In [2]:
```
from google.colab import drive
drive.mount("/content/drive")
```

```
Mounted at /content/drive
```

In [ ]:
```
!pip install tweepy
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Requirement already satisfied: tweepy in /usr/local/lib/python3.7/dist-packages (3.10.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from tweepy) (1.3.1)
Requirement already satisfied: requests[socks]>=2.11.1 in /usr/local/lib/python3.7/dist-packages (from tweepy) (2.23.0)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.7/dist-packages (from tweepy) (1.15.0)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->tweepy) (3.2.0)
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: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy) (2.10)
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: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy) (2022.5.18.1)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.7/dist-packages (from requests[socks]>=2.11.1->tweepy) (1.7.1)
```

In [3]:
```
!pip install --upgrade tweepy
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-whe
els/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Requirement already satisfied: tweepy in /usr/local/lib/python3.7/dist-packages (3.10.0)
Collecting tweepy
  Downloading tweepy-4.10.0-py3-none-any.whl (94 kB)
     |████████████████████████████████| 94 kB 3.4 MB/s
Collecting requests<3,>=2.27.0
  Downloading requests-2.28.0-py3-none-any.whl (62 kB)
     |████████████████████████████████| 62 kB 1.9 MB/s
Requirement already satisfied: oauthlib<4,>=3.2.0 in /usr/local/lib/python3.7/dist-packages (from tweepy)
 (3.2.0)
Requirement already satisfied: requests-oauthlib<2,>=1.2.0 in /usr/local/lib/python3.7/dist-packages (from
 tweepy) (1.3.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from reques
ts<3,>=2.27.0->tweepy) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests<
3,>=2.27.0->tweepy) (2022.6.15)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.
27.0->tweepy) (2.10)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.7/dist-packages (from re
quests<3,>=2.27.0->tweepy) (2.0.12)
Installing collected packages: requests, tweepy
  Attempting uninstall: requests
    Found existing installation: requests 2.23.0
    Uninstalling requests-2.23.0:
      Successfully uninstalled requests-2.23.0
  Attempting uninstall: tweepy
    Found existing installation: tweepy 3.10.0
    Uninstalling tweepy-3.10.0:
      Successfully uninstalled tweepy-3.10.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed.
 This behaviour is the source of the following dependency conflicts.
google-colab 1.0.0 requires requests~=2.23.0, but you have requests 2.28.0 which is incompatible.
datascience 0.10.6 requires folium==0.2.1, but you have folium 0.8.3 which is incompatible.
Successfully installed requests-2.28.0 tweepy-4.10.0
```

In [ ]:
```
import tweepy
import pandas as pd
import numpy as np
```

In [ ]:
```
api_key="4gHKMDwwMwv1EfMzgeXME52de"
api_key_secret="ruBa71JXiwmE6mqBXFsZTjdFcUELkZaqwS8ucilwXtWrGW6UOV"
access_token="1496803214410543107-q3kZ4XAfmpqOMk90XIvnYn31Sb8BVH"
access_token_secret="vjuZDdkn5FbODHfTpybWYmca2wTZKE4OJurPedlfHsvGO"
```

Twitter api use plan:

####1. Select top/active accounts in selected domains. ####2. selected domains and accounts: politics/news ---- Joe Biden entertainment industry---- rihanna sports industry ---- lebron james business---- Elon Musk tech ---- Meta medical---CDC

In [ ]:
```
tweepy.__version__
```

Out[30]:
```
'4.10.0'
```

Network data collection start

##

In [4]:
```
import tweepy
import pandas as pd
import numpy as np
```

In [5]:
```
api_key="4gHKMDwwMwv1EfMzgeXME52de"
api_key_secret="ruBa71JXiwmE6mqBXFsZTjdFcUELkZaqwS8ucilwXtWrGW6UOV"
access_token="1496803214410543107-q3kZ4XAfmpqOMk90XIvnYn31Sb8BVH"
access_token_secret="vjuZDdkn5FbODHfTpybWYmca2wTZKE4OJurPedlfHsvGO"
bearer_token="AAAAAAAAAAAAAAAAAAAAAD8TcwEAAAAA0fZGESjCOgkNLF%2BfCP15aMSCzIQ%3DAR5esyYVU5r6gBEgpyqSdh1U8ZEJmv
```

In [6]:
```
client=tweepy.Client(bearer_token="AAAAAAAAAAAAAAAAAAAAAD8TcwEAAAAA0fZGESjCOgkNLF%2BfCP15aMSCzIQ%3DAR5esyYVU
```

In [7]:
```
client.access_token=access_token
client.consumer_key=api_key
client.consumer_secret=api_key_secret
client.access_token_secret=access_token_secret
```

In [8]:

```
import os
dir_path = r'E:\\account\\'

# list to store files
res = []

# Iterate directory
for path in os.listdir("/content/drive/MyDrive/social net j component/twitter"):
    # check if current path is a file
    if os.path.isfile(os.path.join("/content/drive/MyDrive/social net j component/twitter", path)):
        id=path.split(".")[0]
        res.append(int(id))
print(res)
```

```
[14984096, 307458983, 307458983, 14984096, 307458983, 44666348, 301297627, 44666348, 44666348, 30129762
7, 301297627, 301297627, 44666348, 44666348, 301297627, 68179571, 68179571, 68179571, 68179571, 177573
1, 68179571, 1775731, 14191254, 4387041, 1775731, 1775731, 14191254, 14191254, 14191254, 14191254, 1775
731, 15741636, 23469247, 15741636, 23469247, 4387041, 4387041, 23469247, 4387041, 15741636, 23469247, 1
5741636, 4387041, 23469247, 15741636, 18486243, 18486243, 18486243, 16305227, 16305227, 18486243, 16305
227, 16305227, 18486243, 16305227, 115221382, 115221382, 384761809, 115221382, 384761809, 115221382, 38
4761809, 384761809, 384761809, 115221382, 82726142, 17045060, 82726142, 82726142, 82726142, 82726142, 1
99039532, 17045060, 17045060, 199039532, 17045060, 17045060, 199039532, 199039532, 199039532, 34242979,
34242979, 34242979, 34242979, 34242979, 360025190, 16437633, 360025190, 16437633, 360025190, 16437633,
360025190, 235219400, 360025190, 16437633, 16437633, 235219400, 790205, 321620987, 321620987, 32162098
7, 235219400, 321620987, 321620987, 235219400, 235219400, 40777046, 790205, 21420959, 21420959, 790205,
21420959, 790205, 790205, 21420959, 21420959, 40777046, 137046366, 137046366, 40777046, 40777046, 43003
845, 43003845, 43003845, 137046366, 43003845, 40777046, 137046366, 137046366, 43003845, 14134942, 14134
942, 14134942, 155976326, 14134942, 14134942, 21437537, 155976326, 155976326, 21437537, 21437537, 15597
6326, 21437537, 155976326, 21437537, 15347317, 250340951, 15347317, 15347317, 215328630, 250340951, 153
47317, 250340951, 250340951, 250340951, 15347317, 215328630, 215328630, 215328630, 69548480, 69548480,
69548480, 69548480, 15797184, 215328630, 69548480, 15797184, 15797184, 40813518, 15797184, 15797184, 14
1341662, 40813518, 40813518, 141341662, 120679977, 141341662, 141341662, 40813518, 40813518, 141341662,
124296976, 124296976, 120679977, 120679977, 120679977, 124296976, 124296976, 120679977, 124296976, 2517
```

In [9]:

```
res=list(set(res))
```

selecting 10 random users to whose most recent tweet was no older than 3 years

In [ ]:

```
revised_nodes=[]
for user in res:
    try:
      latest=client.get_tweet(id=client.get_users_tweets(id=user,max_results=5).data[0].id,tweet_fields=["cr
      print(latest.data)
    except:
      continue
    if latest.data.created_at.year>=2019:
       revised_nodes.append(user)
       if len(revised_nodes)==10:
         break
```

```
@Jake_PSD @CinchGaming Anyone wanna be my son?
@Beeestonia 3 words? OFF ME NUT
@Wario64 This is rough
RT @McDonaldsJapan: 「見せてもらおうか。マクドナルドの新しいてりやきの性能とやらを。」

#食レポのシャア
第2弾『 #赤いガーリックてりやき 』🔥

シャア大佐（CV池田秀一）が赤いガーリックてりやきを食レポ😆
音声ONでお楽しみください📣

#シャア専…
@robbiedaymond You can keep Cardiff. There's a dragon over in Caerphilly Castle!
RT @Ukraine: Many people around the world now google: ‘Why Russia invaded Ukraine?’

Indeed, WHY?

Here is a website to explain: https://t.… (https://t.…)
RT @TODAYshow: Tuesday on TODAY, Kim Kardashian joins us live in Studio 1A to talk about her exciting new b
usiness venture, "The Kardashian…
RT @jeypawlik: We ended up rehashing them into our webcomic Dead City, which is completed if you want to ch
eck that out:
https://t.co/VSC7… (https://t.co/VSC7…)
RT @GoodPoliticGuy: his wife literally tried to overturn the election
Can you spot the Saffron-crowned Tanager on this Bird Friendly farm? This stunning bird is a fairly common
 member of mixed flocks and is a resident of cloud forest and forest edges around 1,200–2,400 m from Venezu
ela to Bolivia. #FeatheredFactFriday https://t.co/brGp8WiJH8 (https://t.co/brGp8WiJH8)
```

In [ ]:
```
revised_nodes
```

Out[11]:
```
[402970625,
 203868162,
 14827526,
 16193542,
 24541192,
 11681802,
 26585095,
 15706128,
 2576401,
 21364753]
```

Importing Ego-Networks of the selected users

Ego Network of 402970625

In [10]:
```
ego_1=pd.read_csv("/content/drive/MyDrive/social net j component/twitter/402970625.edges",header=None,sep="
ego_1
```

Out[10]:

	0	1
0	59804598	7860742
1	6240732	18742444
2	9095652	20747847
3	156076078	15234657
4	7565302	290097288
...	...	...
731	294198566	24742040
732	7860742	11928542
733	5906322	22258315
734	47752611	22679419
735	28125177	3359851

736 rows × 2 columns

Community detection using Louvain Algorithm(Modularity algorithm)

In [11]:
```
ego_1.columns=["a","b"]
subset = ego_1[["a","b"]]
edge_tuple = [tuple(x) for x in subset.to_numpy()]
```

In [ ]:
```
import community
dir(community)
```

Out[66]:
```
['__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__']
```

In [12]:
```
!pip uninstall community
!pip install python-louvain
```

```
Found existing installation: community 1.0.0b1
Uninstalling community-1.0.0b1:
  Would remove:
    /usr/local/lib/python3.7/dist-packages/community-1.0.0b1.dist-info/*
    /usr/local/lib/python3.7/dist-packages/community/*
  Would not remove (might be manually added):
    /usr/local/lib/python3.7/dist-packages/community/community_louvain.py
    /usr/local/lib/python3.7/dist-packages/community/community_status.py
Proceed (y/n)? y
  Successfully uninstalled community-1.0.0b1
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Requirement already satisfied: python-louvain in /usr/local/lib/python3.7/dist-packages (0.16)
Requirement already satisfied: networkx in /usr/local/lib/python3.7/dist-packages (from python-louvain) (2.6.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from python-louvain) (1.21.6)
```

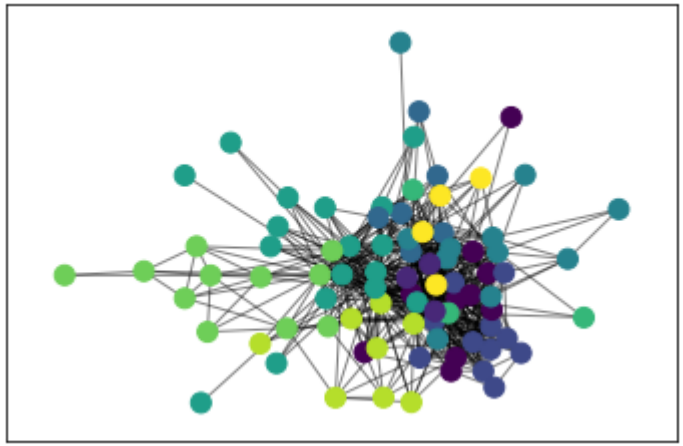
In [13]:

```
from community import community_louvain
import networkx as nx
import matplotlib.pyplot as plt
import matplotlib.cm as cm
# define the graph
edge = edge_tuple
G = nx.Graph()
G.add_edges_from(edge)
# retrun partition as a dict
partition = community_louvain.best_partition(G,resolution=1.5)
print(partition)
```

```
{59804598: 0, 7860742: 0, 6240732: 2, 18742444: 2, 9095652: 2, 20747847: 2, 156076078: 3, 15234657: 2, 7565
302: 2, 290097288: 4, 46146200: 2, 289738351: 5, 156588288: 3, 18927441: 0, 47161442: 0, 3359851: 6, 353767
514: 3, 277225744: 7, 123161418: 8, 14824849: 8, 18948541: 8, 306023079: 8, 24742040: 9, 182141321: 4, 1705
12878: 5, 47752611: 5, 287625695: 3, 286951179: 5, 15913: 5, 28125177: 4, 82030021: 9, 253681548: 5, 263533
393: 5, 5906322: 2, 253731559: 5, 119427247: 3, 15853668: 1, 14716974: 2, 14903310: 0, 88342706: 4, 2475219
52: 4, 16303106: 8, 90420314: 8, 330113543: 4, 185449401: 7, 7517222: 7, 36198161: 0, 46423291: 4, 5039396
0: 0, 22679419: 1, 24529777: 7, 259379883: 7, 18335521: 8, 14589257: 0, 15256613: 5, 26281970: 2, 50962538
4: 5, 52551600: 7, 64873902: 1, 428837461: 5, 369929189: 9, 263223490: 6, 294198566: 4, 299332941: 8, 40819
7359: 5, 128376989: 4, 265577358: 5, 11928542: 0, 451466872: 5, 29514951: 9, 177345928: 7, 22258315: 4, 168
767461: 7, 351228223: 3, 139848744: 7, 43170475: 2, 50570324: 0, 250831586: 7, 213560141: 4, 14568091: 5, 1
72542995: 7, 26604904: 6, 123162953: 5, 423973919: 5}
```

In [14]:

```
# visualization
pos = nx.fruchterman_reingold_layout(G)
cmap = cm.get_cmap('viridis', max(partition.values()) + 1)
nx.draw_networkx_nodes(G, pos, partition.keys(), node_size=100,cmap=cmap, node_color=list(partition.values()))
nx.draw_networkx_edges(G, pos, alpha=0.5)
plt.show()
```



Heuristics for tweet collection:

1. get 10 tweets per person in every community ✅
2. put them all into a data frame ✅
3. clean and preprocess those tweets ✅
4. now concatenate all of them tweets ⏳
5. now perform BART text summarization on them to generate a summary for every community ⏳

NEW MODIFICATIONS* ⏳ to be performed between step 3 and step 4

1. execute Latent dirichlet allocation on the cleaned tweets to assign topic label ✅
2. get the topic label that is in majority now take the most frequent word used in the tweets belonging to that topic and assign that word as the topic and also as the community name

---

CORRECTION OF POINT 2: RATHER THAN TRYING GET THE MAJORITY TOPIC FOR A COMMUNITY. USE THE LIST RETURNED BY THE LDA MODEL AND REMOVE STOP WORDS FROM THAT LIST AND USE THE REMAINING WORDS TO PERFORM THE SEARCH AGAIN ✅

3. now get the list of top 5 or top 10 frequent words in the topic based on frequency
4. now again perform a search for 20 tweets per person containing these the top words

In [15]:
```
communities=list(set(partition.values()))
```

In [ ]:
```
tweets_in_com
```

Out[18]:
```
{0: ['@capybarafan14 Hello,\nPlease follow our Official Support page here on Twitter, so that we can re
ach out to you via DM and assist you appropriately. https://t.co/3ZhVDWKaYU', (https://t.co/3ZhVDWKaY
U',)
  '@Angelo89711626 Hello,\nPlease follow our Official Support page here on Twitter, so that we can reac
h out to you via DM and assist you appropriately. https://t.co/3ZhVDWKaYU', (https://t.co/3ZhVDWKaYU',)
  "RT @UbisoftSupport: We're aware of the issues affecting connectivity in Rainbow Six Siege and are wo
rking towards resolving this ASAP: http…",
  '@Abhis26575489 Hello,\n\nPlease follow our Official page on Twitter in order to get a Direct Message
regarding the issue you are encountering.\n\nThank you!',
  '@VanWinkle2112 Please follow our official page on Twitter in order to be able to get a direct messag
e regarding the issue you are encountering.',
  '@frank_john98 Please follow our Official Xbox Support page on Twitter in order to be able to get a D
irect Message regarding the issue you are encountering https://t.co/3ZhVDWKaYU', (https://t.co/3ZhVDWKa
YU',)
  '@JoelPlant1 Hello\nThank you for reaching Microsoft Xbox Support on Twitter.\nPlease follow our Offi
cial Xbox Support page on Twitter in order to be able to get a Direct Message regarding the issue you a
re encountering.\nHave a nice day! https://t.co/3ZhVDWKaYU', (https://t.co/3ZhVDWKaYU',)
  '@JulianStar18 Hello please  follow our Xbox page on Twitter in  order to Get a Direct Message regard
ing your issue . https://t.co/3ZhVDWKaYU', (https://t.co/3ZhVDWKaYU',)
```

In [ ]:
```
nodes_in_com=dict()
for i in communities:
    nodes=[k for k,v in partition.items() if v == i]
    nodes_in_com[i]=nodes

# now we get 10 tweets for every node from the community list
tweets_in_com=dict()
for k,v in nodes_in_com.items():
    tweet_list=[]
    for n in v:
      response=client.get_users_tweets(id=n,max_results=20)
      if response.data!=None:
        for t in response.data:
            tweet_list.append(t.text)
    tweets_in_com[k]=tweet_list
```

In [ ]:
```
revised_nodes
```

Out[21]:
```
[402970625,
 203868162,
 14827526,
 16193542,
 24541192,
 11681802,
 26585095,
 15706128,
 2576401,
 21364753]
```

In [ ]:
```
try:
    geeky_file = open('402970625.txt', 'wt')
    geeky_file.write(str(tweets_in_com))
    geeky_file.close()

except:
    print("Unable to write to file")
```

```
Unable to write to file
```

In [ ]:
```
import json
import ast
with open("/content/402970625.txt") as f:
  data=f.read()
tweets_in_com=ast.literal_eval(data)
#tweets_in_com= json.loads(json.dumps(data))
```

In [ ]:
```
tweets_in_com=dict(tweets_in_com)
```

In [ ]:
```
tweets_
```

Out[20]: dict

In [ ]:
```
print(tweets_in_com.items())
```


```
dict_items([(0, ["Calling all #TaskForce141 recruits 🚨\n\nDon't be the F.N.G. Tap below and add your A
ctivision ID to get your customized #ModernWarfare2 recruitment patch.", 'Kim Tae Young is not messing
around in her new Operator bundle 👹😤 https://t.co/WRZku39kFl', (https://t.co/WRZku39kFl',) 'Where ar
e you dropping first? \U0001fa82', "Thank you to our community for helping reveal Fortune's Keep, a new
#Warzone Resurgence map, where the amount of danger and action is second only to its riches 💰🏆💥\n\n
Coming soon 🌴 https://t.co/MzVuKATvF7", (https://t.co/MzVuKATvF7",) "RT @eColiEspresso: So... @CallofD
uty just sent me the final piece of the new Fortune's Keep tac map in #Warzone 😬👀 https://t.co/zPLz8
W9qJs", (https://t.co/zPLz8W9qJs",) 'RT @timthetatman: 👀 @CallofDuty https://t.co/aWVPxlXMxu', (http
s://t.co/aWVPxlXMxu',) 'RT @RaidAway: Call of Duty just sent me this 👀 https://t.co/zqJQcRTpTw', (http
s://t.co/zqJQcRTpTw',) 'RT @Dysmo: . @CallofDuty just sent me another piece of the new #FortunesKeep  m
ap 👀 https://t.co/BPvNagCJVD', (https://t.co/BPvNagCJVD',) '@timthetatman Corner pieces are still piec
es #FortunesKeep', 'What is the golden rule in #Warzone?\n\nWe will DM a piece of the map to whoever ge
t's it right #FortunesKeep', 'RT @BeenoxCODPC: Found one! #FortunesKeep 🤔 https://t.co/4Ba6S5bzPl', (h
ttps://t.co/4Ba6S5bzPl',) '@BeenoxCODPC \U0001fae1', '@olirufiange 👀', '#FortunesKeep https://t.co/cHJ
blA8imd (https://t.co/cHJblA8imd) https://t.co/mCbo27XXSs', (https://t.co/mCbo27XXSs',) '@GuardianZombi
e1 Respect the attempt 👏', "RT @LuckyChamu: Here's a HD Look at what was on the cake!\n\nI CAN SEE THE
1v4 CLIPS ALREADY \n\n #WarzoneResurgence | #FortunesKeep https://t.…", (https://t.…",) 'RT @RavenSoftw
are: Fortune favors the bold on #FortunesKeep! 💰\n\nThis highly classified (and likely stolen) intel i
s just one piece of a lar…', '@RavenSoftware Who will be the first to complete the puzzle? 👀', "RT @Be
```


Text cleaning and preprocessing

In [22]:
```
import re
import nltk
```

In [23]:
```
nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Unzipping corpora/omw-1.4.zip.
```

Out[23]:
```
True
```

In [29]:
```
from nltk.stem import WordNetLemmatizer
from keras.preprocessing.text import Tokenizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from nltk.corpus import stopwords
```

In [26]:
```
lemmatizer = WordNetLemmatizer()
stop_words= stopwords.words('english')
```

In [30]:

```
def preprocess1(tweets):

  tweets.encode('ascii', 'ignore').decode('ascii') #remove emojis
  tweets=tweets.lower()                    #convert to lower case
  tweets = re.sub(r"http\S+", "", tweets) #remove urls
  l=tweets.split(" ")
  for n,i in enumerate(l):
    if '@' in i:
      l[n]=""

  tweets=" ".join(l)
  tweets= re.sub('[^A-Za-z]+',' ',tweets) #remove bad character
  tweets = [word for word in tweets.split(" ") if not word in stop_words] # removing stop words
  tweets= [lemmatizer.lemmatize(token, "v") for token in tweets] #Lemmatization
  tweets=" ".join(tweets)
  return tweets
def decontract(text):
    text = re.sub(r"won\'t", "will not", text)
    text = re.sub(r"can\'t", "can not", text)
    text = re.sub(r"n\'t", " not", text)
    text = re.sub(r"\'re", " are", text)
    text = re.sub(r"\'s", " is", text)
    text = re.sub(r"\'d", " would", text)
    text = re.sub(r"\'ll", " will", text)
    text = re.sub(r"\'t", " not", text)
    text = re.sub(r"\'ve", " have", text)
    text = re.sub(r"\'m", " am", text)
    return text
```

In [31]:

```
def preprocess2(tweets):

  tweets.encode('ascii', 'ignore').decode('ascii') #remove emojis
  tweets=tweets.lower()                    #convert to lower case
  tweets = re.sub(r"http\S+", "", tweets) #remove urls
  l=tweets.split(" ")
  for n,i in enumerate(l):
    if '@' in i:
      l[n]=""

  tweets=" ".join(l)
  tweets= re.sub('[^A-Za-z]+',' ',tweets) #remove bad character
  #tweets = [word for word in tweets.split(" ") if not word in stop_words] # removing stop words
  #tweets= [lemmatizer.lemmatize(token, "v") for token in tweets] #Lemmatization
  tweets=" ".join(tweets)
  return tweets
```

In [ ]:

```
tweets_in_com
```

Out[28]:

```
{0: ["Calling all #TaskForce141 recruits 🚨\n\nDon't be the F.N.G. Tap below and add your Activision ID
to get your customized #ModernWarfare2 recruitment patch.",
  'Kim Tae Young is not messing around in her new Operator bundle 👹😤 https://t.co/WRZku39kFl', (http
s://t.co/WRZku39kFl',)
  'Where are you dropping first? \U0001fa82',
  "Thank you to our community for helping reveal Fortune's Keep, a new #Warzone Resurgence map, where t
he amount of danger and action is second only to its riches 💰🏆💥\n\nComing soon 🌴 https://t.co/MzVu
KATvF7", (https://t.co/MzVuKATvF7",)
  "RT @eColiEspresso: So... @CallofDuty just sent me the final piece of the new Fortune's Keep tac map
 in #Warzone 😬👀 https://t.co/zPLz8W9qJs", (https://t.co/zPLz8W9qJs",)
  'RT @timthetatman: 👀 @CallofDuty https://t.co/aWVPxlXMxu', (https://t.co/aWVPxlXMxu',)
  'RT @RaidAway: Call of Duty just sent me this 👀 https://t.co/zqJQcRTpTw', (https://t.co/zqJQcRTpT
w',)
  'RT @Dysmo: . @CallofDuty just sent me another piece of the new #FortunesKeep  map 👀 https://t.co/BP
vNagCJVD', (https://t.co/BPvNagCJVD',)
  '@timthetatman Corner pieces are still pieces #FortunesKeep',
  'What is the golden rule in #Warzone?\n\nWe will DM a piece of the map to whoever get’s it right #For
tunesKeep',
  'RT @BeenoxCODPC: Found one! #FortunesKeep 🤔 https://t.co/4Ba6S5bzPl', (https://t.co/4Ba6S5bzPl',)
```

In [ ]:
```
#uncleaned_summary_input=tweets_in_com
#temp1=uncleaned_summary_input
for k in tweets_in_com.keys():
    tweets_in_com[k]=list(pd.Series(tweets_in_com[k]).apply(lambda x:decontract(x)))
    tweets_in_com[k]=list(pd.Series(tweets_in_com[k]).apply(lambda x:preprocess1(x)))
```

In [ ]:
```
#uncleaned_summary_input=tweets_in_com
#temp2=uncleaned_summary_input
for k in uncleaned_summary_input.keys():
    temp2[k]=list(pd.Series(temp2[k]).apply(lambda x:decontract(x)))
    temp2[k]=list(pd.Series(temp2[k]).apply(lambda x:preprocess2(x)))
```

In [ ]:
```
tweets_in_com
```

Out[30]:
```
{0: ['call taskforce recruit f n g tap add activision id get customize modernwarfare recruitment patch
',
  'kim tae young mess around new operator bundle ',
  'drop first ',
  'thank community help reveal fortune keep new warzone resurgence map amount danger action second rich
es come soon ',
  'rt send final piece new fortune keep tac map warzone ',
  'rt ',
  'rt call duty send ',
  'rt send another piece new fortuneskeep map ',
  ' corner piece still piece fortuneskeep',
  'golden rule warzone dm piece map whoever get right fortuneskeep',
  'rt find one fortuneskeep ',
  ' ',
  ' ',
  ' fortuneskeep ',
  ' respect attempt ',
  'rt hd look cake see v clip already warzoneresurgence fortuneskeep ',
  'rt fortune favor bold fortuneskeep highly classify likely steal intel one piece lar ',
```

In [ ]:
```
for k,v in tweets_in_com:
    for t in range(0,len(v)):
        v[t]=v[t].strip()
```

LDA for topic modeling

In [ ]:
```
import string
import spacy
import gensim
from gensim import corpora
```

In [ ]:
```
tokenized_output=dict()
for k in tweets_in_com.keys():
  new_value=[]
  for v in tweets_in_com[k]:
    tok=v.split(" ")
    new_value.append(tok)
  tokenized_output[k]=new_value
```

Building separate model for every community

In [ ]:
```
for v in tokenized_output.values():
    for l in v:
        if " " in l:
          l.remove(" ")
        if "rt" in l:
          l.remove("rt")
        if "" in l:
          l.remove("")
```

In [ ]:
```
tokenized_output
```

Out[55]:
```
{0: [['call',
   'taskforce',
   'recruit',
   'f',
   'n',
   'g',
   'tap',
   'add',
   'activision',
   'id',
   'get',
   'customize',
   'modernwarfare',
   'recruitment',
   'patch'],
  ['kim', 'tae', 'young', 'mess', 'around', 'new', 'operator', 'bundle'],
  ['drop', 'first'],
  ['thank',
   'community',
```

In [ ]:
```
lda_models=[]
doc_topics=[]
for v in tokenized_output.values():
  dictionary = corpora.Dictionary(v)
  doc_term_matrix = [dictionary.doc2bow(tweet) for tweet in v]
  #doc_term_matrices.append(doc_term_matrix)
  # Creating the object for LDA model using gensim library
  LDA = gensim.models.ldamodel.LdaModel
  # Build LDA model
  lda_model = LDA(corpus=doc_term_matrix, id2word=dictionary, num_topics=7, random_state=100,
                  chunksize=1000, passes=50,iterations=100)
  doc_topics.append(lda_model.get_document_topics(doc_term_matrix))
  lda_models.append(lda_model)
```

In [ ]:
```
lda_models[4].print_topics()
```

Out[30]:
```
[(0,
  '0.017*"new" + 0.013*"world" + 0.009*"want" + 0.009*"could" + 0.009*"power" + 0.009*"character" + 0.009
*"follow" + 0.005*"heroes" + 0.005*"ever" + 0.005*"around"'),
 (1,
  '0.036*"" + 0.036*"follow" + 0.034*"twitter" + 0.034*"page" + 0.034*"official" + 0.034*"please" + 0.031
*"order" + 0.031*"direct" + 0.031*"message" + 0.031*"support"'),
 (2,
  '0.014*"take" + 0.014*"life" + 0.010*"tons" + 0.010*"let" + 0.010*"show" + 0.010*"read" + 0.010*"pt" + 0.
010*"pm" + 0.010*"makeover" + 0.010*"carbon"'),
 (3,
  '0.015*"" + 0.010*"make" + 0.010*"easter" + 0.008*"game" + 0.008*"play" + 0.008*"get" + 0.008*"would" +
0.008*"action" + 0.008*"know" + 0.008*"really"'),
 (4,
  '0.015*"always" + 0.012*"make" + 0.012*"gt" + 0.012*"" + 0.009*"one" + 0.006*"game" + 0.006*"ignsummerofg
aming" + 0.006*"story" + 0.006*"us" + 0.006*"time"'),
 (5,
  '0.018*"pride" + 0.014*"time" + 0.011*"game" + 0.011*"rainbow" + 0.011*"six" + 0.011*"amp" + 0.011*"best"
+ 0.007*"miss" + 0.007*"live" + 0.007*"siege"'),
 (6,
  '0.041*"" + 0.017*"thank" + 0.010*"amp" + 0.010*"enjoy" + 0.008*"years" + 0.008*"show" + 0.008*"hat" + 0.
005*"get" + 0.005*"season" + 0.005*"f"')]
```

In [ ]:
```
com_word_imp=dict()
for m in range(0,len(lda_models)):
    for t in lda_models[m].print_topics():
        s=t[1].split("\"")
        del s[0::2]
        if "" in s:
           s.remove("")
        for x in stop_words:
          if x in s:
             s.remove(x)
        com_word_imp[m]=s
```

In [ ]:
```
com_word_imp
```

Out[63]:
```
{0: ['platinum',
  'bronze',
  'battle',
  'royale',
  'sunday',
  'today',
  'amp',
  'pride',
  'night'],
 1: ['get', 'go', 'mizandmrs', 'take', 'amp', 'today', 'team', 'ever', 'one'],
 2: ['love',
  'game',
  'home',
  'much',
  'back',
  'new',
  'kind',
  'internet',
  'big',
  'also'],
 3: ['make',
  'trailer',
  'xbox',
  'see',
  'five',
  'universe',
  'rise',
  'maker',
  'go'],
 4: ['thank', 'amp', 'enjoy', 'years', 'show', 'hat', 'get', 'season', 'f'],
 5: ['live', 'go', 'life', 'day', 'playstation', 'see', 'know', 'pax', 'work'],
 6: ['call',
  'duty',
  'game',
  'see',
  'experimental',
  'day',
  'card',
  'creator',
  'modernwarfare',
```

```
 'cup'],
7: ['see',
 'get',
 'please',
 'would',
 'come',
 'game',
 'better',
 'today',
 'new'],
8: ['game', 'play', 'free', 'hunt', 'prop', 'trailer', 'link', 'new', 'like']}
```

new refined tweet search

In [16]:
```
nodes_in_com=dict()
for i in communities:
    nodes=[k for k,v in partition.items() if v == i]
    nodes_in_com[i]=nodes
```

In [ ]:
```
client.get_user_tweets()
```

```
Object `client.get_user_tweets` not found.
```

In [17]:
```
def tweets_of_nodes(nodes_in_com,client):
  tweets_in_com=dict()
  for k,v in nodes_in_com.items():
      tweet_list=[]
      for n in v:
        response=client.get_users_tweets(id=n,max_results=40)
        if response.data!=None:
          for t in response.data:
              tweet_list.append(t.text)
      tweets_in_com[k]=tweet_list
  return tweets_in_com
```

In [18]:
```
new_tweets=tweets_of_nodes(nodes_in_com,client)
```

In [ ]:
```
new_tweets
```

Out[75]:
```
{0: ['Play thousands of games across four generations of Xbox. Now that’s all in the family. Here are s
ome things to keep in mind: https://t.co/2JodzPNdy7 (https://t.co/2JodzPNdy7) https://t.co/4tkG0QfIHK',
(https://t.co/4tkG0QfIHK',)
  '@jacobstaffordxo Hello \nPlease follow our Official Xbox Support page on Twitter in order to be able
to get a Direct Message regarding the issue you are encountering https://t.co/3ZhVDWKaYU', (https://t.c
o/3ZhVDWKaYU',)
  '@thjR6_ Hello,\nPlease follow our Official Xbox Support page on Twitter in order to be able to get a
Direct Message regarding the issue you are encountering.\nThank you!',
  '@deezeone2 Hello , please follow our official support page on Twitter in order to be able to receive
a direct message regarding your issue . https://t.co/3ZhVDWszAk', (https://t.co/3ZhVDWszAk',)
  '@WilliamBarefi20 Please follow our official Xbox Support Twitter page to get a direct message regard
ing your questions and requests.',
  '@OncleVador Please follow our official twitter page in order to receive a direct message concerning
 your request',
  '@ZackWalkden Hello\nPlease follow our Official Xbox Support page on Twitter in order to be able to g
et a Direct Message regarding the issue you are encountering. https://t.co/3ZhVDWszAk', (https://t.co/3
ZhVDWszAk',)
  '@JoeDeezers Hello\nPlease follow our Official Xbox Support page on Twitter in order to be able to ge
t a Direct Message regarding the issue you are encountering. https://t.co/3ZhVDWszAk', (https://t.co/3Z
```

In [32]:
```
for k in new_tweets.keys():
    new_tweets[k]=list(pd.Series(new_tweets[k]).apply(lambda x:decontract(x)))
    new_tweets[k]=list(pd.Series(new_tweets[k]).apply(lambda x:preprocess1(x)))
```

In [33]:
```
try:
    geeky_file = open('new_402970625.txt', 'wt')
    geeky_file.write(str(new_tweets))
    geeky_file.close()

except:
    print("Unable to write to file")
```

Filtering out relevant tweets based on select keywords

In [ ]:
```

```

Community-wise Tweet concatenation

In [ ]:
```
a=["1","2"]
p=".".join(a)
p
```

Out[44]:
```
'1.2'
```

In [34]:
```
for k in new_tweets.keys():
    new_tweets[k]=".".join(new_tweets[k])
new_tweets
```

Out[34]:
```
{0: ' hello please follow official page twitter order get direct message regard issue encounter thank .
hello please follow official xbox support page twitter order able get direct message regard issue encou
nter . hello please follow xbox page twitter order get direct message regard issue . hello please follo
w xbox page twitter order toget direct message regard issue .play thousands game across four generation
s xbox family things keep mind . hello please follow official xbox support page twitter order able get
direct message regard issue encounter . hello please follow official xbox support page twitter order ab
le get direct message regard issue encounter thank . hello please follow official support page twitter
order able receive direct message regard issue . please follow official xbox support twitter page get d
irect message regard question request . please follow official twitter page order receive direct messag
e concern request. hello please follow official xbox support page twitter order able get direct message
regard issue encounter . hello please follow official xbox support page twitter order able get direct m
essage regard issue encounter . hello please make sure follow xbox support page twitter way receive dir
ect message query . hello please follow official xbox support page twitter order able get direct messag
e regard issue encounter . hello please follow official support page twitter reach via dm assist approp
riately . hello please make sure follow twitter page receive assistance via direct message . please fol
low official xbox support twitter page get direct message regard question request . hello please follow
official xbox support page twitter order able get direct message regard issue encounter . hello please
follow official microsoft twitter support page twitter order able get direct message regard . hello ple
ase follow official support pageon twitter order ble receivea direct message regard issue . follow offi
```

Text Summarization using BART

BART

In [35]:
```
!git clone https://github.com/huggingface/transformers \
&& cd transformers \
```

```
Cloning into 'transformers'...
remote: Enumerating objects: 98588, done.
remote: Counting objects: 100% (404/404), done.
remote: Compressing objects: 100% (219/219), done.
remote: Total 98588 (delta 226), reused 289 (delta 160), pack-reused 98184
Receiving objects: 100% (98588/98588), 92.16 MiB | 24.81 MiB/s, done.
Resolving deltas: 100% (72519/72519), done.
```

In [36]:
```
!pip install -q ./transformers
```

```
  DEPRECATION: A future pip version will change local packages to be built in-place without first copying t
o a temporary directory. We recommend you use --use-feature=in-tree-build to test your packages with this n
ew behavior before it becomes the default.
   pip 21.3 will remove support for this functionality. You can find discussion regarding this at https://g
ithub.com/pypa/pip/issues/7555. (https://github.com/pypa/pip/issues/7555.)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
     |████████████████████████████████| 596 kB 12.9 MB/s
     |████████████████████████████████| 86 kB 6.6 MB/s
     |████████████████████████████████| 6.6 MB 60.8 MB/s
  Building wheel for transformers (PEP 517) ... done
```

In [37]:
```
import torch
import transformers
from transformers import BartTokenizer, BartForConditionalGeneration
from transformers import BartConfig
```

In [38]:
```
torch_device = 'cpu'
```

In [39]:
```
tokenizer = BartTokenizer.from_pretrained('facebook/bart-large-cnn')
model = BartForConditionalGeneration.from_pretrained('facebook/bart-large-cnn')
```

```
Downloading:   0%|          | 0.00/878k [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/446k [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/1.55k [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/1.51G [00:00<?, ?B/s]
```

In [40]:
```
import google.colab.output

def bart_summarize(text, num_beams, length_penalty, max_length, min_length, no_repeat_ngram_size):

  text = text.replace('\n','')
  text_input_ids = tokenizer.batch_encode_plus([text], return_tensors='pt', max_length=1024)['input_ids'].to
  summary_ids = model.generate(text_input_ids, num_beams=int(num_beams), length_penalty=float(length_penalty
  summary_txt = tokenizer.decode(summary_ids.squeeze(), skip_special_tokens=True)
  return summary_txt

#register callback for Javascript
google.colab.output.register_callback('bart_summarize',bart_summarize )
```

TEST RUN

In [ ]:

```
from IPython.display import HTML

#spinner from https://codepen.io/vovchisko/pen/vROoYQ
spinner_css = """
<style>
@keyframes c-inline-spinner-kf {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}

.c-inline-spinner,
.c-inline-spinner:before {
  display: inline-block;
  width: 11px;
  height: 11px;
  transform-origin: 50%;
  border: 2px solid transparent;
  border-color: #74a8d0 #74a8d0 transparent transparent;
  border-radius: 50%;
  content: "";
  animation: linear c-inline-spinner-kf 300ms infinite;
  position: relative;
  vertical-align: inherit;
  line-height: inherit;
}
.c-inline-spinner {
  top: 3px;
  margin: 0 3px;
}
.c-inline-spinner:before {
  border-color: #74a8d0 #74a8d0 transparent transparent;
  position: absolute;
  left: -2px;
  top: -2px;
  border-style: solid;
}
</style>
"""
```

```
input_form = """
<link rel="stylesheet" href="https://unpkg.com/purecss@1.0.1/build/pure-min.css" integrity="sha384-oAOxQR6Dk

<div style="background-color:white; border:solid #ccc; width:800px; padding:20px; color: black;">
<p><strong>BART</strong> Seq2Seq model with SoTA summarization performance</p>
<textarea id="main_textarea" cols="75" rows="20" placeholder="Paste your text here..." style="font-family: '
<div class="pure-form pure-form-aligned">
   <div class="pure-control-group">
     <label for="no_repeat_ngram_size"><strong>no_repeat_ngram_size:</strong></label>
     <input type="number" id="no_repeat_ngram_size" value="3" style="background-color: white;">
    </div>
    <div class="pure-control-group">
      <label for="num_beams"><strong>num_beams:</strong></label>
      <input type="number" min="0" max="10" step="1" id="num_beams" value="4" style="background-color: white
    </div>
    <div class="pure-control-group">
        <label for="length_penalty"><strong>length_penalty:</strong></label>
        <input type="number" min="0.0" max="10.0" step="0.1" id="length_penalty" value="2.0" style="backgrou
    </div>
    <div class="pure-control-group">
        <label for="max_length"><strong>max_length:</strong></label>
        <input type="number" id="max_length" value="142" style="background-color: white;">
    </div>
     <div class="pure-control-group">
        <label for="min_length"><strong>min_length:</strong></label>
        <input type="number" id="min_length" value="56" style="background-color: white;">
    </div>
    <p><a target="_blank" href='https://pastebin.com/raw/BMPcUS6v'>Try to summarize this example article</a>
    <div style="width: 300px; display: block; margin-left: auto !important; margin-right: auto !important;">
        <p><button class="pure-button pure-button-primary" style="font-size: 125%%;" onclick="summarize()">S
        <span class="c-inline-spinner" style="visibility: hidden;" id="spinner"></span></p>
    </div>
</div>
</div>
"""

javascript = """
<script type="text/Javascript">


       function saveTextAsFile(textToWrite, fileNameToSaveAs)
    {
```

```
    var textFileAsBlob = new Blob([textToWrite], {type:'text/plain'});
    var downloadLink = document.createElement("a");
    downloadLink.download = fileNameToSaveAs;
    downloadLink.innerHTML = "Download File";
    if (window.webkitURL != null)
    {
        // Chrome allows the link to be clicked
        // without actually adding it to the DOM.
        downloadLink.href = window.webkitURL.createObjectURL(textFileAsBlob);
    }
    else
    {
        // Firefox requires the link to be added to the DOM
        // before it can be clicked.
        downloadLink.href = window.URL.createObjectURL(textFileAsBlob);
        downloadLink.onclick = destroyClickedElement;
        downloadLink.style.display = "none";
        document.body.appendChild(downloadLink);
    }

    downloadLink.click();
}


function summarize(){

    var text = document.getElementById('main_textarea').value;
    var no_repeat_ngram_size = document.getElementById('no_repeat_ngram_size').value;
    var num_beams = document.getElementById('num_beams').value;
    var length_penalty = document.getElementById('length_penalty').value;
    var max_length = document.getElementById('max_length').value;
    var min_length = document.getElementById('min_length').value;

    var kernel = google.colab.kernel;

    var resultPromise = kernel.invokeFunction("bart_summarize", [text,num_beams,length_penalty,max_lengt
    resultPromise.then(
        function(result) {
          document.getElementById('main_textarea').value = 'da resultado';
          document.getElementById('main_textarea').value = result.data["text/plain"];
          document.getElementById('spinner').style = "visibility: hidden;";
          saveTextAsFile(result.data["text/plain"], 'summary.txt')
    }).catch(function(error){document.getElementById('main_textarea').value = error;});
```

```
        document.getElementById('spinner').style = "visibility: visible;";
    };
</script>
"""


HTML(spinner_css + input_form + javascript)
```

Out[14]:

BART Seq2Seq model with SoTA summarization performance

Paste your text here...

no_repeat_ngram_size: 3

num_beams: 4

length_penalty: 2.0

max_length: 142

min_length: 56

Try to summarize this example article (https://pastebin.com/raw/BMPcUS6v)

Summarize

```
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncati
on=True` to explicitly truncate examples to max length. Defaulting to 'longest_first' truncation strategy.
If you encode pairs of sequences (GLUE-style) with the tokenizer you can select this strategy more precisel
y by providing a specific strategy to `truncation`.
```

In [42]:
```
com_summary=dict()
for k in new_tweets.keys():
    com_summary[k]=bart_summarize(new_tweets[k],4,2,142,56,3)
com_summary
```

```
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncati
on=True` to explicitly truncate examples to max length. Defaulting to 'longest_first' truncation strategy.
If you encode pairs of sequences (GLUE-style) with the tokenizer you can select this strategy more precisel
y by providing a specific strategy to `truncation`.
```

Out[42]:
```
{0: ' play thousands game across four generations xbox family things keep mind. Watch rainbow six pro leagu
e pc finals happen right.rt team tie head third map semi final match house dust line reveal follow. tbt tim
e go old n game look find og rainbow six best.waaah new gemstone skin pretty kinda want.rt demo ghost recon
wildlands hopefully give tip trick.hype real e.yay glad get enjoy rainbow six hat.rainbow six siege time co
me join viewer game. congratulations hat sir shall send dm regard ship.',
 1: 'Luckybastards.wine language romance speak fluently impress date host well uncork.take stroll good beer
aisle rite passage like bar mitzvah mazel tov brewdogs. parksandrec.gladiators gridiron come size miss retu
rn fridaynighttykes january. pierce brosnan bond marathon day.say nothing party like rental spend day pierc
e  bond marathon.year quit soon legal slap smokers public enjoy pierce\xa0bond\xa0marathon.',
 2: 'hat pet monsters sell well nothing.ready fight conversation street fighter game director takayuki naka
yama.soul hackers unshakably unashamedly unmistakably shin megami tensei.sonic frontiers.callisto protocol
horror game new generation hand preview.hand lightyear frontier tranquil adventure unveil amid chaos e seas
on.',
 3: 'today emotional rollercoaster thank god bikram yoga.never procrastinate late work shananagins hrs enti
re week project blame pax. yay home town sad see time around next time though.great see old friends weekend
see everyone south hopefully. travel days crazy hair makeup shit give hate layovers. get things always frea
k pheonix byepax. yup.rt pax prime complete. know around lately. hey long time talk.thank recommendation yu
mmy. take one near convention center lobby look rly familiar couch idk.',
 4: 'need iron rule gold. w.rt love hype new warzone resurgence map fortune keep among discussions see one
q... real recognize real. call duty.squad play objective..one need hand deck iconic battleship back uss tex
as make vanguard debut june read codblog intel.resurgence payload blood money make fat stack ws weekly warz
one playlist.mayhemisphere rocket hill blitz escalate quickly weekly vanguard playlist.rt introduce classic
round base gameplay vanguard season four shinonuma also feature round base specific gameplay..biggest deter
rent cheat ban teamricochet issue daily several reason often large wave since last update issue another k b
an across warzone vanguard combine',
 5: 'guy kill end ofstory.boom spend money either place hahahahaha.watch guy n play destinythegame hater kn
ow. gear.gearrrrssssss..rt package order..begin. thesame corporate crap man pay one biggest help kid school
anyone really show true greed. yea saw someone announce giveaway check supposedly walmart n circle k local
indian store.',
 6: 'Hiring gamejobs animator. always excite new cod always part dna especially mw character also people ha
nd craft game highly trust.. go need sequential number.. come back meet. look intense incredible congrats t
eam. let totem pole call top.engage community core philosophy midnight society begin new york june nd come
```

```
hang partner open everyone rsvp. challenge accept. oh yeah standards expect f p game battle pass character
skin weapon skin bunch common mobile game like day boost bundle resources craft enhance rifts play look gre
at though.diablo immortal great download see play ipad stop ipad die one quest one rift one dungeon much se
amlessly jump.top gun incredible.play part',
 7: 'month go slow want steelers football.preseason wait see steelers. fall back religion hateful comment.
guy say support gay marry fine talk people bash gays fall back.anybody go chick fil support hate say religi
on unfollow right go fuck. baby give everyone least lol. like pez. happen. would hilarious lol halo today.
lol funny hell.rt yes yes yes. yes yesYes. really tire today.',
 8: "Murder build is a new series on the Velocity Network. The show follows the lives of a group of actors
and writers as they try to create a world they can't wait to explore. The first episode of the new season p
remieres on Tuesday, March 6 at 9pm on Velocity.",
 9: 'Free tomorrow play. overqualified available.prop hunt.let play..learn make music live gonna love guara
ntee. rank apex pro gamer live best life.. close gimmie point.road diamond continue today.diamonds.hello lo
ok. one heck journey thank partner. ad thank sponsor video one winter storm financial literacy pure joy pro
p hunt victories learn getyourmoneyright visit. daaang awesome well do. nd pfizer dose complete.interest pl
ay rainbowsixsiege free play march th link learn ad. v challenge accept rainbowsIXsiege ad.. game exist ol
d.behold pirate cat.alright must color pirate cat thank guidance.gift cat color'}
```

Conflicting Sentence Detection

In [43]:

```
import spacy
from nltk.corpus import wordnet
#nlp = spacy.load('en')
from spacy import displacy
from collections import Counter
import en_core_web_sm
from nltk.tokenize import word_tokenize
```

In [44]:

```
#Explacy


import sys
from collections import defaultdict

from pprint import pprint

_do_print_debug_info = False

def _print_table(rows):
    col_widths = [max(len(s) for s in col) for col in zip(*rows)]
    fmt = ' '.join('%%-%ds' % width for width in col_widths)
    rows.insert(1, ['—' * width for width in col_widths])
    for row in rows:
        # Uncomment this version to see code points printed out (for debugging).
        # print(list(map(hex, map(ord, list(fmt % tuple(row))))))
        print(fmt % tuple(row))

def _start_end(arrow):
    start, end = arrow['from'].i, arrow['to'].i
    mn = min(start, end)
    mx = max(start, end)
    return start, end, mn, mx

def print_parse_info(nlp, sent):
    """ Print the dependency tree of `sent` (sentence), along with the lemmas
        (de-inflected forms) and parts-of-speech of the words.

        The input `sent` is expected to be a unicode string (of type unicode in
        Python 2; of type str in Python 3). The input `nlp` (for natural
        language parser) is expected to be the return value from a call to
        spacy.load(), in other words, it's the callable instance of a spacy
        language model.
    """

    unicode_type = unicode if sys.version_info[0] < 3 else str
    assert type(sent) is unicode_type

    # Parse our sentence.
    doc = nlp(sent)
```

```
    # Build a list of arrow heights (distance from tokens) per token.
    heights = [[] for token in doc]

    # Build the arrows.

    # Set the from and to tokens for each arrow.
    arrows = [{'from': src, 'to': dst, 'underset': set()}
              for src in doc
              for dst in src.children]

    # Set the base height; these may increase to allow room for arrowheads after this.
    arrows_with_deps = defaultdict(set)
    for i, arrow in enumerate(arrows):
        if _do_print_debug_info:
            print('Arrow %d: "%s" -> "%s"' % (i, arrow['from'], arrow['to']))
        num_deps = 0
        start, end, mn, mx = _start_end(arrow)
        for j, other in enumerate(arrows):
            if arrow is other:
                continue
            o_start, o_end, o_mn, o_mx = _start_end(other)
            if ((start == o_start and mn <= o_end <= mx) or
                (start != o_start and mn <= o_start <= mx)):
                num_deps += 1
                if _do_print_debug_info:
                    print('%d is over %d' % (i, j))
                arrow['underset'].add(j)
        arrow['num_deps_left'] = arrow['num_deps'] = num_deps
        arrows_with_deps[num_deps].add(i)

    if _do_print_debug_info:
        print('')
        print('arrows:')
        pprint(arrows)

        print('')
        print('arrows_with_deps:')
        pprint(arrows_with_deps)

    # Render the arrows in characters. Some heights will be raised to make room for arrowheads.

    lines = [[] for token in doc]
    num_arrows_left = len(arrows)
```

```
  while num_arrows_left > 0:

      assert len(arrows_with_deps[0])

      arrow_index = arrows_with_deps[0].pop()
      arrow = arrows[arrow_index]
      src, dst, mn, mx = _start_end(arrow)

      # Check the height needed.
      height = 3
      if arrow['underset']:
          height = max(arrows[i]['height'] for i in arrow['underset']) + 1
      height = max(height, 3, len(lines[dst]) + 3)
      arrow['height'] = height

      if _do_print_debug_info:
          print('')
          print('Rendering arrow %d: "%s" -> "%s"' % (arrow_index,
                                                      arrow['from'],
                                                      arrow['to']))
          print('  height = %d' % height)

      goes_up = src > dst

      # Draw the outgoing src line.
      if lines[src] and len(lines[src]) < height:
          lines[src][-1].add('w')
      while len(lines[src]) < height - 1:
          lines[src].append(set(['e', 'w']))
      if len(lines[src]) < height:
          lines[src].append({'e'})
      lines[src][height - 1].add('n' if goes_up else 's')

      # Draw the incoming dst line.
      lines[dst].append(u'►')
      while len(lines[dst]) < height:
          lines[dst].append(set(['e', 'w']))
      lines[dst][-1] = set(['e', 's']) if goes_up else set(['e', 'n'])

      # Draw the adjoining vertical line.
      for i in range(mn + 1, mx):
          while len(lines[i]) < height - 1:
              lines[i].append(' ')
```

```
            lines[i].append(set(['n', 's']))

        # Update arrows_with_deps.
        for arr_i, arr in enumerate(arrows):
            if arrow_index in arr['underset']:
                arrows_with_deps[arr['num_deps_left']].remove(arr_i)
                arr['num_deps_left'] -= 1
                arrows_with_deps[arr['num_deps_left']].add(arr_i)

        num_arrows_left -= 1

    arr_chars = {'ew'  : u'─',
                 'ns'  : u'│',
                 'en'  : u'└',
                 'es'  : u'┌',
                 'enw' : u'┴',
                 'ensw': u'┼',
                 'esw' : u'┬'}

    # Convert the character lists into strings.
    max_len = max(len(line) for line in lines)
    for i in range(len(lines)):
        lines[i] = [arr_chars[''.join(sorted(ch))] if type(ch) is set else ch for ch in lines[i]]
        lines[i] = ''.join(reversed(lines[i]))
        lines[i] = ' ' * (max_len - len(lines[i])) + lines[i]

    # Compile full table to print out.
    rows = [['Dep tree', 'Token', 'Dep type', 'Lemma', 'Part of Sp']]
    for i, token in enumerate(doc):
        rows.append([lines[i], token, token.dep_, token.lemma_, token.pos_])
    _print_table(rows)
```

In [45]:

```
#Antonyms finder function
synonyms = []
antonyms = []
#word="went"
def antysyn(word):
    from nltk.corpus import wordnet
    for syn in wordnet.synsets(word):
        #print(syn)
        for l in syn.lemmas():
            #print(l)
            synonyms.append(l.name())
            if l.antonyms():
                antonyms.append(l.antonyms()[0].name())
    #print("Synonym:",set(synonyms))
    print("Antonym:",set(antonyms))
```

In [46]:

```
#Initializing required variables and lists.
wrdlist=list()
antony=list()
contr_tracker=0
antonym_tracker=0

negdoc1=0
negdoc2=0
verb1=""
verb2=""
num_contr_tracker=0
```

In [47]:

```
#Function for checking negation
def checknegationcontradiction(antonym_tracker,negdoc1,negdoc2):
    temp_var=negdoc1+negdoc2+antonym_tracker
    if(temp_var%2!=0 and temp_var<3):
        return 1
    else:
        return 0
```

In [53]:

```
def check_words(doc):
    merged_word=""
    #tokens = word_tokenize(doc)
    WordNum  = len(doc)
    print("WordNum is:"+str(WordNum))
    for i in range(WordNum):
        print(i,doc[i],doc[i].ent_type_)
        print(str.isdigit(doc[i].text))
        if(doc[i].ent_type_=="CARDINAL" or doc[i].pos_=="NUM"):
             merged_word = doc[i].text
        if((doc[i].ent_type_=="CARDINAL" or doc[i].pos_=="NUM") and str.isdigit(doc[i].text)):
            if(not(str.isdigit(doc[i-1].text))):
                if(not(str.isdigit(doc[i-2].text))):
                    merged_word= doc[i-2].text+' '+doc[i-1].text+' '+doc[i].text
    return merged_word
```

In [54]:

```
def check_values(t1,t2):
    for phrase in checklist_more:
        if(t1.find(phrase)!=-1 and t2.find(phrase)==-1):
            num1 = t1.replace(phrase,'')
            num2 = [int(s) for s in str.split(t2) if s.isdigit()]
            num2=num2[0]
            if int(num1)>num2:
                print("case1")
                return('Contradiction')
            else:
                return("No Contradiction")
        else:
            return("No Contradiction")
    for phrase in checklist_more:
        if(t2.find(phrase)!=-1 and t1.find(phrase)==-1):
            num2 = t2.replace(phrase,'')
            num1 = [int(s) for s in str.split(t2) if s.isdigit()]
            num1=num1[0]
            if int(num2)>num1:
                print("case2")
                return('Contradiction')
            else:
                return("No Contradiction")
        else:
            return("No Contradiction")
    for phrase in checklist_less:
        if(t1.find(phrase)!=-1 and t2.find(phrase)==-1):
            num1 = t1.replace(phrase,'')
            num2 = [int(s) for s in str.split(t2) if s.isdigit()]
            num2=num2[0]
            if int(num1)<num2:
                print("case3")
                return('Contradiction')
            else:
                return("No Contradiction")
        else:
                return("No Contradiction")
    for phrase in checklist_less:
        if(t2.find(phrase)!=-1 and t1.find(phrase)==-1):
            num2 = t2.replace(phrase,'')
            num1 = [int(s) for s in str.split(t2) if s.isdigit()]
            num1=num1[0]
```

```
            if int(num1)>num2:
                print("case4")
                return('Contradiction')
            else:
                return("No Contradiction")
        else:
                return("No Contradiction")
    else:
        if(t1!=t2):
            return('Contradiction')
```

Load sentences here

In [58]:

```
#Taking the sentences as input
#sent1=input("Input the first sentence:")
#sent2=input("Input the second sentence:")
contradicting_comms=dict()
for i in range(0,len(com_summary.keys())):
  value_total=[]
  for j in range(i+1,len(com_summary.keys())):
      value_list=[]
      nlp = en_core_web_sm.load()
      sent1=com_summary[i]
      sent2=com_summary[j]
      #print_parse_info(nlp,sent1)
      #print("\n")
      #print_parse_info(nlp,sent2)
      doc1 = nlp(sent1)
      doc2 = nlp(sent2)
      #Processing Sentence 1
      for token in doc1:
          if(token.dep_=="neg"):
              negdoc1=1
              verb1+="NOT "
          #Storing the antonyms of root words
          if(token.pos_=="VERB" and token.dep_=="ROOT"):
              print(token.text)
              verb1+=token.lemma_
              antysyn(token.lemma_)
              for anton in antonyms:
                  antony.append(anton)
      #Processing Sentence 2
      for token in doc2:
          if(token.dep_=="neg"):
              negdoc2=1
              verb2+="NOT "
          if(token.pos_=="VERB" and token.dep_=="ROOT"):
              verb2+=token.lemma_
              if(token.lemma_ in antony):
                  antonym_tracker=1
      #Checking contradiction due to negation
      contr_tracker=checknegationcontradiction(antonym_tracker,negdoc1,negdoc2)
      #Finding numerical mismatch
      checklist_more=['more than ', 'greater than ', 'above']
      checklist_less=['less than ', 'lesser than ', 'below']
```

```
      x1 = check_words(doc1)
      y1 = check_words(doc2)
      number_contr_tracker=check_values(x1,y1)
      if(number_contr_tracker=='Contradiction'):
        num_contr_tracker=1
      else:
        num_contr_tracker=0

      value_list.append(j)
      if contr_tracker==1:
          value_list.append("->Antonymity/Negation contradiction FOUND.")
          value_list.append("REASON: "+verb1.upper()+"and"+verb2.upper()+"can't happen simultaneously.")

      else:
          value_list.append("\n->Antonymity/Negation contradiction NOT found.")

      if num_contr_tracker==1:
          value_list.append("->Numeric Mismatch Contradiction FOUND.")
      else:
          value_list.append("->Numeric Mismatch Contradiction NOT Found.")
      value_total.append(value_list)
  contradicting_comms[i]=value_total
```

```
Streaming output truncated to the last 5000 lines.
False
32 deck
False
33 iconic
False
34 battleship
False
35 back
False
36 uss
False
37 texas
False
38 make
False
39 vanguard
False
40 debut
```

In [59]:
```
contradicting_comms
```

Out[59]:
```
{0: [[1,
   '\n->Antonymity/Negation contradiction NOT found.',
   '->Numeric Mismatch Contradiction NOT Found.'],
  [2,
   '\n->Antonymity/Negation contradiction NOT found.',
   '->Numeric Mismatch Contradiction NOT Found.'],
  [3,
   '\n->Antonymity/Negation contradiction NOT found.',
   '->Numeric Mismatch Contradiction NOT Found.'],
  [4,
   '\n->Antonymity/Negation contradiction NOT found.',
   '->Numeric Mismatch Contradiction NOT Found.'],
  [5,
   '\n->Antonymity/Negation contradiction NOT found.',
   '->Numeric Mismatch Contradiction NOT Found.'],
  [6,
   '->Antonymity/Negation contradiction FOUND.',
   "REASON: PLAYRAINBOWFINDSENDKEEPRAINBOWFINDSENDKEEPRAINBOWFINDSENDKEEPRAINBOWFINDSENDKEEPRAINBOWFIND
SENDKEEPRAINBOWFINDSENDKEEPRAINBOWFINDSENDandSPEAKPIERCESPEAKPIERCESELLTHANKSEEGIVEGETKNOWTAKENEEDLOVES
```