AIT-664 May 14, 2018

# DISASTER MANAGEMENT

K. SIVA NAGA LAKSHMI

G01099587

***Goal***: Hands-on experience to process data, to extract information, and discover patterns or knowledge using data mining method

**MILESTONE1: Data Acquisition**

| MESSAGE | DATETIME | LATITUDE | LONGITUDE |
|---|---|---|---|
| @Zuora wants to help @Network4Good with Hurricane Relief. Text SANDY to 80888 & donate $10 to @redcross @AmeriCares & @SalvationArmyUS #help | 2012-10-30 22:15:41 | 37.4777 | -122.223 |

i. MESSAGE: message content from social media
- String

ii. DATETIME: date and time of message arrival
- Date-Time

iii. LATITUDE
- Numeric (or 'Null' if no such information is unavailable in a tweet)

iv. LONGITUDE
- Numeric (or 'Null' if no such information is unavailable in a tweet)

a) I read the data using python from the JSON file. After reading the data checked for null values in the data set.

```
data=pd.read_json("http://ist.gmu.edu/~hpurohit/courses/ait690-proj-data-spring17.json")
data.head()
```

| | DATETIME | DOCUMENT_ID | LATITUDE | LONGITUDE | MESSAGE |
|---|---|---|---|---|---|
| 0 | 2012-10-30 22:15:41 | 263403828328665088 | 37.4777 | -122.223 | @Zuora wants to help @Network4Good with Hurric... |
| 1 | 2012-11-01 23:11:04 | 264142543883739136 | 36.7783 | -119.418 | @ztrip please help spread the good word on hel... |
| 2 | 2012-11-04 16:07:02 | 265122997348753408 | 10000.0000 | 10000.000 | #ZSwaggers @Zendaya96 did this,you should too-... |
| 3 | 2012-11-05 17:04:19 | 265499798952628224 | 10000.0000 | 10000.000 | @Zendaya96 u have inspired me girl! So can eve... |
| 4 | 2012-11-04 18:41:56 | 265161977662435328 | 10000.0000 | 10000.000 | ''@Zendaya96 let's help the Hurricane Sandy vi... |

```
data.isnull().any()
```

```
DATETIME        False
DOCUMENT_ID     False
LATITUDE        False
LONGITUDE       False
MESSAGE         False
dtype: bool
```

## MILESTONE 2: DATA PREPROCESSING

In this mile stone I replaced the 10000 values with using mean of respective columns.

```
data["LATITUDE"]=data["LATITUDE"].replace(10000.0000,0)
data["LONGITUDE"]=data["LONGITUDE"].replace(10000.0000,0)
```

```
mean_latitude=data["LATITUDE"].mean()
mean_latitude
```

```
22.535964570972713
```

```
mean_longitude=data["LONGITUDE"].mean()
mean_longitude
```

```
-46.63928404976049
```

```
data["LATITUDE"]=data["LATITUDE"].replace(0,mean_latitude)
data["LONGITUDE"]=data["LONGITUDE"].replace(0,mean_longitude)
```

```
data["LONGITUDE"].head()
```

```
0    -122.223000
1    -119.418000
2     -46.639284
3     -46.639284
4     -46.639284
Name: LONGITUDE, dtype: float64
```

```
data["LATITUDE"].head()
```

```
0     37.477700
1     36.778300
2     22.535965
3     22.535965
4     22.535965
Name: LATITUDE, dtype: float64
```

I separated the DATETIME column as separate two columns.

```
new_dates, new_times = zip(*[(d.date(), d.time()) for d in data["DATETIME"]])
data = data.assign(DATE=new_dates, TIME=new_times)
```

```
data.head()
```

| | DATETIME | DOCUMENT_ID | LATITUDE | LONGITUDE | MESSAGE | DATE | TIME |
|---|---|---|---|---|---|---|---|
| 0 | 2012-10-30 22:15:41 | 263403828328665088 | 37.477700 | -122.223000 | @Zuora wants to help @Network4Good with Hurric... | 2012-10-30 | 22:15:41 |
| 1 | 2012-11-01 23:11:04 | 264142543883739136 | 36.778300 | -119.418000 | @ztrip please help spread the good word on hel... | 2012-11-01 | 23:11:04 |
| 2 | 2012-11-04 16:07:02 | 265122997348753408 | 22.535965 | -46.639284 | #ZSwaggers @Zendaya96 did this,you should too-... | 2012-11-04 | 16:07:02 |
| 3 | 2012-11-05 17:04:19 | 265499798952628224 | 22.535965 | -46.639284 | @Zendaya96 u have inspired me girl! So can eve... | 2012-11-05 | 17:04:19 |
| 4 | 2012-11-04 18:41:56 | 265161977662435328 | 22.535965 | -46.639284 | ''@Zendaya96 let's help the Hurricane Sandy vi... | 2012-11-04 | 18:41:56 |

I cleaned the message column, removed stop words using the library and then used lemmatize and stemmer functions for the message to be proper. Removed two letter words as well.

```
corpus=[]
for i in range(0,len(d)):
    msg=d[i]
    msg= decontracted(msg)
    msg=msg.lower()
    msg=remove_characters(msg)
    msg=url(msg)
    msg=word_tokenize(msg)
    msg= [j for j in msg if j.isalpha()]
    msg = [ps.stem(word) for word in msg if not word in set(stwords)]
    msg = [lemmatizer.lemmatize(word, pos= "a") for word in msg ]
    msg=[spell(k) for k in msg]
    msg=  [Remove_len_two_words(l) for l in msg]
    msg = ' '.join(msg)
    corpus.append(msg)
```

```
corpus
```

```
['want help hurricane relief text Sandi donat',
 'pleas help spread good word help victim hurricane Sandi',
 'convoy send hurricane Sandi relief',
 ' inspire everyone pleas donat hurricane Sandi convoy not  want',
 'let help hurricane Sandi donat goe relief syria',
 'help hurricane Sandi victim text convoy donat pleas',
 'help hurricane Sandi text convoy spread world',
 'help hurricane Sandi victim text convoy donat goe relief effort',
 'person discount code help hurricane relief effort',
 'help donat american red cross hurricane Sandi relief effort text recross',
 'already phone affect',
 'text donat hurricane relief fund visit',
 'send hurricane Sandi relief supply via amazon wish',
 'use Ithun support hurricane Sandi relief',
```

After cleaning the message, I removed extra columns and saved it as CSV file.

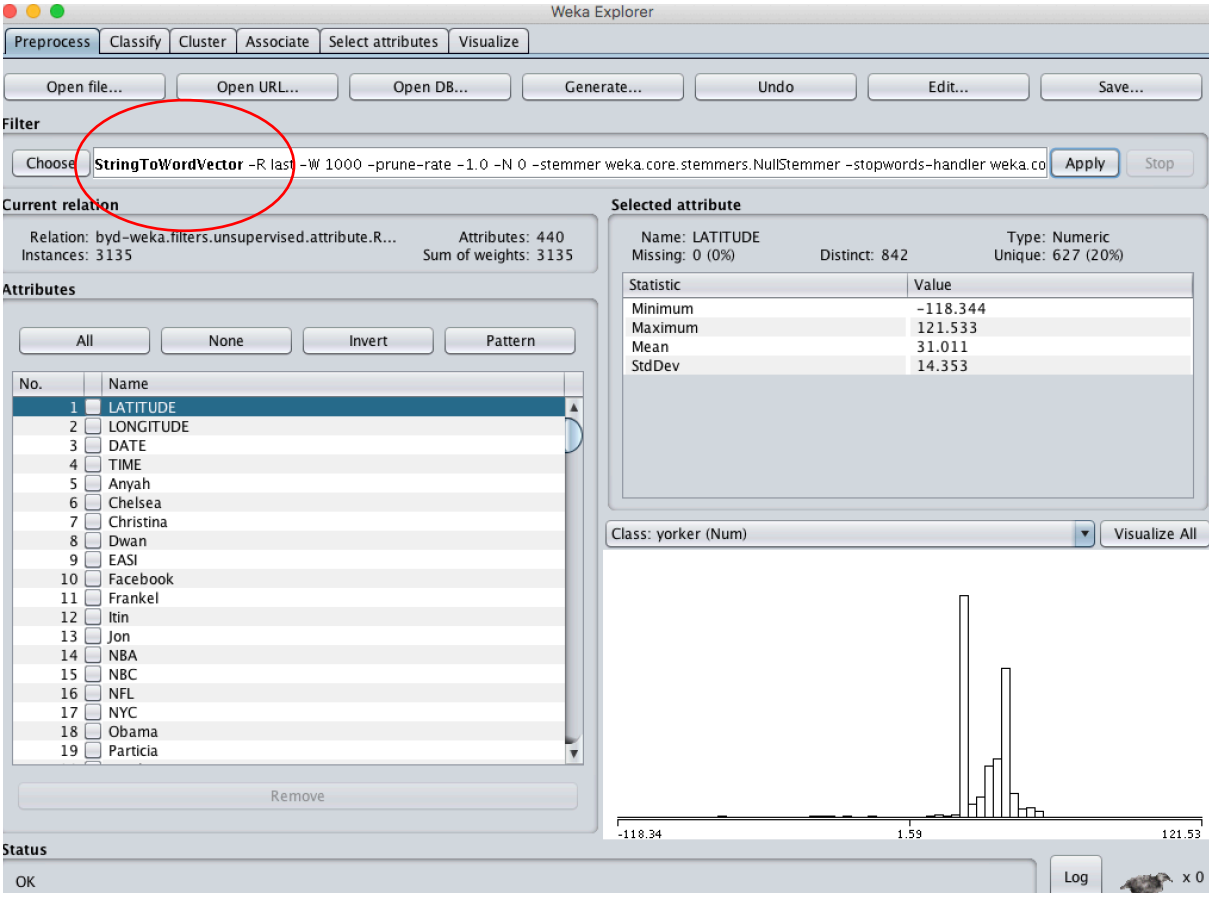
| A | B | C | D | E | F |
|---|---|---|---|---|---|
| | LATITUDE | LONGITUDE | DATE | TIME | MESSAGE |
| 0 | 37.4777 | -122.223 | 10/30/12 | 22:15:41 | want help hurricane relief text Sandi donat |
| 1 | 36.7783 | -119.418 | 11/1/12 | 23:11:04 | pleas help spread good word help victim hurricane Sandi |
| 2 | 22.5359646 | -46.639284 | 11/4/12 | 16:07:02 | convoy send hurricane Sandi relief |
| 3 | 22.5359646 | -46.639284 | 11/5/12 | 17:04:19 | inspire everyone pleas donat hurricane Sandi convoy not want |
| 4 | 22.5359646 | -46.639284 | 11/4/12 | 18:41:56 | let help hurricane Sandi donat goe relief syria |
| 5 | 46.2276 | 2.21375 | 11/5/12 | 16:52:22 | help hurricane Sandi victim text convoy donat pleas |
| 6 | 34.0522 | -118.243 | 11/5/12 | 16:49:40 | help hurricane Sandi text convoy spread world |
| 7 | 22.5359646 | -46.639284 | 11/4/12 | 20:42:18 | help hurricane Sandi victim text convoy donat goe relief effort |
| 8 | 34.0522 | -118.243 | 10/30/12 | 21:02:03 | person discount code help hurricane relief effort |
| 9 | 25.6675 | -80.3589 | 11/5/12 | 22:17:12 | help donat american red cross hurricane Sandi relief effort text recross |
| 10 | 42.3302 | -83.0459 | 10/30/12 | 23:28:59 | already phone affect |
| 11 | 40.7561 | -73.987 | 11/5/12 | 17:58:35 | text donat hurricane relief fund visit |
| 12 | 34.0522 | -118.243 | 11/5/12 | 19:26:44 | send hurricane Sandi relief supply via amazon wish |
| 13 | 22.5359646 | -46.639284 | 11/1/12 | 1:43:21 | use Ithun support hurricane Sandi relief |

## MILESTONE 3: MINING TOOL PREPARATION

In this milestone, the file needed to be load in Weka, so I converted the CSV to arff using R.

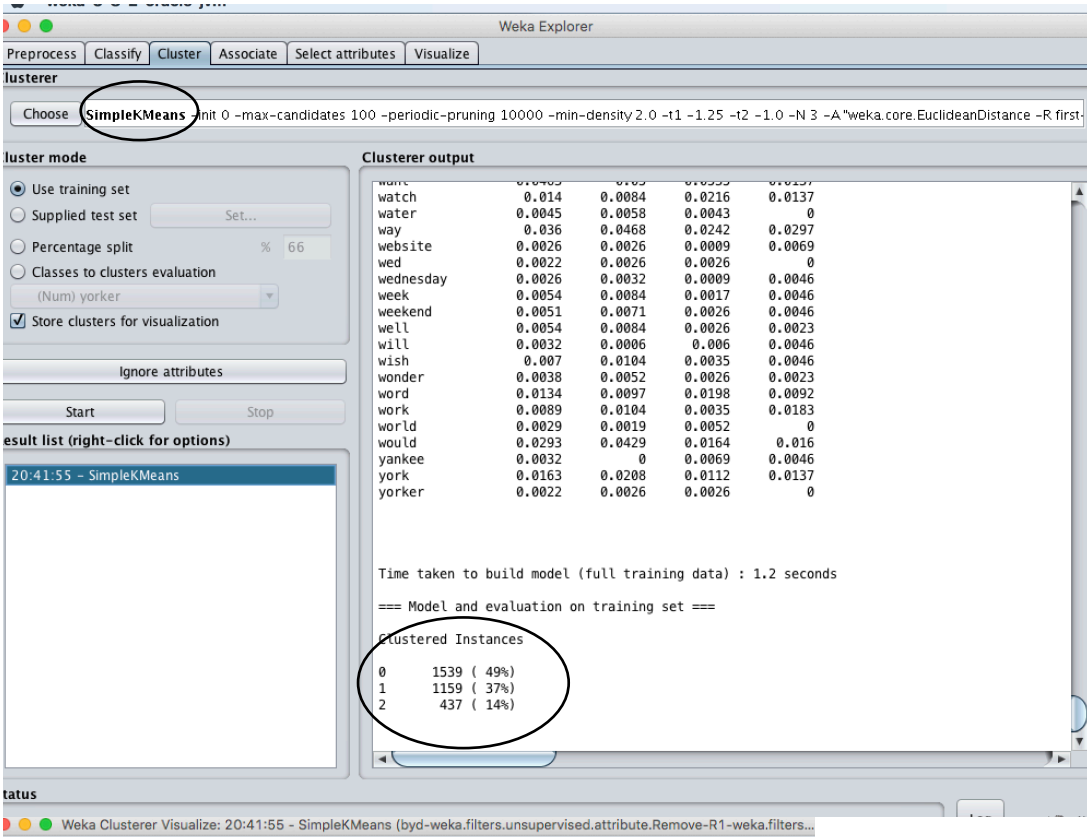
```
setwd('/Users/lakshmi_shetty/Desktop/664-project')
getwd()
Messagedata<-read.csv('./Data_message.csv')
View(Messagedata)
library(dplyr)
library(lubridate)
library(foreign)

byd = read.csv('Data_message.csv')

byd %>% glimpse()

byd = byd %>% mutate(tradeDate = as.Date(tradeDate))

write.arff(byd, file='Finaldata.arff')
```

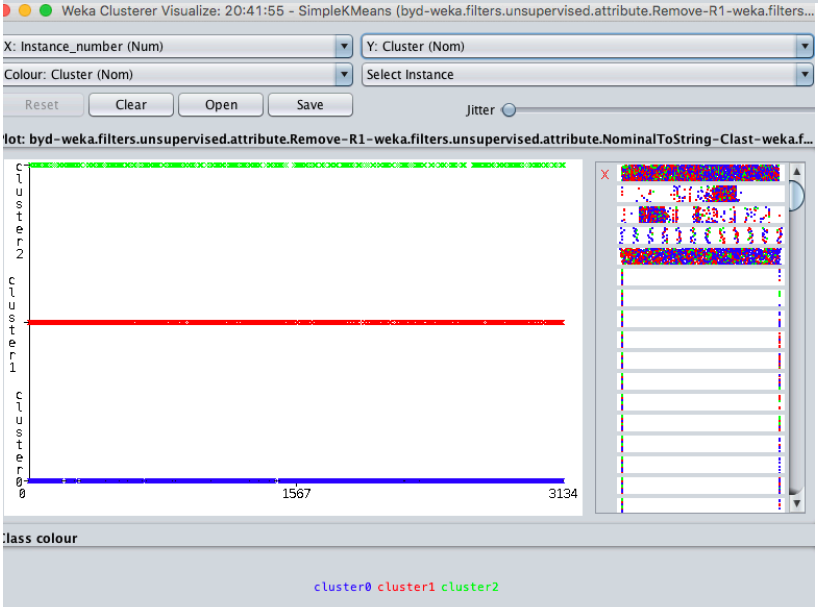
After converting to arff file, as the message is nominal. Using unsupervised learning and attributes I applied the nominal to string and string to word vector filter.



## MILESTONE 4: CLUSTERING ANALYSIS

In this milestone, used simple K- Means algorithm and applied on the message column having clusters as K=3.




As we can see in the above picture we have 3 clusters. Inter cluster distance is maximized and intra cluster distance is minimized.

**MILESTONE 5: VISUALIZATION**

In this milestone, for visualizing the given information I converted the output of arff to CSV file.

| E | F | G |
|---|---|---|
| cluster | MESSAGE | |
| cluster2 | want help hurricane relief text Sandi donat | |
| cluster1 | pleas help spread good word help victim hurricane Sandi | |
| cluster1 | convoy send hurricane Sandi relief | |
| cluster2 | inspire everyone pleas donat hurricane Sandi convoy not want | |
| cluster2 | let help hurricane Sandi donat goe relief syria | |
| cluster2 | help hurricane Sandi victim text convoy donat pleas | |
| cluster1 | help hurricane Sandi text convoy spread world | |
| cluster2 | help hurricane Sandi victim text convoy donat goe relief effort | |
| cluster1 | person discount code help hurricane relief effort | |
| cluster3 | help donat american red cross hurricane Sandi relief effort text recross | |
| cluster1 | already phone affect | |
| cluster2 | text donat hurricane relief fund visit | |
| cluster1 | send hurricane Sandi relief supply via amazon wish | |
| cluster1 | use Ithun support hurricane Sandi relief | |
| cluster3 | use Ithun support hurricane Sandi american red cross | |

After getting CSV file, made word cloud for each cluster using only top 100 words and they are explained further using DIKW.

**Word Cloud for Top-100 words**

Cluster-1


Data:

Affect, people, cloth, blood, money, concert, money, funds, New York, new jersey, needs, prayers.

Information:

New York and new jersey people need prayers. Affected from hurricane people need cloth, blood and money. Concert for raising money, funds for people.

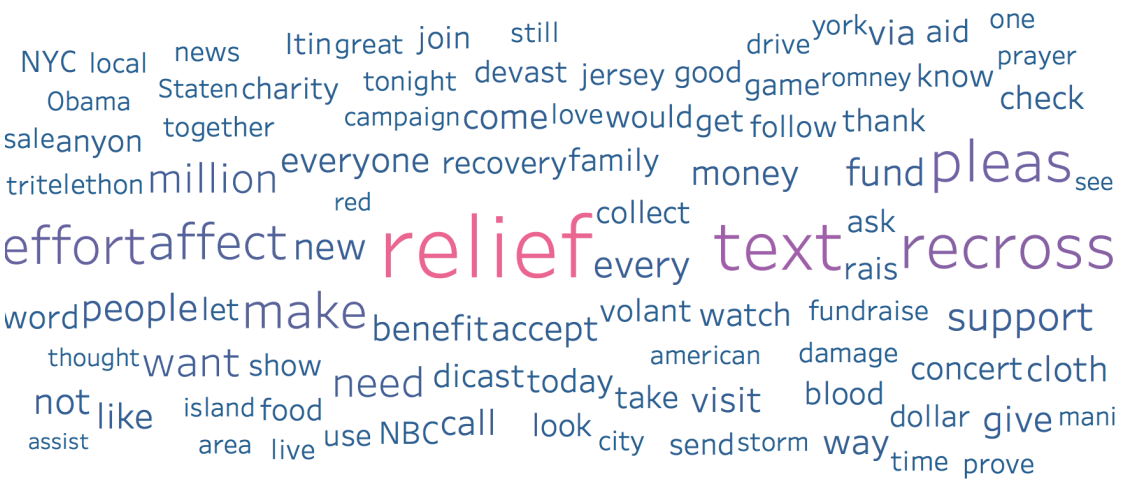
Knowledge:

People Affected by hurricane sandy at New York and New jersey are in need for the cloth, blood and money. Money is collected through the concerts and funds are being collected. And they need everyone's prayers.

Wisdom:

The outcome from the given cluster is needs for the people who are affected due to hurricane.

Cluster-2




Data:

Americans, affect, people, devast, charity, campaign, NBC, area, live, show, damage, storms, effort, New York.

