In [1]:

```
import csv

import json

import pandas as pd

import numpy as np

import seaborn as sns
sns.set()

import random

import re

from my_nlp import *
```

Read csv

In [2]:

```
DarpanFcraDF = pd.read_csv('Darpan21FCRA.csv', dtype = {'S.No.': 'string', 'Registration': 'string'})
```

Generate 10k random sample

In [3]:

```
DarpanFcraDF = DarpanFcraDF.sample(n = 10000)
DarpanFcraDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10000 entries, 66103 to 38780
Data columns (total 49 columns):
 #   Column                        Non-Null Count
Dtype
---  ------                        --------------
-----
 0   Name                          10000 non-null
object
 1   ngo url                       2373 non-null
object
 2   Mobile                        9998 non-null
float64
 3   UniqueID                      10000 non-null
object
 4   Off phone1                    1483 non-null
object
 5   Email                         10000 non-null
object
 6   Major Activities1             7508 non-null
object
 7   operational states db         7901 non-null
object
 8   issues working db             7936 non-null
object
 9   operational district db       7901 non-null
object
 10  fcrano                        2002 non-null
object
 11  nr regNo                      10000 non-null
object
 12  nr add                        10000 non-null
object
 13  nr orgName                    10000 non-null
object
 14  ngo reg date                  10000 non-null
object
 15  nr actName                    9902 non-null
object
 16  nr city                       9991 non-null
object
 17  TypeDescription               10000 non-null
```

```
object
 18  StateName                   10000 non-null
object
 19  president name              5233 non-null
object
 20  president email             5233 non-null
object
 21  president mobile            5233 non-null
float64
 22  Chairman name               2710 non-null
object
 23  Chairman email              2709 non-null
object
 24  Chairman mobile             2708 non-null
float64
 25  Secretary name              6684 non-null
object
 26  Secretary email             6683 non-null
object
 27  Secretary mobile            6682 non-null
float64
 28  Asisstant Secretary name    92 non-null
object
 29  Asisstant Secretary email   92 non-null
object
 30  Asisstant Secretary mobile  92 non-null
float64
 31  Board Member name           429 non-null
object
 32  Board Member email          429 non-null
object
 33  Board Member mobile         429 non-null
float64
 34  Vice Chairman name          499 non-null
object
 35  Vice Chairman email         499 non-null
object
 36  Vice Chairman mobile        499 non-null
float64
 37  Member name                 2672 non-null
object
 38  Member email                2672 non-null
```

```
object
 39  Member mobile                2672 non-null
float64
 40  CleanName                    10000 non-null
object
 41  CleanState                   10000 non-null
object
 42  Location                     1002 non-null
object
 43  S.No.                        1002 non-null
string
 44  Registration                 1002 non-null
string
 45  AssociationName              1002 non-null
object
 46  Address                      999 non-null
object
 47  Nature                       1002 non-null
object
 48  FCRA                         2126 non-null
object
dtypes: float64(8), object(39), string(2)
memory usage: 3.8+ MB
```

Write sample csv

In [4]:

```
DarpanFcraDF.to_csv('SampleDarpan21FCRA.csv', index=False)
```

Make sets of IDs for all tags

tags are "All", "FCRA", "URL", "MA1"

In [5]:

```
TagsToIDVL = []
TagsToIDD = {}
```

All NGO IDs

In [6]:

```
AllIDV = set(DarpanFcraDF.UniqueID)
```

In [7]:

```
TagsToIDVL.append({'tag': 'All', 'IDSet': AllIDV})
TagsToIDD['All'] = list(AllIDV)
len(AllIDV)
```

Out[7]:

```
9967
```

FCRA number exists?

In [8]:

```
FCRAV = set(DarpanFcraDF.UniqueID[DarpanFcraDF.FCRA.notnull()])
```

In [9]:

```
TagsToIDVL.append({'tag': 'FCRA', 'IDSet': FCRAV})
TagsToIDD['FCRA'] = list(FCRAV)
len(FCRAV)
```

Out[9]:

```
2093
```

URL exists?

In [10]:

```
URLV = set(DarpanFcraDF.UniqueID[DarpanFcraDF['ngo url'].notnull()])
```

In [11]:

```
TagsToIDVL.append({'tag': 'URL', 'IDSet': URLV})
TagsToIDD['URL'] = list(URLV)
len(URLV)
```

Out[11]:

```
2371
```

Major activities exists?

