Ukrain Russia War Twitter Sentiment Analysis

Data Collection

In [1]:

```
!pip install snscrape
```

```
Defaulting to user installation because normal site-packages is not writea
ble
Requirement already satisfied: snscrape in c:\users\admin\appdata\roaming
\python\python39\site-packages (0.6.1.20230314)
Requirement already satisfied: filelock in c:\programdata\anaconda3\lib\si
te-packages (from snscrape) (3.6.0)
Requirement already satisfied: beautifulsoup4 in c:\programdata\anaconda3
\lib\site-packages (from snscrape) (4.11.1)
Requirement already satisfied: requests[socks] in c:\programdata\anaconda3
\lib\site-packages (from snscrape) (2.27.1)
Requirement already satisfied: lxml in c:\programdata\anaconda3\lib\site-p
ackages (from snscrape) (4.8.0)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\l
ib\site-packages (from beautifulsoup4->snscrape) (2.3.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\ana
conda3\lib\site-packages (from requests[socks]->snscrape) (1.26.9)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\li
b\site-packages (from requests[socks]->snscrape) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anacon
da3\lib\site-packages (from requests[socks]->snscrape) (2021.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\programdata
\anaconda3\lib\site-packages (from requests[socks]->snscrape) (2.0.4)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in c:\programdata\an
aconda3\lib\site-packages (from requests[socks]->snscrape) (1.7.1)
```

In [ ]:

```
# Data collection from Twitter
import snscrape.modules.twitter as sntwitter
import pandas as pd

query = "Ukrain Russia War"
tweets =[]
limit = 3200

for tweet in sntwitter.TwitterSearchScraper(query).get_items():
    if len(tweets) == limit :
        break
    else:
        tweets.append(tweet.content)

df=pd.DataFrame(tweets)
print(df)
```

In [ ]:

```
df.to_csv("UkrainRussiaWar.csv")
```

In [5]:

```
# Import Required Libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import nltk
import re
from nltk.corpus import stopwords
import string
```

In [7]:

```
#import data and show top 5 rows
data = pd.read_csv(r"C:\Users\ADMIN\OneDrive\Desktop\Projects\ukrain russia\UkrainRussia
print(data.head())
```

```
                    id      conversation_id               created_at  \
0  1630366235354451969  1630152070530576385  2023-02-28 00:36:15 UTC
1  1630366226424778753  1630366226424778753  2023-02-28 00:36:13 UTC
2  1630366225930027011  1630366225930027011  2023-02-28 00:36:13 UTC
3  1630366223056662530  1630351686974992385  2023-02-28 00:36:12 UTC
4  1630366221483884545  1629903982255644672  2023-02-28 00:36:12 UTC

         date      time  timezone              user_id     username  \
0  2023-02-28  00:36:15         0  1493761817406894086  tomasliptai
1  2023-02-28  00:36:13         0  1526694166662721536  paperfloure
2  2023-02-28  00:36:13         0  1053018392939167746    katetbar1
3  2023-02-28  00:36:12         0            602371247    jlhrdhmom
4  2023-02-28  00:36:12         0  1053594763214184448    phemikali

                  name place  ... geo source user_rt_id user_rt retweet_id
\
0         Tomas Liptai   NaN  ... NaN    NaN        NaN     NaN        NaN
1      Smell the roses   NaN  ... NaN    NaN        NaN     NaN        NaN
2                @etak   NaN  ... NaN    NaN        NaN     NaN        NaN
3               JLHrdh   NaN  ... NaN    NaN        NaN     NaN        NaN
4  rolarkcybersecurity   NaN  ... NaN    NaN        NaN     NaN        NaN

                                          reply_to  retweet_date  transl
ate  \
0  [{'screen_name': 'nazijaeger__', 'name': 'nazi...           NaN
NaN
1                                                []           NaN
NaN
2                                                []           NaN
NaN
3  [{'screen_name': 'MainelifeR', 'name': 'Mainel...           NaN
NaN
4  [{'screen_name': 'Pottingpinks', 'name': 'GRS'...           NaN
NaN

  trans_src trans_dest
0       NaN        NaN
1       NaN        NaN
2       NaN        NaN
3       NaN        NaN
4       NaN        NaN

[5 rows x 36 columns]
```

In [8]:

```
# column names
print(data.columns)
```

```
Index(['id', 'conversation_id', 'created_at', 'date', 'time', 'timezone',
       'user_id', 'username', 'name', 'place', 'tweet', 'language', 'menti
ons',
       'urls', 'photos', 'replies_count', 'retweets_count', 'likes_count',
       'hashtags', 'cashtags', 'link', 'retweet', 'quote_url', 'video',
       'thumbnail', 'near', 'geo', 'source', 'user_rt_id', 'user_rt',
       'retweet_id', 'reply_to', 'retweet_date', 'translate', 'trans_src',
       'trans_dest'],
      dtype='object')
```