Information:

NBC channel showed live areas damage and the storms.
Americans are the most people who got affected and devasted.
Charities are making efforts by doing the campaigns for the people in New York.

Knowledge:

The news channel was giving live coverage of the hurricane sandy damages occurred due to storms and showed that many people who got affected was Americans'. Charities are conducting the campaigns for the people in NY and NJ.

Wisdom:

The result we can conclude from this cluster was News channels was giving an entire updates and live news was shown from hurricane sandy.

Cluster-3


Data:

Red cross, send, text, help, aid, collect, assist, effort, donation, family, millions, money.

Information:

Red cross is making efforts to help the people. People donating millions of money to red cross for the hurricane affected people. People affected are sending text need help.

Knowledge:

Red cross is helping the people who got affected from the hurricane, by the money which was collected by the donations and was receiving the text continuously to need help.

Wisdom:

Result we can conclude from this cluster is people were aware of the disaster and how donation and how help is provided to people.

#TF-IDF Implementation/Top-10 representative word count for each cluster.

**Cluster-1**

```
#cluster-1
Clus1<- Corpus(VectorSource(Finaldata$Cluster.1))
data1 <- TermDocumentMatrix(Clus1)
matr<- as.matrix(data1)
arrange <- sort(rowSums(matr),decreasing=TRUE)
dc1<- data.frame(word = names(arrange),freq=arrange)
head(dc1, 10)
```

```
               word freq
hurricane hurricane 1467
sandi         sandi  960
help           help  739
relief       relief  579
victim       victim  402
effort       effort  220
donat         donat  157
affect       affect  132
pleas         pleas  108
want           want   79
```

**Cluster-2**

```
#cluster-2
Clus2<- Corpus(VectorSource(Finaldata$cluster.2))
data2 <- TermDocumentMatrix(Clus2)
matri2<- as.matrix(data2)
arrange2 <- sort(rowSums(matri2),decreasing=TRUE)
dc2<- data.frame(word = names(arrange2),freq=arrange2)
head(dc2, 10)
```

```
               word freq
hurricane hurricane 1120
donat         donat  931
sandi         sandi  718
help           help  465
relief       relief  461
victim       victim  310
text           text  259
recross     recross  202
pleas         pleas  162
effort       effort  133
```

**Cluster-3**

```
Clus3<- Corpus(VectorSource(Finaldata$cluster.3))
data3 <- TermDocumentMatrix(Clus3)
matri3<- as.matrix(data3)
arrange3 <- sort(rowSums(matri3),decreasing=TRUE)
dc3<- data.frame(word = names(arrange3),freq=arrange3)
head(dc3, 10)
```

```
                word freq
red              red  449
hurricane hurricane  421
cross          cross  419
donat          donat  283
sandi          sandi  279
help            help  161
relief       relief  154
american   american  112
victim       victim   80
pleas         pleas   59
```

Insights for clusters using top 10 words

Cluster-1: refers to hurricane sandy has affected the people donate and help the victims.

Cluster-2: refers to red cross is receiving lot of text message help needed from the hurricane sandy and donations to the victims.

Cluster-3: Americans are the most people who got affected due to hurricane sandy, they are victims of it and help is needed to them.

**Functions defined in the Milestone 2 (Code only)**

These functions I defined in the Milestone 2 to clean the message, I attached the screenshots of defined function here.

```
def decontracted(phrase):
    phrase = re.sub(r"n\'t", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase
```

```
def  remove_characters(sentence):

        y=sentence.split(" ")
#          print(y)
        list2=[]
        for i in y:
            i=i.lower()
            if re.search('[^a-zA-Z0-9.$]',i):
                pass
            else:
                list2.append(i)
#         print(list2)
        x=" ".join(list2)
        return x
```

```
from autocorrect import spell
```

```
def url(msg):
    msg= re.sub(r'\w+:\/{2}[\d\w-]+(\.[\d\w-]+)*(?:(?:\/[^\s/]*))*', '', msg)
    return msg
```

```
def Remove_len_two_words(msg):
    msg= re.sub(r'\b\w{1,2}\b', '', msg)
    return msg
```

```
ps = PorterStemmer()
lemmatizer = WordNetLemmatizer()
stwords = stopwords.words('english')
stwords.remove('not')
stwords.remove('no')
```