens)+" "

     wordcloud = WordCloud(width = 800, height = 800,
                     background_color ='white',
                     stopwords = stopwords,
                     min_font_size = 10).generate(comment_words)

     # plot the WordCloud image
     plt.figure(figsize = (8, 8), facecolor = None)
     plt.imshow(wordcloud)
     plt.axis("off")
     plt.tight_layout(pad = 0)

     plt.show()
```


farmer
farmers protest
protesting farmers
timeforfarmreform
police barricades
barricades thrown
drain protesting
thrown drain
ghazipur border
near ghazipur
border goons
farmers near
farmersprotest
goons put
protest will
khalistani supporters
punjab haryana
food
delhi
sikh people
million
slogans
fake farmers
farmers poor
defame india
conspiracy targeted
keep indian
rs crore
go long
stall india
arrested anti
act fast
modi will
time
rajasthan due
india progress
law
govt india
poor farm
accepting now
corporates procure
placed farmers
abused called
will die
protesting khalistani
called farmers
nhai suffers
india int
road
group govt
farmers demanded
fast farmlaws
demand charter
blocked
change worsening
bjp
crore toll
country
toll revenue
protest read
read modi
india climate
farmlaws go
used amp
people
int community
bku leader
india needs
suffers rs
revenue loss
progress keep
protest ones
farm leaders
india release
release arrested
worsening india
procure wheat
international conspiracy
produce accepting
true depiction
needs act
targeted defame
farmers used
conspiracy stall
will end
depiction urbannaxalism
end threatening
deeper conspiracy farmers deeper
charter placed
uk parliament
climate change
supporters denied
demanded corporates
leaders nothing
urbannaxalism farmers
parliament worrying
die protest
haryana rajasthan
loss punjab
wheat produce
sheer level
handling fake
worrying indian
protest sheer
amp abused

- Neutral

```
[19] comment_words = ''
     stopwords = set(STOPWORDS)
     for val in tw_list_neutral.text:

         # typecaste each val to string
         val = str(val)

         # split the value
         tokens = val.split()

         # Converts each token into lowercase
         for i in range(len(tokens)):
             tokens[i] = tokens[i].lower()

         comment_words += " ".join(tokens)+" "

     wordcloud = WordCloud(width = 800, height = 800,
                     background_color ='white',
                     stopwords = stopwords,
                     min_font_size = 10).generate(comment_words)

     # plot the WordCloud image
     plt.figure(figsize = (8, 8), facecolor = None)
     plt.imshow(wordcloud)
     plt.axis("off")
     plt.tight_layout(pad = 0)

     plt.show()
```


will now
trudeau police called arhtiyas
farmers andhra
day rioters
blockade justin
liberate farmers
raise voice
quality cotton
now dal
central govt
dal national
indigenous people
portrayed bjp
views farmlaws2020
rail blockade
short memory
singh sirsa
corporation india
deepsidhu portrayed
bku demands
bank transfer
pm
brothers understood
falsely believe
cotton farmers
bill will
donation sfj
will lead
earned rs
confirmed received
understood law
nationwide shutdown
farm bill
farmer
indigenous
march setup
1st march
man now
feb
people canada
act abolish
image
republic day
first deepsidhu
selling qual
blast past
farmlaws2020 amp
canada rail
police dismantled
timeforfarmreformsrights despite
farmersprotest
cotton corporation
amp
selling quality
gives befitting
manjinder singh
past bku
agents earned
new farm
direct bank
amp future
andhra pradesh selling
stand
fixed per
reply rahul
receive msp
abolish essential
believe new
now receive
amp requesting
bjp man
despite much
much evidence
india fixed
say
direct
msp
arhtiyas abolish
bharatbandh
punjab agents
abolish apmc
farmbills2020
future notobjp
per quintal
arhtiyas
lead suppression
national spokesperson
indians removing
received lakh
apmc act
jibe msp
govt gives
happened republic
bharat bandh
setup commission
justin trudeau
people falsely
agents called
quintal quality
befitting reply
rahul jibe
agents punjab
sfj 1st
timeforfarmreformss agents
expressing views
demands liberate
amp timeforfarmreformss
spokesperson manjinder
picture condition
un confirmed

Calculating tweet's length & word count

```
[20] #Calculating tweet's length and word count
     tw_list['text_len'] = tw_list['text'].astype(str).apply(len)
     tw_list['text_word_count'] = tw_list['text'].apply(lambda x: len(str(x).split()))
```

```
[21] round(pd.DataFrame(tw_list.groupby("sentiment").text_len.mean()),2)
```

sentiment	text_len
negative	114.89
neutral	106.45
positive	114.27

```
[22] round(pd.DataFrame(tw_list.groupby("sentiment").text_word_count.mean()),2)
```

sentiment	text_word_count
negative	18.06
neutral	14.11
positive	18.31

- Removing Punctuation and Stopwords
- Applying Tokenization and Stemmer

```
[23] #Removing Punctuation
     def remove_punct(text):
         text  = "".join([char for char in text if char not in string.punctuation])
         text = re.sub('[0-9]+', '', text)
         return text

     tw_list['punct'] = tw_list['text'].apply(lambda x: remove_punct(x))
```

```
[24] #Appliyng tokenization
     def tokenization(text):
         text = re.split('\W+', text)
         return text

     tw_list['tokenized'] = tw_list['punct'].apply(lambda x: tokenization(x.lower()))
```

```
[25] #Removing stopwords
     stopword = nltk.corpus.stopwords.words('english')
     def remove_stopwords(text):
         text = [word for word in text if word not in stopword]
         return text

     tw_list['nonstop'] = tw_list['tokenized'].apply(lambda x: remove_stopwords(x))
```

```
[26] #Appliyng Stemmer
     ps = nltk.PorterStemmer()

     def stemming(text):
         text = [ps.stem(word) for word in text]
         return text

     tw_list['stemmed'] = tw_list['nonstop'].apply(lambda x: stemming(x))
```

```
tw_list.head()
```

	0	text	polarity	subjectivity	sentiment	neg	neu	pos	compound	text_len	text_word_count	punct	tokenized	nonstop	stemmed
0	RT @LeviSquad_SK: @VarshaEGaikwad @INCIndia @I...	hey can you tell	0.000000	0.0000	neutral	0.000	1.000	0.000	0.0000	34	4	hey can you tell	[, hey, can, you, tell,]	[, hey, tell,]	[, hey, tell,]
1	RT @MdLal67739928: Central Govt gives a befitt...	central govt gives a befitting reply to rahul...	0.050000	0.3250	neutral	0.000	1.000	0.000	0.0000	122	21	central govt gives a befitting reply to rahul...	[, central, govt, gives, a, befitting, reply, ...	[, central, govt, gives, befitting, reply, rah...	[, central, govt, give, befit, repli, rahul, j...
2	@FishforBrains3 @TalkSalmon So you aren't admi...	so you aren t admitting that the figures f...	0.000000	0.0000	positive	0.000	0.820	0.180	0.5106	98	17	so you aren t admitting that the figures f...	[, so, you, aren, t, admitting, that, the, fig...	[, admitting, figures, chemical, use, amp, mor...	[, admit, figur, chemic, use, amp, mortal, bl,]
3	NHAI suffers Rs. 814 crore toll revenue loss i...	nhai suffers rs 814 crore toll revenue loss i...	-0.362500	0.5375	negative	0.363	0.637	0.000	-0.7506	103	16	nhai suffers rs crore toll revenue loss in p...	[nhai, suffers, rs, crore, toll, revenue, loss...	[nhai, suffers, rs, crore, toll, revenue, loss...	[nhai, suffer, rs, crore, toll, revenu, loss, ...
4	Farmers had a long due demand of open market a...	farmers had a long due demand of open market a...	-0.058333	0.4250	positive	0.063	0.753	0.184	0.4404	118	23	farmers had a long due demand of open market a...	[farmers, had, a, long, due, demand, of, open,...	[farmers, long, due, demand, open, market, wou...	[farmer, long, due, demand, open, market, woul...