TASK 3 - FAKE NEWS DETECTION WITH NLP AND LSTM

Import the libraries required

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import nltk
import re
import string
from nltk.corpus import stopwords
import gensim
from gensim import parsing
from wordcloud import WordCloud,STOPWORDS

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer,TfidfTransformer
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.linear_model import LogisticRegression

import keras
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.preprocessing import text,sequence
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Dense,Embedding,LSTM,Dropout,Bidirectional
```

In [2]:
```
real_data=pd.read_csv('True.csv')
fake_data=pd.read_csv('Fake.csv')
```

In [3]:
```
real_data.head()
```

Out[3]:

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

In [4]:
```
fake_data.head()
```

Out[4]:

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

In [5]:
```
real_data['target']=1
fake_data['target']=0
```

In [6]:
```
real_data.tail()
```

Out[6]:

	title	text	subject	date	target
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	1
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	1
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	1
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	1
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	1

In [7]:
```
fake_data.tail()
```

Out[7]:

	title	text	subject	date	target
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016	0
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016	0
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016	0
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016	0
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016	0

In [8]:
```
#merging fake and real data
data=pd.concat([real_data,fake_data])
```

In [9]:
```
data.head(10)
```

Out[9]:

	title	text	subject	date	target

0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	1
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	1
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	1
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	1
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017	1
5	White House, Congress prepare for talks on spe...	WEST PALM BEACH, Fla./WASHINGTON (Reuters) - T...	politicsNews	December 29, 2017	1
6	Trump says Russia probe will be fair, but time...	WEST PALM BEACH, Fla (Reuters) - President Don...	politicsNews	December 29, 2017	1
7	Factbox: Trump on Twitter (Dec 29) - Approval ...	The following statements were posted to the ve...	politicsNews	December 29, 2017	1
8	Trump on Twitter (Dec 28) - Global Warming	The following statements were posted to the ve...	politicsNews	December 29, 2017	1
9	Alabama official to certify Senator-elect Jone...	WASHINGTON (Reuters) - Alabama Secretary of St...	politicsNews	December 28, 2017	1

In [10]:
```
data.tail(10)
```

Out[10]:

	title	text	subject	date	target
23471	Seven Iranians freed in the prisoner swap have...	21st Century Wire says This week, the historic...	Middle-east	January 20, 2016	0
23472	#Hashtag Hell & The Fake Left	By Dady Chery and Gilbert MercierAll writers ...	Middle-east	January 19, 2016	0
23473	Astroturfing: Journalist Reveals Brainwashing ...	Vic Bishop Waking TimesOur reality is carefull...	Middle-east	January 19, 2016	0
23474	The New American Century: An Era of Fraud	Paul Craig RobertsIn the last years of the 20t...	Middle-east	January 19, 2016	0
23475	Hillary Clinton: 'Israel First' (and no peace ...	Robert Fantina CounterpunchAlthough the United...	Middle-east	January 18, 2016	0
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016	0
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016	0
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016	0
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016	0
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016	0

In [11]:
```
data.isnull().sum()
```

Out[11]:
```
title      0
text       0
subject    0
date       0
target     0
dtype: int64
```

Data contains no null value.

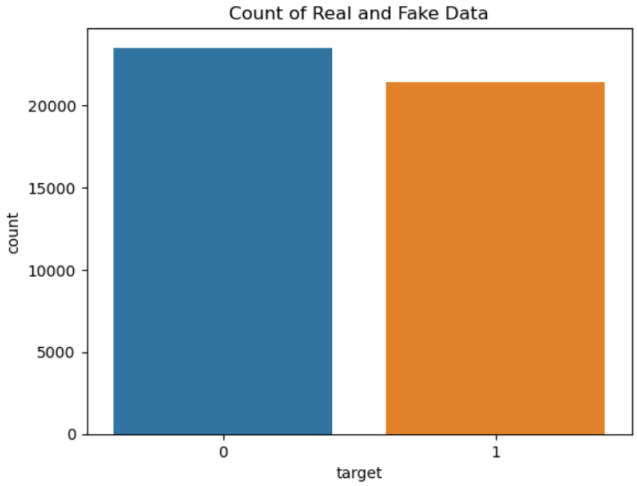
Visualization

In [12]:
```
print(data['target'].value_counts())
```

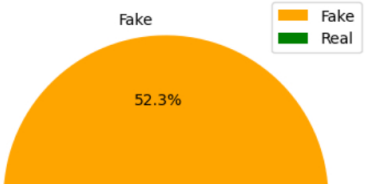
```
0    23481
1    21417
Name: target, dtype: int64
```

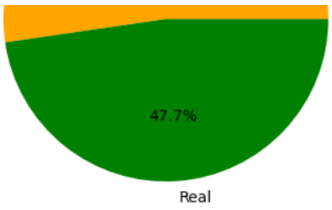
In [13]:
```
plot=sns.countplot(x='target', data=data)
plot.set_title('Count of Real and Fake Data')
```

Out[13]:
```
Text(0.5, 1.0, 'Count of Real and Fake Data')
```



In [14]:
```
mylabels=["Fake", "Real"]
mycolors=['orange', 'green']
plt.pie(data['target'].value_counts(), labels=mylabels, colors=mycolors,autopct='%1.1f%%')
plt.legend()
plt.show()
```



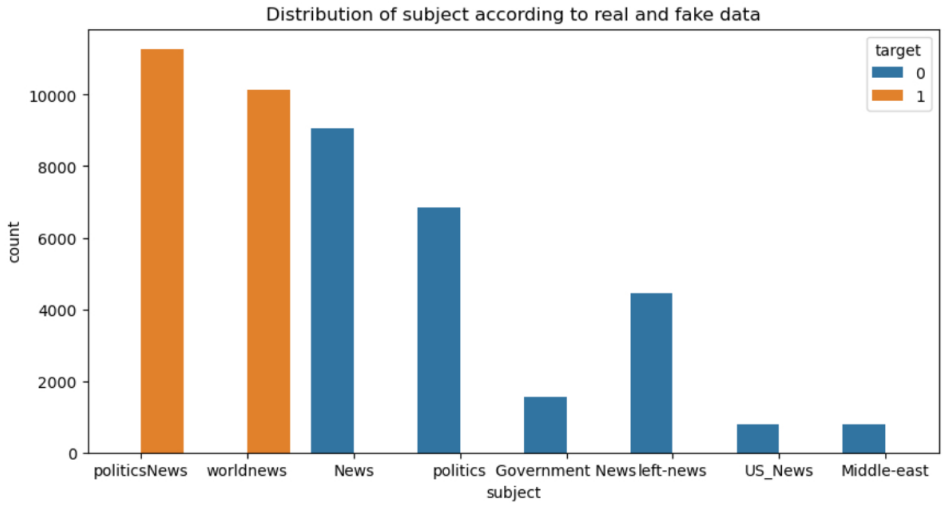


In [15]:
```
print(data['subject'].value_counts())
```

```
politicsNews       11272
worldnews          10145
News                9050
politics            6841
left-news           4459
Government News     1570
US_News              783
Middle-east          778
Name: subject, dtype: int64
```

In [16]:
```
plt.figure(figsize=(10,5))
sns.countplot(x='subject', hue='target', data=data)
plt.title("Distribution of subject according to real and fake data")
```

Out[16]:
```
Text(0.5, 1.0, 'Distribution of subject according to real and fake data')
```



Data Cleaning

In [17]:
```
data.drop('title', inplace=True, axis=1)
data.drop('date', inplace=True, axis=1)
data.drop('subject', inplace=True, axis=1)
```

In [18]:
```
data.tail(10)
```

Out[18]:

	text	target
23471	21st Century Wire says This week, the historic...	0
23472	By Dady Chery and Gilbert MercierAll writers ...	0
23473	Vic Bishop Waking TimesOur reality is carefull...	0
23474	Paul Craig RobertsIn the last years of the 20t...	0
23475	Robert Fantina CounterpunchAlthough the United...	0
23476	21st Century Wire says As 21WIRE reported earl...	0
23477	21st Century Wire says It s a familiar theme. ...	0
23478	Patrick Henningsen 21st Century WireRemember ...	0
23479	21st Century Wire says Al Jazeera America will...	0
23480	21st Century Wire says As 21WIRE predicted in ...	0

Data Preprocessing

In [19]:
```
def transformText(text):
  # All the necessary preprocessing on our text of choice
    stops = set(stopwords.words("english"))
  # Convert text to lower
    text = text.lower()
  # Removing non ASCII chars
    text = re.sub(r'[^\x00-\x7f]',r' ',text)
    text = re.sub('\[[^]]*\]', '', text)
    text= gensim.parsing.preprocessing.strip_non_alphanum(text)
  # Strip multiple whitespaces
    text = gensim.corpora.textcorpus.strip_multiple_whitespaces(text)
  # Removing all the stopwords
    filtered_words = [word for word in text.split() if word not in stops]
  # Removing all the tokens with lesser than 3 characters
    filtered_words = gensim.corpora.textcorpus.remove_short(filtered_words, minsize=3)
```

```
    # Preprocessed text after stop words removal
      text = " ".join(filtered_words)
    # Remove the punctuation
      text = gensim.parsing.preprocessing.strip_punctuation2(text)
    # Strip all the numerics
      text = gensim.parsing.preprocessing.strip_numeric(text)
    # Strip multiple whitespaces
      text = gensim.corpora.textcorpus.strip_multiple_whitespaces(text)
    # Stemming
      return gensim.parsing.preprocessing.stem_text(text)
```

In [ ]:
```
data['text']=data['text'].apply(transformText)
```

In [20]:
```
import gensim
from gensim.parsing.preprocessing import STOPWORDS

def transformText(text):
    # Tokenize the text
    tokens = gensim.utils.simple_tokenize(text)

    # Remove stopwords and short words (less than 3 characters)
    filtered_tokens = [word for word in tokens if word not in STOPWORDS and len(word) >= 3]

    # Join the filtered tokens back into a text string
    text = " ".join(filtered_tokens)

    return text

# Assuming 'data' is your DataFrame and 'text' is the column you want to transform
data['text'] = data['text'].apply(transformText)
```

In [22]:
```
data.head(20)
```

Out[22]:

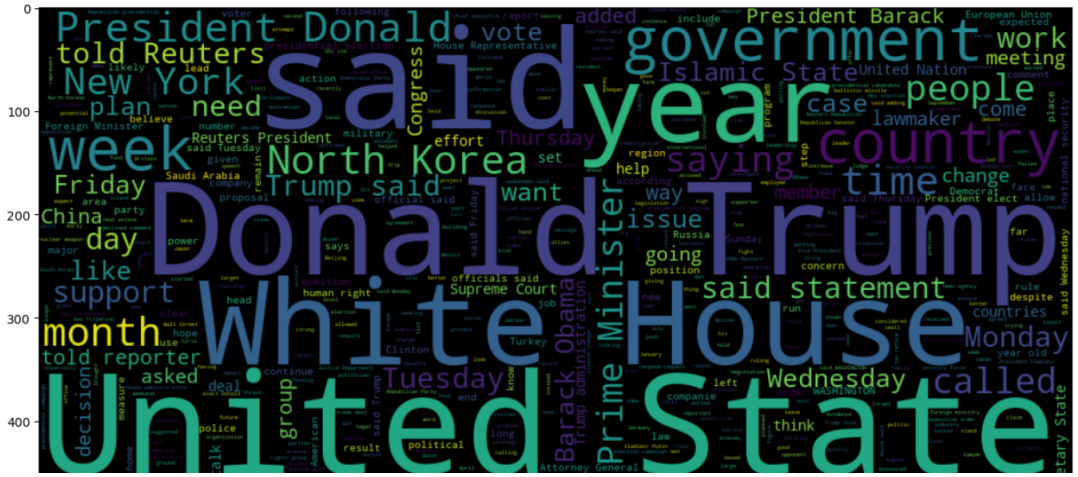
	text	target
0	WASHINGTON Reuters The head conservative Repub...	1
1	WASHINGTON Reuters Transgender people allowed ...	1
2	WASHINGTON Reuters The special counsel investi...	1
3	WASHINGTON Reuters Trump campaign adviser Geor...	1
4	SEATTLE WASHINGTON Reuters President Donald Tr...	1
5	WEST PALM BEACH Fla WASHINGTON Reuters The Whi...	1
6	WEST PALM BEACH Fla Reuters President Donald T...	1
7	The following statements posted verified Twitt...	1
8	The following statements posted verified Twitt...	1
9	WASHINGTON Reuters Alabama Secretary State Joh...	1
10	Reuters Alabama officials Thursday certified D...	1
11	NEW YORK WASHINGTON Reuters The new tax code t...	1
12	The following statements posted verified Twitt...	1
13	The following statements posted verified Twitt...	1
14	Dec story second paragraph corrects Strong emp...	1
15	Reuters lottery drawing settle tied Virginia l...	1
16	WASHINGTON Reuters Georgian American businessm...	1
17	The following statements posted verified Twitt...	1
18	Reuters appeals court Washington Tuesday uphel...	1
19	Reuters gift wrapped package addressed Treasur...	1

A wordcloud is a visual reprsentation of text data. Words are usually single words, and the importance of each is shown with font size or color. Pyhton has a wordcloud library allowing to build them

WordCloud for Real News

In [23]:
```
plt.figure(figsize = (15,15))
wc = WordCloud(max_words = 500 , width = 1000 , height = 500 , stopwords = STOPWORDS).generate(" ".join(data[data.target == 1].te
plt.imshow(wc , interpolation = 'bilinear')
```

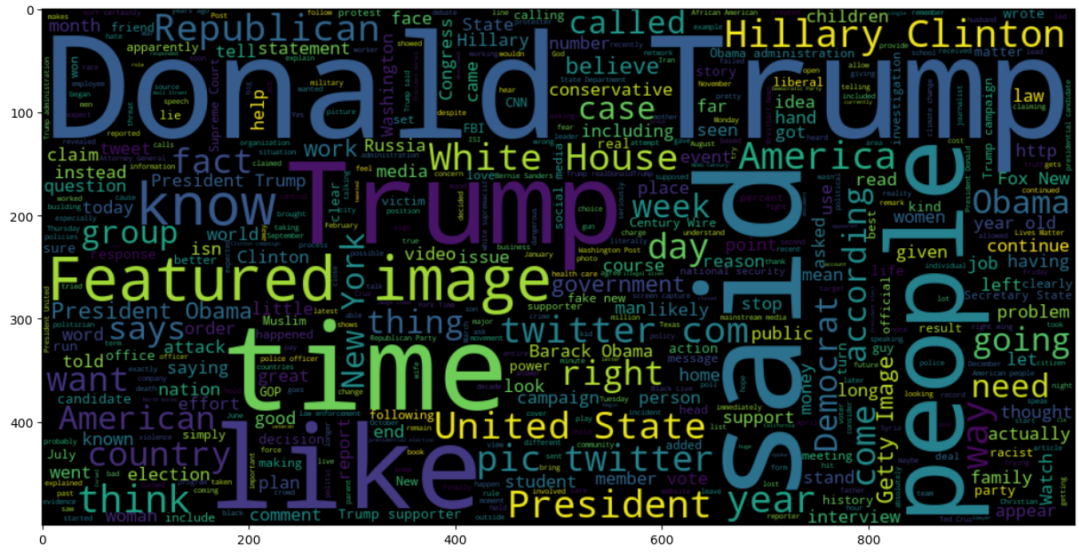
Out[23]:
```
<matplotlib.image.AxesImage at 0x2039062efb0>
```



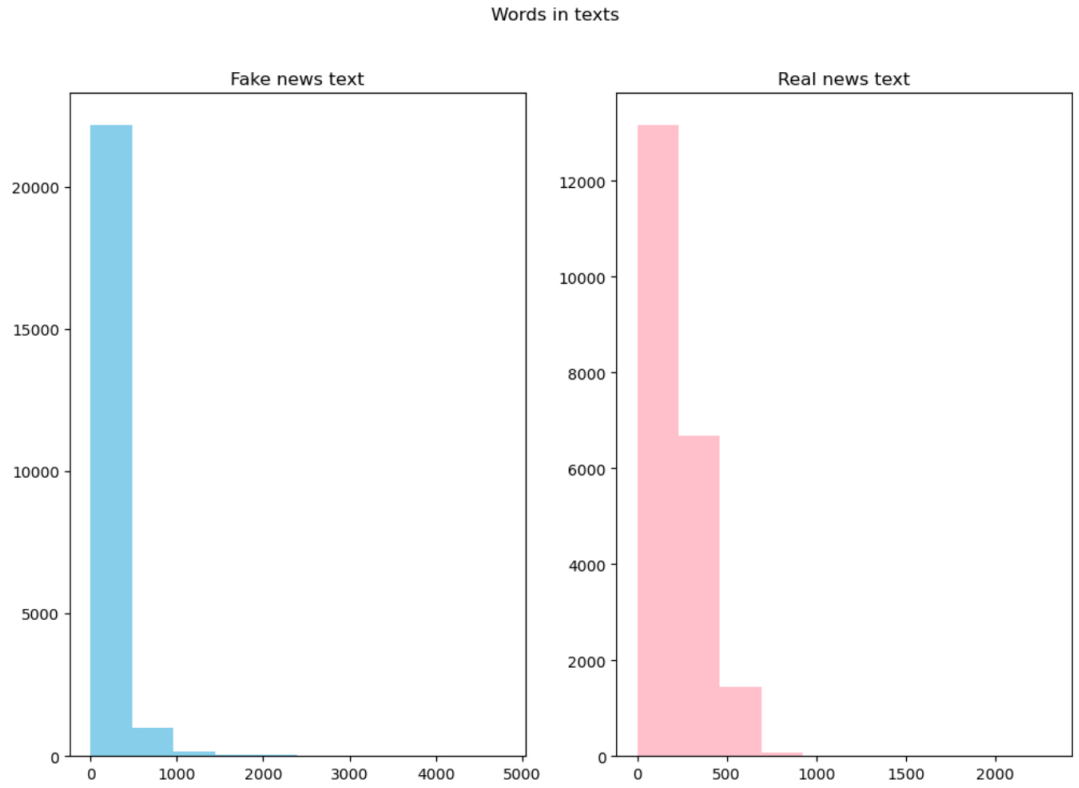
WordCloud for Fake News

In [24]:
```
plt.figure(figsize = (15,15))
wc = WordCloud(max_words = 500 , width = 1000 , height = 500 , stopwords = STOPWORDS).generate(" ".join(data[data.target == 0].te
plt.imshow(wc , interpolation = 'bilinear')
```

Out[24]: <matplotlib.figure.AxesImage at 0x203905865c0>



In [25]:
```
fig,(ax1,ax2)=plt.subplots(1,2,figsize=(12,8))
text_len=data[data['target']==0]['text'].str.split().map(lambda x: len(x))
ax1.hist(text_len,color='SkyBlue')
ax1.set_title('Fake news text')
text_len=data[data['target']==1]['text'].str.split().map(lambda x: len(x))
ax2.hist(text_len,color='pink')
ax2.set_title('Real news text')
fig.suptitle('Words in texts')
plt.show()
```



N-Gram Analysis

In [26]:
```
texts=' '.join(data['text'])
words=texts.split(" ")
```

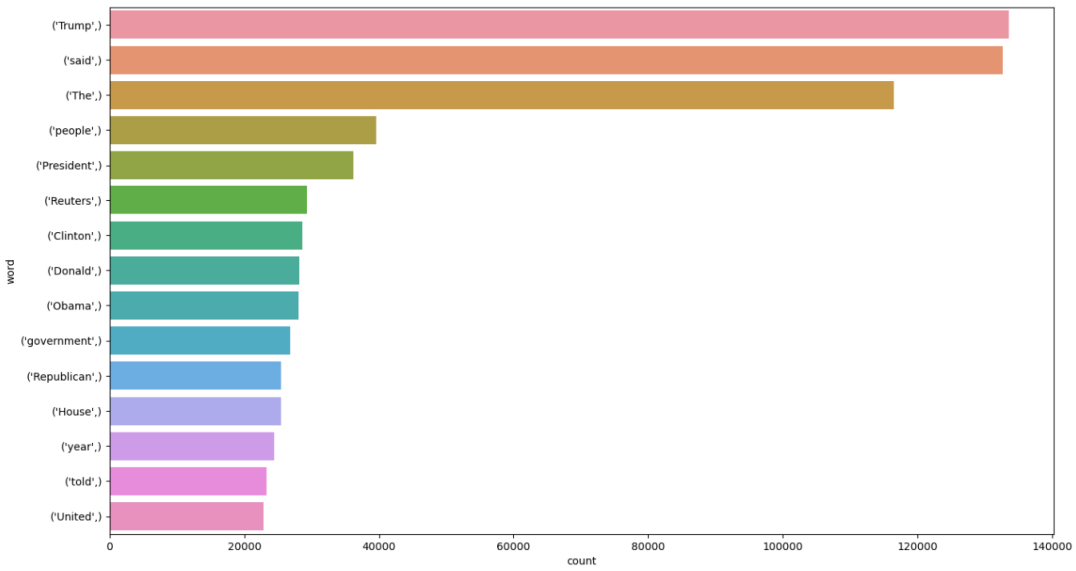
In [27]:
```
def draw_n_gram(words,i):
    n_gram=(pd.Series(nltk.ngrams(words,i)).value_counts())[:15]
    n_gram_df=pd.DataFrame(n_gram)
    n_gram_df=n_gram_df.reset_index()
    n_gram_df = n_gram_df.rename(columns={"index": "word", 0: "count"})
    print(n_gram_df.head())
    plt.figure(figsize = (16,9))
    return sns.barplot(x='count',y='word', data=n_gram_df)
```

Unigram Analysis

In [28]:
```
draw_n_gram(words,1)
```

```
          word   count
0     (Trump,)  133552
1      (said,)  132673
2       (The,)  116488
3     (people,)   39643
4  (President,)   36214
```

Out[28]: <Axes: xlabel='count', ylabel='word'>

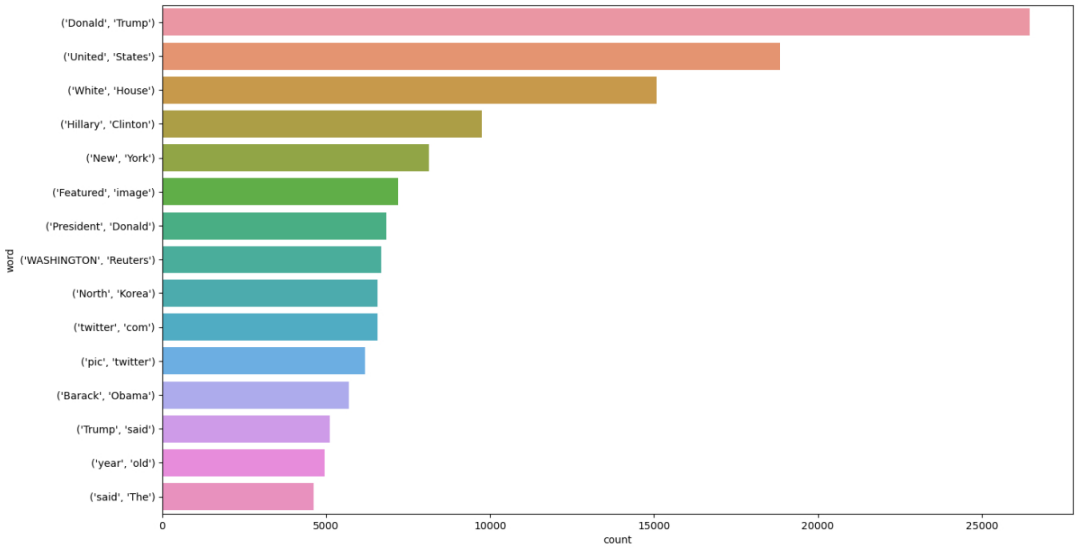


Bigram Analysis

In [29]:
```
draw_n_gram(words,2)
```

```
                word  count
0     (Donald, Trump)  26457
1    (United, States)  18841
2      (White, House)  15088
3  (Hillary, Clinton)   9750
4         (New, York)   8132
```

Out[29]: <Axes: xlabel='count', ylabel='word'>

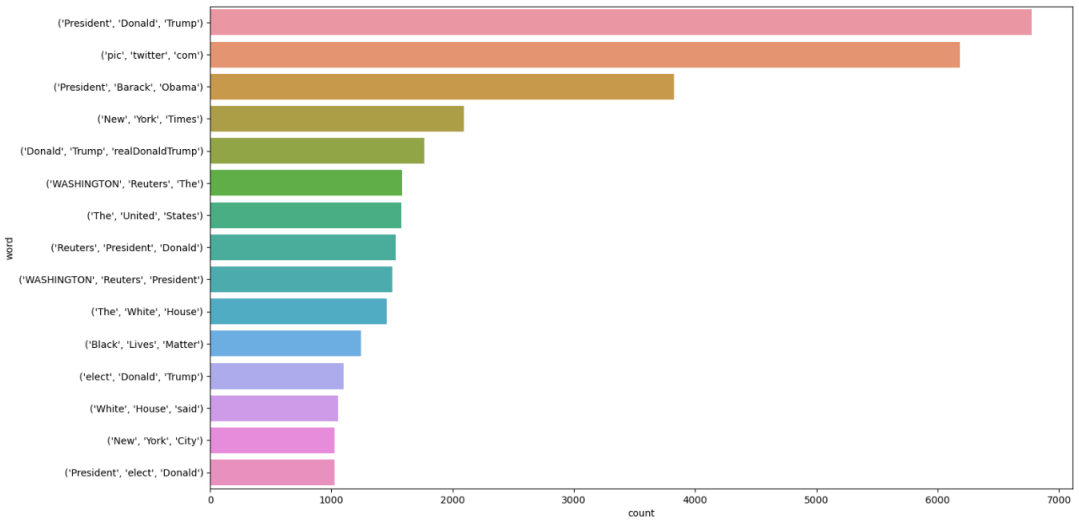


Trigram Analysis

In [30]:
```
draw_n_gram(words,3)
```

```
                              word  count
0        (President, Donald, Trump)   6776
1              (pic, twitter, com)   6185
2        (President, Barack, Obama)   3829
3                (New, York, Times)   2091
4  (Donald, Trump, realDonaldTrump)   1768
```

Out[30]: <Axes: xlabel='count', ylabel='word'>



Done with Preprocessing, so we need to split the data into train and test

In [31]:
```
X=data['text']
y=data['target']
X_train,X_test, y_train,y_test= train_test_split(X,y, test_size=0.20, random_state=1)
```

In [32]:
```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(35918,)
(8980,)
(35918,)
(8980,)
```

Feature Extraction with CountVectorizer and TfidfTransformer(Term Frequency-Inverse Document Frequency)

In [33]:
```
vectorizer=CountVectorizer()
transformer=TfidfTransformer()
```

For Train Data

In [34]:
```
X_train_vect=vectorizer.fit_transform(X_train)
X_train_tfidf=transformer.fit_transform(X_train_vect)
```

For Test Data

In [35]:
```
X_test_vect=vectorizer.transform(X_test)
X_test_tfidf=transformer.transform(X_test_vect)
```

Classification with Logistic Regression

In [36]:
```
model=LogisticRegression()
model.fit(X_train_tfidf,y_train)
y_pred=model.predict(X_test_tfidf)
acc=accuracy_score(y_pred,y_test)
print(acc)
```

```
0.9863028953229399
```

Training LSTM model(Long Short Term Memory)

In [37]:
```
# copnverting dependent variable into categorical variable
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

In [38]:
```
# converting to text to sequences
tokenizer=Tokenizer(20000,lower=True,oov_token='UNK')
tokenizer.fit_on_texts(X_train)
X_train = tokenizer.texts_to_sequences(X_train)
X_test = tokenizer.texts_to_sequences(X_test)
```

In [39]:
```
X_train = pad_sequences(X_train,maxlen=300,padding='post')
X_test = pad_sequences(X_test,maxlen=300,padding='post')
```

In [40]:
```
model_1 = Sequential()
```

```
model_1.add(Embedding(20000,64,input_length=300))
model_1.add(Dropout(0.5))
model_1.add(Bidirectional(LSTM(64,return_sequences=True)))
model_1.add(Bidirectional(LSTM(128)))
model_1.add(Dropout(0.3))
model_1.add(Dense(128))
model_1.add(Dense(2,activation="softmax"))

model_1.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

In [41]:
```
print(model_1.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 300, 64)           1280000

 dropout (Dropout)           (None, 300, 64)           0

 bidirectional (Bidirection  (None, 300, 128)          66048
 al)

 bidirectional_1 (Bidirecti  (None, 256)               263168
 onal)

 dropout_1 (Dropout)         (None, 256)               0

 dense (Dense)               (None, 128)               32896

 dense_1 (Dense)             (None, 2)                 258

=================================================================
Total params: 1642370 (6.27 MB)
Trainable params: 1642370 (6.27 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

In [42]:
```
history = model_1.fit(X_train, y_train,epochs=3, batch_size=512, shuffle=True, verbose =1)
```

```
Epoch 1/3
71/71 [==============================] - 2450s 35s/step - loss: 0.1918 - accuracy: 0.9068
Epoch 2/3
71/71 [==============================] - 4047s 57s/step - loss: 0.0101 - accuracy: 0.9968
Epoch 3/3
71/71 [==============================] - 4178s 59s/step - loss: 0.0022 - accuracy: 0.9993
```

In [43]:
```
model_1.evaluate(X_test,y_test)
```

```
281/281 [==============================] - 185s 653ms/step - loss: 0.0141 - accuracy: 0.9969
```

Out[43]:
```
[0.014149317517876625, 0.9968819618225098]
```

In [50]:
```
y_pred = model_1.predict(X_test)
y_pred
```

```
281/281 [==============================] - 158s 562ms/step
```

Out[50]:
```
array([[1.1057441e-09, 1.0000000e+00],
       [8.9882982e-01, 1.0117015e-01],
       [1.1286835e-11, 1.0000000e+00],
       ...,
       [1.0000000e+00, 3.9596124e-08],
       [9.9999893e-01, 1.0365569e-06],
       [9.9999988e-01, 1.7617661e-07]], dtype=float32)
```

In [ ]:
```

```