In [12]:

```
MA1V = set(DarpanFcraDF.UniqueID[DarpanFcraDF['Major Activities1'].notnull()])
```

In [13]:

```
TagsToIDVL.append({'tag': 'MajorActivities', 'IDSet': MA1V})
TagsToIDD['MA1'] = list(MA1V)
len(MA1V)
```

Out[13]:

```
7483
```

Write csv for tags and IDs

https://www.geeksforgeeks.org/how-to-save-a-python-dictionary-to-a-csv-file/ (https://www.geeksforgeeks.org/how-to-save-a-python-dictionary-to-a-csv-file/)

In [14]:

```
field_names = ['tag', 'IDSet']
with open('SampleTagsToIDV.csv', 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames = field_name
s)
    writer.writeheader()
    writer.writerows(TagsToIDVL)
```

write json for tags and IDs

https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/ (https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/)

In [15]:

```
# TagsToIDD[tag] = [UniqueID]
with open("SampleTagsToIDList.json", "w") as outfile:
    json.dump(TagsToIDD, outfile)
```

Make reverse look up dictionaries for Issues, States and Districts

Issues: "Agriculture,Environment & Forests,Health & Family Welfare," States: "UTTAR PRADESH, testingswss, UTTAR PRADESH, testingswss, UTTAR PRADESH," Need to strip, remove '' and 'testingswss' and dedupe.

In [16]:

```
IssueToIDD = {}
StateToIDD = {}
IDToStateDistD = {}
for index, row in DarpanFcraDF.iterrows():
    UniqueID = row['UniqueID']
    Issues = row['issues working db']
    States = row['operational states db']
    Dists =row['operational district db']

    # issues dict
    try:
        IssuesL = list(set(Issues.split(',')))
        IssuesL.remove('')
        for issue in IssuesL:
            if issue in IssueToIDD:
                IssueToIDD[issue].append(UniqueID)
            else:
                IssueToIDD[issue] = [UniqueID]
    except (AttributeError, ValueError):
        pass

    # states dict
    try:
        StatesL = list(set(map(lambda s: s.strip(), States.s
plit(','))))
        StatesL.remove('')
        StatesL.remove('testingswss')
        for state in StatesL:
            if state in StateToIDD:
                StateToIDD[state].append(UniqueID)
            else:
                StateToIDD[state] = [UniqueID]
    except (AttributeError, ValueError):
        pass

    # districts list
    try:
        Dists1 = Dists.replace('->', ',')
        DistL = list(map(lambda s: s.strip(), Dists1.split(
',')))
    except (AttributeError, ValueError):
        pass

    DistL = [elem for elem in DistL if elem != '']
    DistL = [elem for elem in DistL if elem != 'testingswss'
]

    # ID to states/districts
    IDToStateDistD[UniqueID] = {}
    for state in StatesL:
        IDToStateDistD[UniqueID][state] = []

    for location in DistL:
        if location in StatesL:
            state = location
        else:
            IDToStateDistD[UniqueID][state].append(location)

    for state in StatesL:
        IDToStateDistD[UniqueID][state] = list(set(IDToState
DistD[UniqueID][state]))
```

In [17]:

```
StateDistToIDD = {}
for ID in IDToStateDistD:
    for state in IDToStateDistD[ID]:
        if state in StateDistToIDD:
            pass
        else:
            StateDistToIDD[state] = {}
        for dist in IDToStateDistD[ID][state]:
            if dist in StateDistToIDD[state]:
                StateDistToIDD[state][dist].append(ID)
            else:
                StateDistToIDD[state][dist] = [ID]
```

In [18]:

```
StatesL = list(StateDistToIDD.keys())
StatesSer = pd.Series(StatesL)

IssuesSer = pd.Series(list(IssueToIDD.keys()))
```

write json for sets of NGOs in Issue, State and State, Dist

https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/ (https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/)

In [19]:

```
# IssueToIDD[issue] = [UniqueID]
with open("SampleIssueToIDList.json", "w") as outfile:
    json.dump(IssueToIDD, outfile)

# StateToIDD[state] = [UniqueID]
with open("SampleStateToIDList.json", "w") as outfile:
    json.dump(StateToIDD, outfile)

# StateDistToIDD[state][dist] = [UniqueID]
with open("SampleStateDistToIDList.json", "w") as outfile:
    json.dump(StateDistToIDD, outfile)
```

Write csv for sets of NGOs by feature

Write csv for sets of NGOs by issue

In [20]:

```
IssueToIDVL = []
for issue in IssueToIDD:
    IssueToIDVL.append({'issue': issue, 'IDSet': IssueToIDD[issue]})

field_names = ['issue', 'IDSet']
with open('SampleIssueToIDV.csv', 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames = field_names)
    writer.writeheader()
    writer.writerows(IssueToIDVL)
```

Write csv for sets of NGOs by State

In [21]:

```
StateToIDVL = []
for state in StateToIDD:
    StateToIDVL.append({'state': state, 'IDSet': StateToIDD[state]})

field_names = ['state', 'IDSet']
with open('SampleStateToIDV.csv', 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames = field_names)
    writer.writeheader()
    writer.writerows(StateToIDVL)
```

Write csv for sets of NGOs by State and Dist

In [22]:

```
StateDistToIDVL = []
for state in StateDistToIDD:
    for dist in StateDistToIDD[state]:
        StateDistToIDVL.append({'state': state, 'dist': dist
, 'IDSet': StateDistToIDD[state][dist]})

field_names = ['state', 'dist', 'IDSet']
with open('SampleStateDistToIDV.csv', 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames = field_name
s)
    writer.writeheader()
    writer.writerows(StateDistToIDVL)
```

Filter NGOs

Select Issues

In [23]:

```
IssuesSer
```

Out[23]:

```
0                                   Any Other
1                                 Agriculture
2                     Health & Family Welfare
3                         Disaster Management
4                                    Children
5                                    HIV/AIDS
6                                      Sports
7                       Environment & Forests
8             Women's Development & Empowerment
9                        Education & Literacy
10                             Tribal Affairs
11                             Drinking Water
12                       Dairying & Fisheries
13                           Animal Husbandry
14                              Art & Culture
15                        Vocational Training
16                            Minority Issues
17                       Science & Technology
18                              Biotechnology
19                             Land Resources
20           Micro Small & Medium Enterprises
21                               Human Rights
22                            Water Resources
23           Scientific & Industrial Research
24                      Legal Awareness & Aid
25    Urban Development & Poverty Alleviation
26                           Dalit Upliftment
27                            Food Processing
28                              Youth Affairs
29    Rural Development & Poverty Alleviation
30     Information & Communication Technology
31                               Aged/Elderly
32                          Differently Abled
33                             Panchayati Raj
34                        Labour & Employment
35            Right to Information & Advocacy
36                                    Housing
37                               Civic Issues
38                                  Nutrition
39                       Micro Finance (SHGs)
```

```
40                     New & Renewable Energy
41                          Prisoner's Issues
42                                    Tourism
43                          Skill Development
dtype: object
```

In [24]:

```
selection = input("Select index of (preferably) one issue (or indices of upto 3 Issues) you are interested in, separated by ',' ind1, ind2, ind3 from above list\n").split(',')

IDInIssuesV = set()
for ind in selection:
    print("Number of NGOs in Issue", IssuesSer[int(ind)], "=", len(IssueToIDD[IssuesSer[int(ind)]]))
    IDInIssuesV = IDInIssuesV.union(set(IssueToIDD[IssuesSer[int(ind)]]))
print("Number of NGOs in any of the Issues =", len(IDInIssuesV))
```

```
Select index of (preferably) one issue (or indices of upto 3 Issues) you are interested in, separated by ',' ind1, ind2, ind3 from above list
37, 33, 36
Number of NGOs in Issue Civic Issues = 1323
Number of NGOs in Issue Panchayati Raj = 1091
Number of NGOs in Issue Housing = 798
Number of NGOs in any of the Issues = 1999
```

Select Region (States or Districts in a State)

In [26]:

```
DistrictsOrStates = str(input("To select up to 3 districts from a single state, type '1', else '0' - you will have the choice of selecting up to 3 states\n"))

if DistrictsOrStates == '1':
    print(StatesL,'\n')

    TheState = str(input("Select ONLY ONE state whose districts you are interested in\n"))

    StateDistL = list(StateDistToIDD[TheState].keys())
    print('\n', StateDistL, '\n')

    selection = str(input("Select upto 3 districts you are interested in from above list, separated by ','\n")).split(',')

    IDInRegionV = set()
    for dist in selection:
        print("number of NGOs in", dist, "=", len(StateDistToIDD[TheState][dist.strip()]))
        IDInRegionV = IDInRegionV.union(set(StateDistToIDD[TheState][dist.strip()]))
    print("number of NGOs in region = ", len(IDInRegionV))

else:
    print(StatesSer)
    selection = input("\nSelect indices of upto 3 states you are interested in, separated by ',' ind1, ind2, ind3 from above list\n").split(',')

    IDInRegionV = set()
    for ind in selection:
        print("number of NGOs in", StatesSer[int(ind)], "=", len(StateToIDD[StatesSer[int(ind)]]))
        IDInRegionV = IDInRegionV.union(set(StateToIDD[StatesSer[int(ind)]]))
    print("number of NGOs in region =", len(IDInRegionV))
```

```
To select up to 3 districts from a single state,
type '1', else '0' - you will have the choice of
selecting up to 3 states
0
0                     KARNATAKA
1                     RAJASTHAN
2                         DELHI
3                    TAMIL NADU
4                 UTTAR PRADESH
5                   WEST BENGAL
6                   MAHARASHTRA
7                MADHYA PRADESH
8                        KERALA
9                ANDHRA PRADESH
10              JAMMU & KASHMIR
11                      HARYANA
12                        BIHAR
13                 CHHATTISGARH
14                      GUJARAT
15                   PUDUCHERRY
16             HIMACHAL PRADESH
17                       ORISSA
18                       PUNJAB
19                       LADAKH
20                   CHANDIGARH
21                  UTTARAKHAND
22                      TRIPURA
23                    JHARKHAND
24                        ASSAM
25                      MANIPUR
26            ARUNACHAL PRADESH
27                    TELANGANA
28                          GOA
29                      MIZORAM
30                     NAGALAND
31    ANDAMAN & NICOBAR ISLANDS
32                    MEGHALAYA
33                       SIKKIM
34         DADRA & NAGAR HAVELI
35                  LAKSHADWEEP
36                  DAMAN & DIU
dtype: object
```

```
Select indices of upto 3 states you are interest
ed in, separated by ',' ind1, ind2, ind3 from ab
ove list
0, 1, 2
number of NGOs in KARNATAKA = 6
number of NGOs in RAJASTHAN = 14
number of NGOs in DELHI = 19
number of NGOs in region = 20
```

In [27]:

```
FinalV = IDInIssuesV.intersection(IDInRegionV)
print("Number of NGOs in Issues and region =", len(FinalV))
```

```
Number of NGOs in Issues and region = 12
```

Select tags

In [28]:

```
FCRATag = str(input("Are you a looking to make a donation to
an NGO in Foreign Currency?\n'1' for 'Yes' '0' for 'No'\n"))

FCRAReqV = AllIDV
if FCRATag == '1':
    FCRAReqV = FCRAV
```

```
Are you a looking to make a donation to an NGO i
n Foreign Currency?
'1' for 'Yes' '0' for 'No'
0
```

In [29]:

```
URLTag = str(input("Do you want to be able to explore the NGO's website?\n'1' for 'Yes' '0' for 'No'\n"))

URLReqV = AllIDV
if URLTag == '1':
    URLReqV = URLV
```

```
Do you want to be able to explore the NGO's webs
ite?
'1' for 'Yes' '0' for 'No'
1
```

In [30]:

```
MATag = str(input("Would you like to be able to see the NGO's description of Major Activities?\n'1' for 'Yes' '0' for 'No'\n"))

MAReqV = AllIDV
if MATag == '1':
    MAReqV = MA1V
```

```
Would you like to be able to see the NGO's descr
iption of Major Activities?
'1' for 'Yes' '0' for 'No'
1
```

Final set

In [31]:

```
FinalV = FinalV.intersection(FCRAReqV)
print("Number of filtered NGOs =", len(FinalV))
```

```
Number of filtered NGOs = 12
```

In [32]:

```
FinalV = FinalV.intersection(URLReqV)
print("Number of filtered NGOs =", len(FinalV))
```

```
Number of filtered NGOs = 8
```

In [33]:

```
FinalV = FinalV.intersection(MAReqV)
print("Number of filtered NGOs =", len(FinalV))
```

```
Number of filtered NGOs = 8
```

In [ ]:

```

```

Show info for sample of 5

In [34]:

```
FinalL = list(FinalV)
sampleL = random.sample(FinalL, 5)
```

In [35]:

```
DarpanFcraDF.set_index("UniqueID", inplace=True)
```

In [36]:

```
DarpanFcraDF.loc[sampleL][['Name', 'ngo url', 'Email', 'Mobile', 'Major Activities1', 'Secretary name', 'Secretary mobile', 'Secretary email']]
```

Out[36]:

	Name	ngo url	
UniqueID			
DL/2010/0026305	Sansaptak	http://www.sansaptaktheatre.com	s
TN/2016/0111485	Bright Light Society	http://blsngo.org	
UP/2011/0045804	Veena Vadini Children And Woman Educational We...	http://veenavadini.co.in/	
DL/2009/0000637	DHRUVH SOCIAL AWARENESS FORUM	http://dhruvh.org.in	
MP/2010/0030527	KIRTI BALLET AND PERFORMING ARTS	http://	

In [ ]: