

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import string
import re
import nltk
from nltk.util import pr
from nltk.corpus import stopwords
import warnings
warnings.filterwarnings('ignore')
stemmer = nltk.SnowballStemmer("english")
nltk.download('stopwords')
stopword=set(stopwords.words('english'))
```

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

In [6]:

```
data = pd.read_excel("war_tweets.xls")
```

In [8]:

```
data.head()
```

Out[8]:

	id	conversation_id	created_at	date	time	timezone	
0	1504325620976819968	1504082706598139904	2022-03-17 05:15:51 UTC	2022-03-17	05:15:51	0	
1	1504325620179909888	1504323038044509952	2022-03-17 05:15:51 UTC	2022-03-17	05:15:51	0	1420236
2	1504325618829250048	1504325618829250048	2022-03-17 05:15:51 UTC	2022-03-17	05:15:51	0	1387736
3	1504325616589489920	1504325616589489920	2022-03-17 05:15:50 UTC	2022-03-17	05:15:50	0	
4	1504325616320989952	1504324574766320128	2022-03-17 05:15:50 UTC	2022-03-17	05:15:50	0	

5 rows × 36 columns

In [9]:

```
data.tail()
```

Out[9]:

	id	conversation_id	created_at	date	time	timezone	
10009	1504308144968760064	1503515544871439872	2022-03-17 04:06:25 UTC	2022-03-17	04:06:25	0	148
10010	1504308143953779968	1504308143953779968	2022-03-17 04:06:24 UTC	2022-03-17	04:06:24	0	150
10011	1504308143399920128	1486861730202459904	2022-03-17 04:06:24 UTC	2022-03-17	04:06:24	0	147
10012	1504308142120869888	1504288918430269952	2022-03-17 04:06:24 UTC	2022-03-17	04:06:24	0	123
10013	1504308140199790080	1504110924730619904	2022-03-17 04:06:23 UTC	2022-03-17	04:06:23	0	146

5 rows × 36 columns

In [10]:

```
data.shape
```

Out[10]:

(10014, 36)

In [11]:



```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10014 entries, 0 to 10013
Data columns (total 36 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                     10014 non-null  int64
1   conversation_id        10014 non-null  int64
2   created_at             10014 non-null  object
3   date                   10014 non-null  datetime64[ns]
4   time                   10014 non-null  object
5   timezone               10014 non-null  int64
6   user_id                10014 non-null  int64
7   username               10014 non-null  object
8   name                   10014 non-null  object
9   place                  1 non-null      object
10  tweet                  10014 non-null  object
11  language                10014 non-null  object
12  mentions                10014 non-null  object
13  urls                    10014 non-null  object
14  photos                  10014 non-null  object
15  replies_count           10014 non-null  int64
16  retweets_count          10014 non-null  int64
17  likes_count             10014 non-null  int64
18  hashtags                10014 non-null  object
19  cashtags                10014 non-null  object
20  link                    10014 non-null  object
21  retweet                 10014 non-null  bool
22  quote_url               876 non-null    object
23  video                   10014 non-null  int64
24  thumbnail               936 non-null    object
25  near                    0 non-null      float64
26  geo                     0 non-null      float64
27  source                  0 non-null      float64
28  user_rt_id              0 non-null      float64
29  user_rt                 0 non-null      float64
30  retweet_id              0 non-null      float64
31  reply_to                10014 non-null  object
32  retweet_date            0 non-null      float64
33  translate               0 non-null      float64
34  trans_src               0 non-null      float64
35  trans_dest              0 non-null      float64
dtypes: bool(1), datetime64[ns](1), float64(10), int64(8), object(16)
memory usage: 2.7+ MB
```

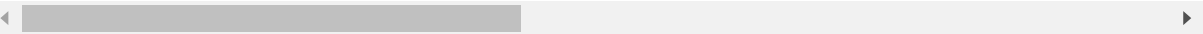
In [12]:



```
data.describe()
```

Out[12]:

	id	conversation_id	timezone	user_id	replies_count	retweets_count	
count	1.001400e+04	1.001400e+04	10014.0	1.001400e+04	10014.000000	10014.000000	1
mean	1.504317e+18	1.502877e+18	0.0	6.984499e+17	0.313661	0.552227	
std	5.075717e+12	2.728863e+16	0.0	6.443610e+17	2.549457	10.848945	
min	1.504308e+18	4.371802e+17	0.0	7.421430e+05	0.000000	0.000000	
25%	1.504312e+18	1.504181e+18	0.0	4.921743e+08	0.000000	0.000000	
50%	1.504317e+18	1.504309e+18	0.0	8.388104e+17	0.000000	0.000000	
75%	1.504321e+18	1.504316e+18	0.0	1.354872e+18	0.000000	0.000000	
max	1.504326e+18	1.504326e+18	0.0	1.504322e+18	142.000000	666.000000	



In [13]:



```
data.isnull().sum()
```

Out[13]:

id	0
conversation_id	0
created_at	0
date	0
time	0
timezone	0
user_id	0
username	0
name	0
place	10013
tweet	0
language	0
mentions	0
urls	0
photos	0
replies_count	0
retweets_count	0
likes_count	0
hashtags	0
cashtags	0
link	0
retweet	0
quote_url	9138
video	0
thumbnail	9078
near	10014
geo	10014
source	10014
user_rt_id	10014
user_rt	10014
retweet_id	10014
reply_to	0
retweet_date	10014
translate	10014
trans_src	10014
trans_dest	10014

dtype: int64

In [14]:



```
data.columns
```

Out[14]:

```
Index(['id', 'conversation_id', 'created_at', 'date', 'time', 'timezone',
      'user_id', 'username', 'name', 'place', 'tweet', 'language', 'mentions',
      '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 [15]:



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

Out[15]:

	tweet
0	@PeterSchiff @PadaPrabu @SteveKrohn1 If it wer...
1	@meatballsubzero Are you pro russia or pro Ukr...
2	@SUBWAY Please stop doing business in Russia....
3	Is Russia prepared for an economic crisis? Dev...
4	@BW Putin is Fake News ðŸ™° The Ruble is trash...

In [16]:



```
data["language"].value_counts()
```

Out[16]:

en	9018
pt	211
und	158
it	118
hi	80
in	79
ru	69
ja	54
es	22
pl	19
tl	18
nl	15
de	14
ar	13
fr	13
zh	11
th	10
ca	9
ta	8
ro	6
et	6
or	5
fi	5
bn	5
mr	5
ne	5
uk	4
kn	4
ml	4
cs	4
te	3
no	3
el	3
gu	3
ur	3
tr	2
iw	2
sl	1
fa	1
am	1

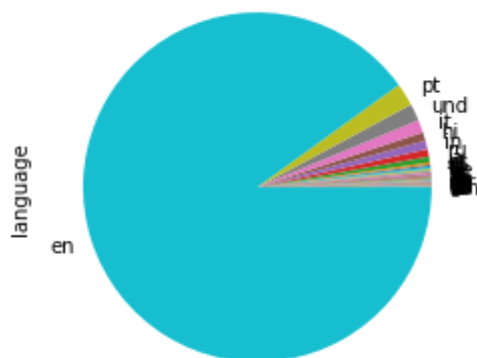
Name: language, dtype: int64

In [23]:

```
data.language.value_counts().sort_values().plot(kind = 'pie')
```

Out[23]:

<matplotlib.axes._subplots.AxesSubplot at 0xa33d4921c0>



In [24]:

```
data["tweet"][0]
```

Out[24]:

'@PeterSchiff @PadaPrabu @SteveKrohn1 If it were you you would have shit y
our pants and changed the name of your country to Russia.'

In [25]:

```
def hashtag_extract(text_list):
    hashtags = []
    # Loop over the words in the tweet
    for text in text_list:
        ht = re.findall(r"#(\w+)", text)
        hashtags.append(ht)

    return hashtags

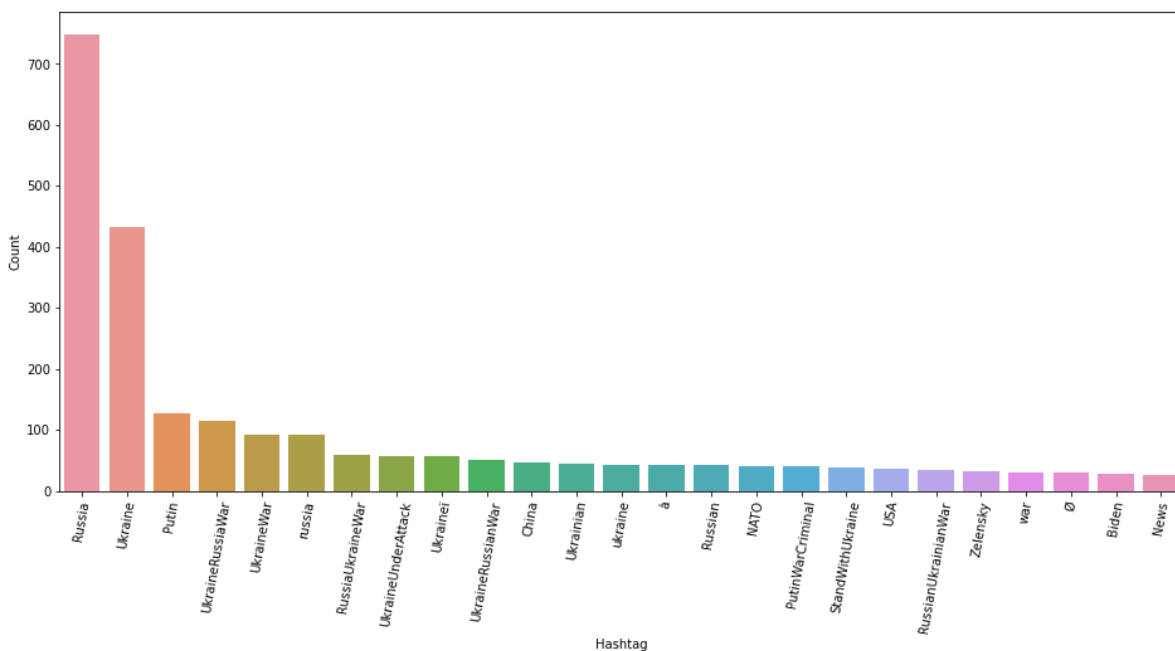
def generate_hashtag_freqdist(hashtags):
    a = nltk.FreqDist(hashtags)
    d = pd.DataFrame({'Hashtag': list(a.keys()),
                      'Count': list(a.values())})
    # selecting top 15 most frequent hashtags
    d = d.nlargest(columns="Count", n = 25)
    plt.figure(figsize=(16,7))
    ax = sns.barplot(data=d, x= "Hashtag", y = "Count")
    plt.xticks(rotation=80)
    ax.set(ylabel = 'Count')
    plt.show()
```

In [27]:

```
hashtags = hashtag_extract(data["tweet"])
hashtags = sum(hashtags, [])
```

In [28]:

```
generate_hashtag_freqdist(hashtags)
```



In [29]:



```
data['total_length_characters'] = data['tweet'].str.len()
print(data['total_length_characters'])
total_length_characters = data['total_length_characters'].sum()
print(total_length_characters)
count = 0
for y in data["tweet"]:
    count = count + 1
print(count)
average_length = total_length_characters / count
print (average_length)
```

```
0      130
1      162
2      167
3      220
4       81
...
10009   255
10010    84
10011   176
10012   249
10013   216
Name: total_length_characters, Length: 10014, dtype: int64
1809200
10014
180.66706610744956
```

In [30]:



```
data['total_count_words'] = data['tweet'].str.split().str.len()
print(data['total_count_words'])
total_words = data['total_count_words'].sum()
print(total_words)
count = 0
for y in data["tweet"]:
    count = count + 1
print(count)
average_words = total_words / count
print (average_words)
```

```
0      22
1      28
2      26
3      32
4      15
..
10009   44
10010   11
10011   32
10012   39
10013   32
Name: total_count_words, Length: 10014, dtype: int64
271703
10014
27.13231475933693
```

In [31]:



```
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 stopwords]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["tweet"] = data["tweet"].apply(clean)
```

In [32]:



```
data['total_length_characters'] = data['tweet'].str.len()
print(data['total_length_characters'])
total_length_characters = data['total_length_characters'].sum()
print(total_length_characters)
count = 0
for y in data["tweet"]:
    count = count + 1
print(count)
average_length = total_length_characters / count
print (average_length)
```

```
0          64
1          98
2         121
3         134
4          63
...
10009      126
10010       74
10011      115
10012      130
10013      137
Name: total_length_characters, Length: 10014, dtype: int64
1151855
10014
115.02446574795286
```

In [33]:



```
data['total_count_words'] = data['tweet'].str.split().str.len()
print(data['total_count_words'])
total_words = data['total_count_words'].sum()
print(total_words)
count = 0
for y in data["tweet"]:
    count = count + 1
print(count)
average_words = total_words / count
print (average_words)
```

```
0          9
1         16
2         19
3         19
4         11
..
10009      20
10010      10
10011      19
10012      20
10013      18
Name: total_count_words, Length: 10014, dtype: int64
163755
10014
16.35260635110845
```

In [34]:



```
from textblob import TextBlob
```

In [35]:



```
def analyze_sentiment(tweet):
    analysis = TextBlob(clean(tweet))
    if analysis.sentiment.polarity > 0:
        return 1
    elif analysis.sentiment.polarity == 0:
        return 0
    else:
        return -1
```

In [36]:



```
data['Sentiment'] = data['tweet'].apply(lambda x:analyze_sentiment(x))
data['Source'] = 'random_user'
data['Length'] = data['tweet'].apply(len)
data['Word_counts'] = data['tweet'].apply(lambda x:len(str(x).split()))
```

In [37]:

```
data1=data[['tweet','retweets_count', 'Sentiment', 'Source',
            'Length','Word_counts']]
data1.head()
```

Out[37]:

	tweet	retweets_count	Sentiment	Source	Length	Word_counts
0	peterschiff padaprabu would shit pant chang n...	0	-1	random_user	64	9
1	meatballsubzero pro russia pro ukraine cannot ...	0	0	random_user	98	16
2	subway pleas stop busi russia everi dollar sp...	0	1	random_user	121	19
3	russia prepar econom crisi develop expert nata...	0	0	random_user	134	19
4	bw putin fake news ðŸ“° rubl trash ðŸ— russi...	0	-1	random_user	63	11

In [39]:

```
data1['Clean tweet'] = data1['tweet'].apply(lambda x:clean(x))
```

In [40]:

```
data1[["Clean tweet","Sentiment"]].iloc[100]
```

Out[40]:

```
Clean tweet    ewarren war russia putin peopl wef go peopl la...
Sentiment                                           0
Name: 100, dtype: object
```

In [41]:

```
sentiment = data1['Sentiment'].value_counts()
sentiment
```

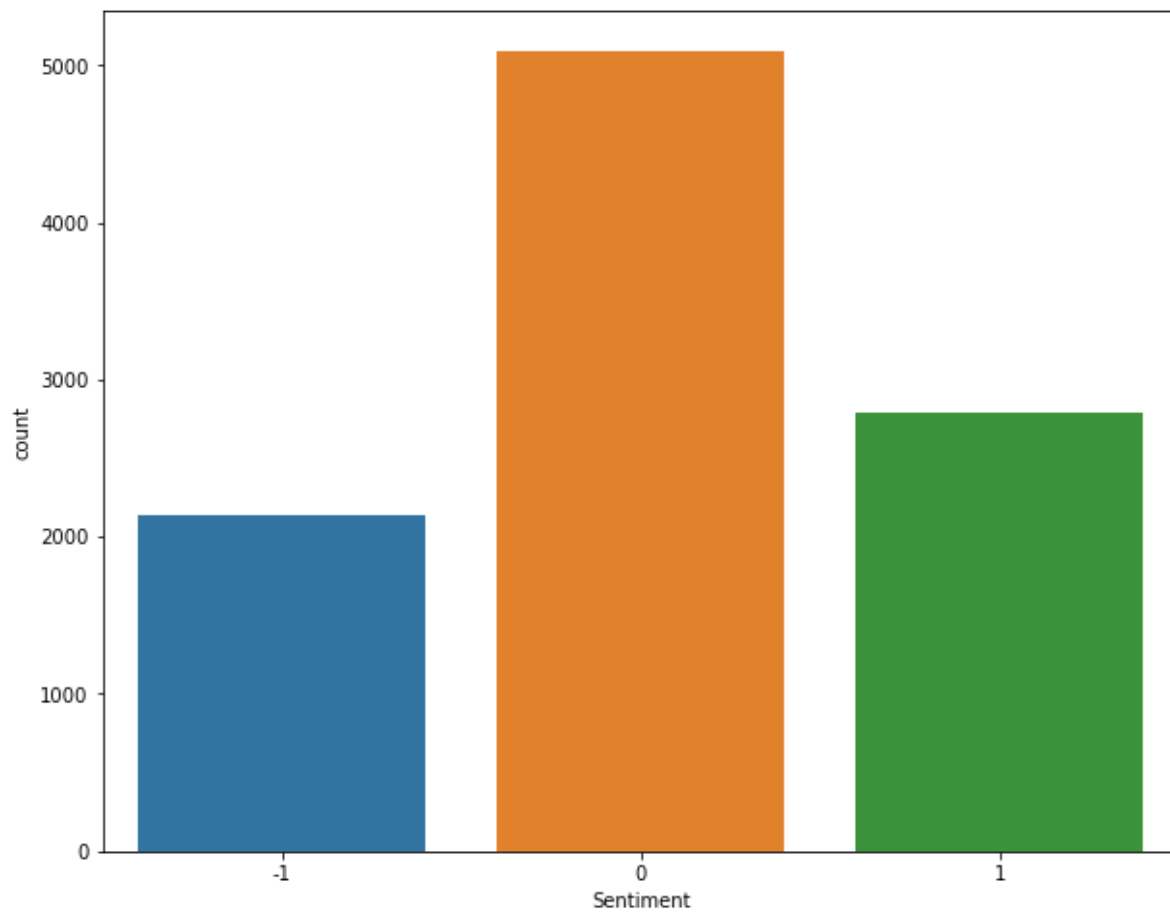
Out[41]:

```
0    5094
1    2788
-1   2132
Name: Sentiment, dtype: int64
```

In [42]:



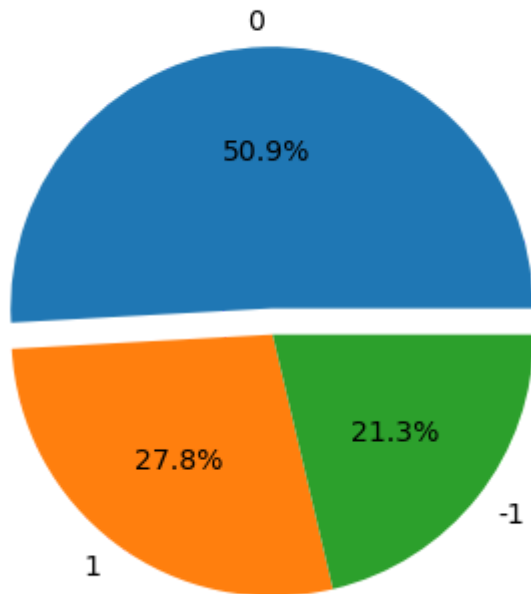
```
plt.figure(figsize = (10,8))  
sns.countplot(data = data1, x = 'Sentiment')  
plt.show()
```



In [43]:

```
fig, ax = plt.subplots(figsize = (6, 6))
sizes = [count for count in data1['Sentiment'].value_counts()]
labels = list(data1['Sentiment'].value_counts().index)
explode = (0.1, 0, 0)
ax.pie(x = sizes, labels = labels, autopct = '%1.1f%', explode = explode, textprops={'color': 'black'})
ax.set_title('Sentiment Polarity on invasion Tweets Data \n (total = 9127 tweets)', fontweight='bold')
plt.show()
```

Sentiment Polarity on invasion Tweets Data
(total = 9127 tweets)



In [46]:

```
neutral = data1[data1['Sentiment'] == 0]
positive = data1[data1['Sentiment'] == 1]
negative = data1[data1['Sentiment'] == -1]
```

In [47]:



```
negative.iloc[1]
```

Out[47]:

```
tweet          bw putin fake news ðŸ“° rubl trash ðŸ– russi...
retweets_count          0
Sentiment              -1
Source                random_user
Length                63
Word_counts            11
Clean tweet          bw putin fake news ðŸ“° rubl trash ðŸ– russi...
Name: 4, dtype: object
```

In [48]:



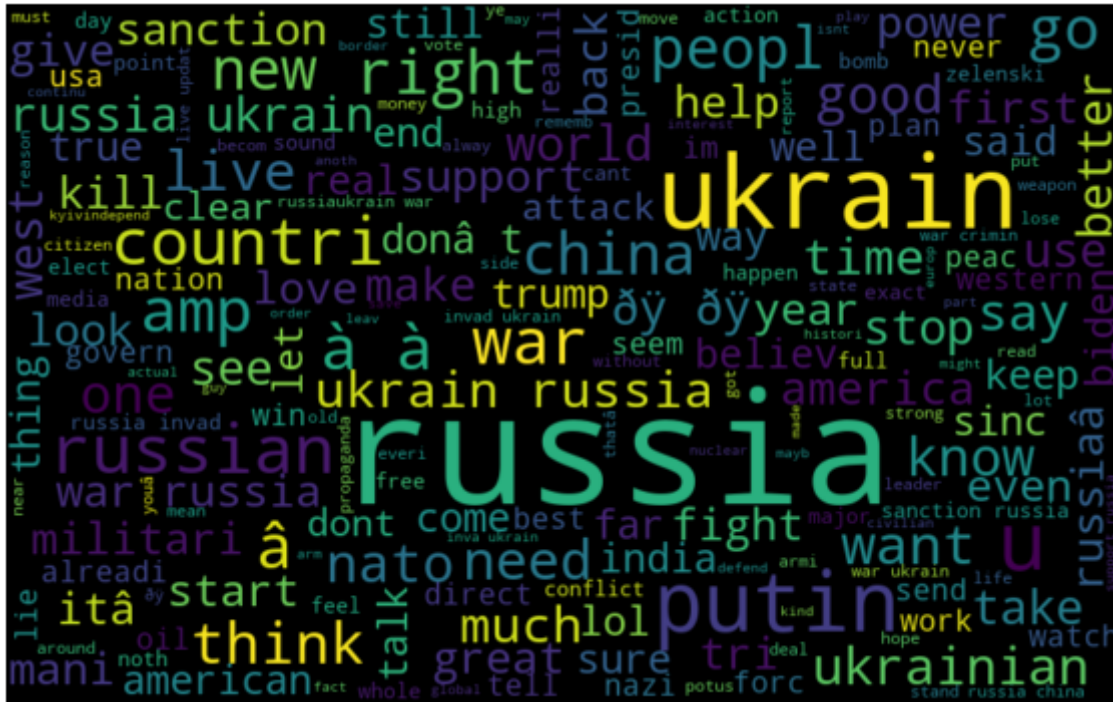
```
#neutral_text
print("Neutral tweet example :",neutral['tweet'].values[15])
# Positive tweet
print("Positive Tweet example :",positive['tweet'].values[37])
#negative_text
print("Negative Tweet example :",negative['tweet'].values[1])
```

```
Neutral tweet example : us allow russia troop cuba mexico attack said cou
ntri
Positive Tweet example : new post media israeliani russia aperta idea di n
egoziati gerusalemm â€œ
Negative Tweet example : bw putin fake news ðŸ“° rubl trash ðŸ– russia hi
stori ðŸ'ðŸ½
```

In [53]:



```
from wordcloud import WordCloud
```



In [57]:

```
neutral_words = ' '.join([text for text in data1['Clean tweet'] if data1['Sentiment'] == 0])
wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(neutral_words)
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```