In [9]:

```
# Required columns
data = data[["username", "tweet", "language"]]
```

In [10]:

```
# Missing values and there sum
data.isnull().sum()
```

Out[10]:

```
username    0
tweet       0
language    0
dtype: int64
```

Look like there is no missing value

In [11]:

```
# check how many tweets are available in which languages
data["language"].value_counts()
```

Out[11]:

```
en     8858
pt      440
it      194
qme     105
und      60
in       47
ru       44
ja       42
es       36
ca       20
qht      20
th       19
fr       18
de       14
ko        9
vi        8
nl        8
ro        7
fi        7
ar        6
zxx       6
uk        6
cs        6
zh        5
pl        5
qam       4
tl        4
da        3
eu        2
no        2
hi        2
tr        2
hu        1
cy        1
lv        1
el        1
bn        1
Name: language, dtype: int64
```

Most of the tweets are in english language

In [13]:

```
# Download english stopwords
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
stopword=set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\ADMIN\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

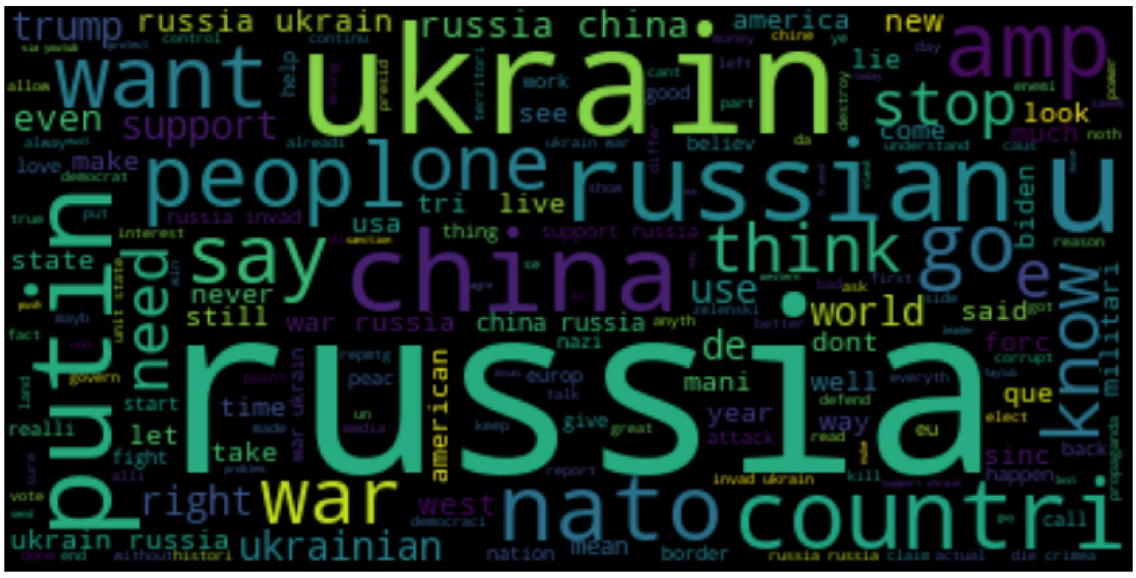
In [14]:

```
# Removing links, punctuation, symbols etc
def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["tweet"] = data["tweet"].apply(clean)
```

Wordcloud of preprocess tweets

In [16]:

```
# wordcloud of most frequent words
text = " ".join(i for i in data.tweet)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, background_color="Black").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



In [17]:

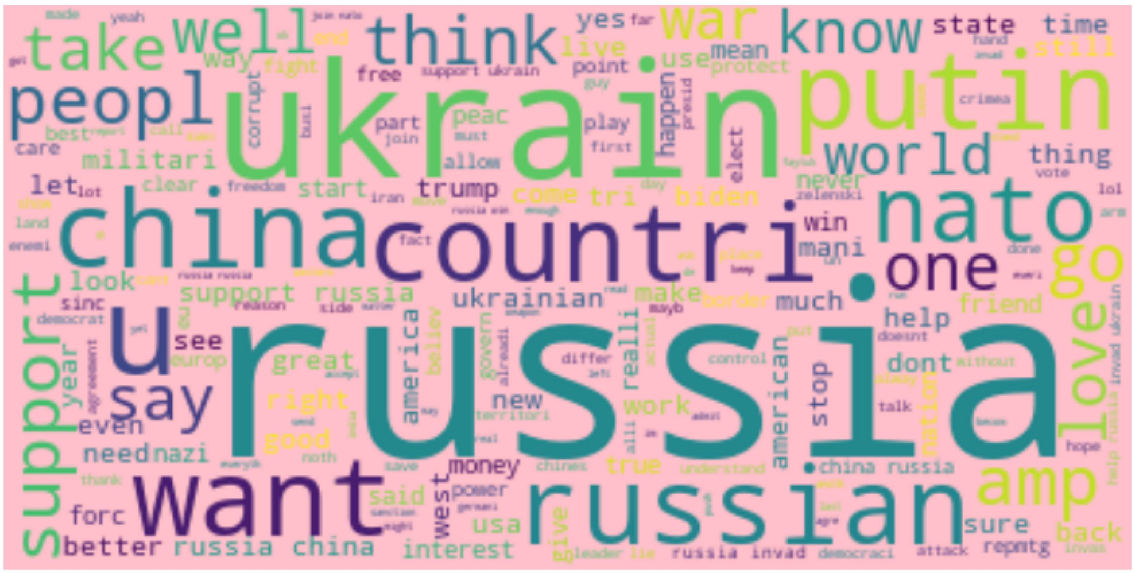
```
# Adding and checking sentiment score of Positive, Negative, Neutral
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i in data["tweet"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i in data["tweet"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i in data["tweet"]]
data = data[["tweet", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\ADMIN\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!

                                               tweet  Positive  Negative
\
0      nazijaeg derwen  russia place satan rule well     0.259     0.000
1  russia haarp could destroy usa one fell swoop ...     0.000     0.280
2       putin give steven seagal  order friendship       0.367     0.000
3       mainelif baddcompani it alway project russia     0.000     0.000
4  pottingpink mfarussia modrussia milhistrf muze...     0.068     0.078

   Neutral
0    0.741
1    0.720
2    0.633
3    1.000
4    0.854
```

Positive Wordcloud

In [21]:

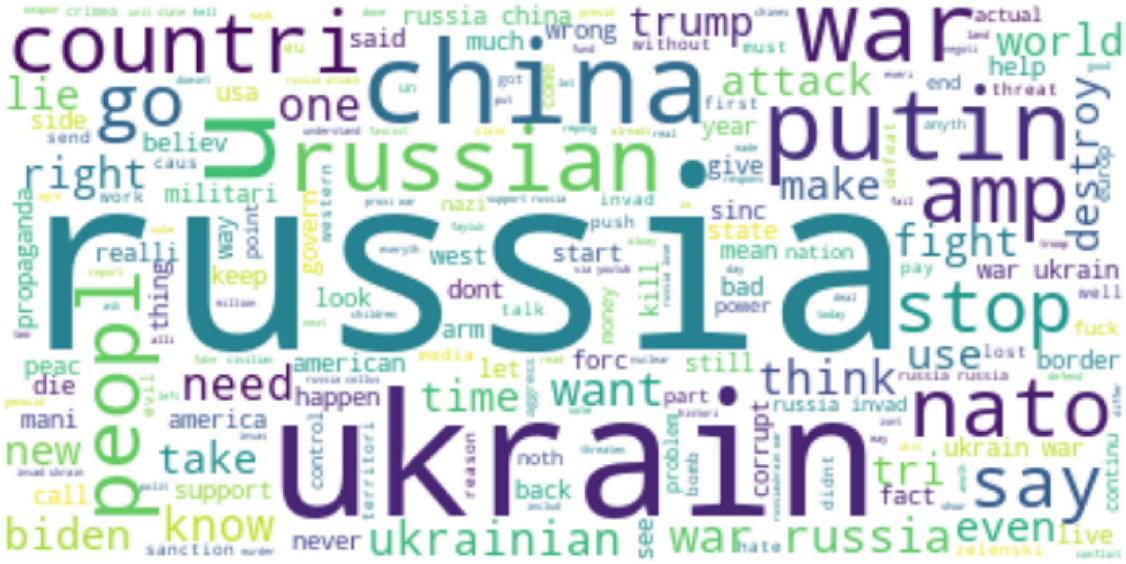
```
# frequent word with positive sentiment
positive =' '.join([i for i in data['tweet'][data['Positive'] > data["Negative"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, background_color="Pink").generate(positive)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



Negative Wordcloud

In [22]:

```
# frequent word with negative sentiment
negative =' '.join([i for i in data['tweet'][data['Negative'] > data["Positive"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(negative)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



Summary

There are a lot of tweets about the Ukraine and Russia war where people tend to update about the ground truths, what they feel about it, and who they are supporting. I used those tweets for the task of Twitter sentiment analysis on the Ukraine and Russia war.

In [ ]: