

# Text classification

## NLP Midterm

### Contents

#### 1. Introduction

- Problem area
- Objectives
- Dataset
- Evaluation methodology

#### 2. Implementation

- Pre-processing
- Baseline
- Classification methodology
- Programming style

#### 3. Outcome

- Performance
- Summary

## INTRODUCTION

### Problem area

**Classification** is the task of choosing the correct class label for a given input. In basic classification tasks, each input is considered in isolation from all other inputs, and the set of labels is defined in advance.

A simple definition for *Text classification* would be "the process of labeling text by its content". This is one of the most widely used NLP tasks, you may not realize it, but your e-mail spam filter is most likely an example of *text classification* in action. Other common types of text classification are:

- **Spam Classifier:** In the 90s, researchers had begun to apply machine-learning algorithms to automatically sort large amounts of text into categories. Today, it is still known as one of the earliest examples of automatic text classification and it has impacted our lives in many different ways.

- **Sentiment Analysis:** Aims to identify the polarity of a given text. From a computer's perspective, emotions are split into three categories positive, neutral, and negative with the corresponding scores of plus one, zero, and minus one. The keywords of each sentence will be categorized into one of the three categories and then the summation of the final score will determine the emotion.

Let's look at the sentence "the movie was great" the first step is to remove unnecessary words and identify the important ones, so the sentence will become "movie great" then, each word will be classified into one of the three categories. Depending on the trained data this will most likely result in a positive(1) for *great* and neutral(0) for *movie* which in return will have the sum of one (Positive).

This is a great tool to optimize customer search engines, social platforms and etc.

- **Fake News Detection** The individual and societal impacts of news media and social media has led to the invention of automated fake news detection, which aims to assess the truthfulness of stories. The goal of automatic fake news detection is not just to reduce the human time and effort that it takes to detect fake news, but to ultimately stop the spread of it.

The most common way to detect fake news is to categorize it as either "real" or "fake." However, such a binary system can be difficult to manage, since some fake news stories are partially real.

I chose to investigate Fake news because it relates to real-world violent events that threaten public safety.

## 2. IMPLEMENTATION

### Importing Libraries

```
In [1]: # Base Level imports for data science work

import numpy as np
import pandas as pd
import re,string,unicodedata
import os
from os import path

# Visualization Libs
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

# NLP Libs
import nltk
from sklearn.preprocessing import LabelBinarizer
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from wordcloud import WordCloud,STOPWORDS
```

```

from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize,sent_tokenize
from bs4 import BeautifulSoup
from keras.preprocessing import text, sequence
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelBinarizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC

# Additional Libs
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
from sklearn.model_selection import train_test_split
from string import punctuation
from nltk import pos_tag
from nltk.corpus import wordnet

# Deep Learning Libs
import keras
from keras.models import Sequential
from keras.layers import Dense,Embedding,LSTM,Dropout
from keras.callbacks import ReduceLROnPlateau
import tensorflow as tf

```

## Let's explore our data

In [2]:

```
# Import our data
#source: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset
real = pd.read_csv("Data/True.csv")
fake = pd.read_csv("Data/Fake.csv")
```

In [3]:

```
#Check if the dataset are ordered the same
(fake.shape,real.shape)
```

Out[3]:

```
((23481, 4), (21417, 4))
```

In [4]:

```
print("There are ", len(real), " real news in \"True.csv\".")
print("Here are its first five observations.")
real.head()
```

There are 21417 real news in "True.csv".  
 Here are its first five observations.

Out[4]:

	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 [5]: print("There are ", len(fake), " fake news in \"Fake.csv\".")
print("Here are its first five observations.")
fake.head()
```

There are 23481 fake news in "Fake.csv".  
 Here are its first five observations.

Out[5]:

	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 Milwaukee... Sheriff David Clarke has been accused of being a r...	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

I'll add a new column 'target' as a means of reference whether or not it is a real news or fake news then, join them.

In [6]:

```
# creating new category for whether they are true or fake.
real['target'] = 1
fake['target'] = 0
#Joining the 2 datasets
data = pd.concat([fake, real])
```

In [7]:

```
data.shape
```

Out[7]:

```
(44898, 5)
```

In [8]:

```
data.sample(5)
```

Out[8]:

		title	text	subject	date	target
17901		North Korean missiles will be able to reach U....	MOSCOW (Reuters) - North Korea's leadership ha...	worldnews	October 10, 2017	1
12738		MILLIONS OF FRAUDULENT AND INVALID VOTER REGIS...	This week, liberals have been repeating their ...	politics	Oct 14, 2016	0
5264		South Carolina Republican's town hall starts r...	NORTH CHARLESTON, S.C. (Reuters) - U.S. Senato...	politicsNews	February 25, 2017	1
18229		U.S. condemns arrest of Istanbul consulate worker	ANKARA (Reuters) - A U.S. consulate employee i...	worldnews	October 5, 2017	1
9386		Chicago immigrant claims U.S. retaliated over ...	CHICAGO (Reuters) - A Chicago activist sued U....	politicsNews	May 25, 2016	1

## Visualization of our dataset structure

In [9]:

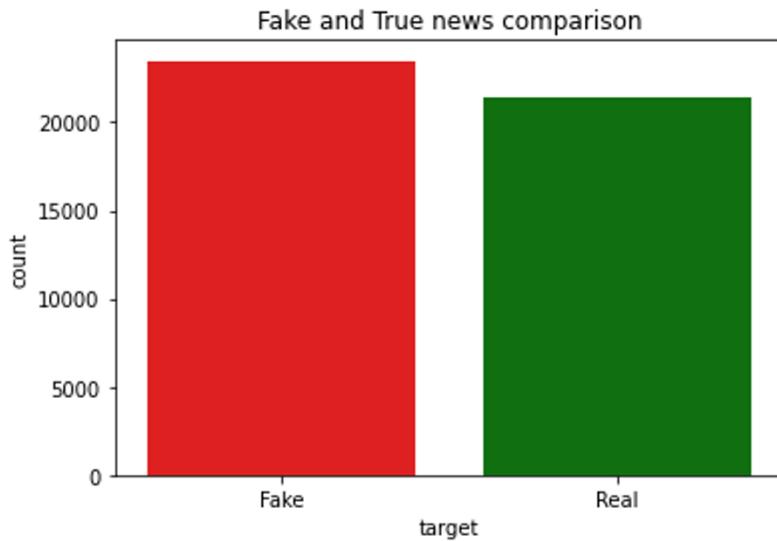
```
plt.title('Fake and True news comparison')
sns.countplot(x='target', data=data, palette=['red', 'green'])
plt.xticks([0, 1], ['Fake', 'Real'])

# ## or use this
# import plotly.graph_objects as go
# colors = ['gold', 'mediumturquoise']
# labels = ['Fake', 'Real']
# values = data['target'].value_counts()/data['target'].shape[0]

# fig = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.3)])
# fig.update_traces(hoverinfo='label+percent', textinfo='percent', textfont_size=20,
#                     marker=dict(colors=colors, line=dict(color='#000000', width=2)))
# fig.update_layout(
#     title_text="Target Balance",
#     title_font_color="white",
#     legend_title_font_color="yellow",
#     paper_bgcolor="black",
#     plot_bgcolor='black',
#     font_color="white",
# )
# fig.show()
# #source: https://www.kaggle.com/code/ohseokkim/fake-news-easy-nlp-text-classification
```

Out[9]:

```
([<matplotlib.axis.XTick at 0x204649a9550>,
 <matplotlib.axis.XTick at 0x204649a9520>],
 [Text(0, 0, 'Fake'), Text(1, 0, 'Real')])
```



We can see that the data are fairly balance with fake news being slightly bigger.

```
In [10]: print("There is a total of",len(real)+len(fake), 'news articles')
```

There is a total of 44898 news articles

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

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

Out of the 44898 news articles, we are fortunate to have no null values.

```
In [12]: print("There are",len(data.subject.value_counts()), "different subjects that are covered in this dataset")
```

There are 8 different subjects that are covered in this dataset

```
In [13]: pd.DataFrame(data.groupby(['subject','target'])['title'].count()).rename(columns = {'title': 'amount'})
```

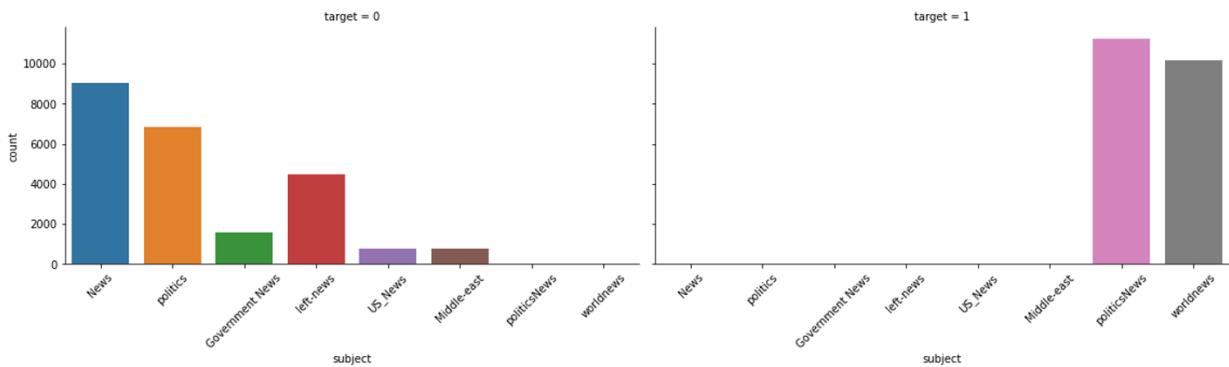
		amount
subject	target	
<b>Government News</b>	<b>0</b>	1570
<b>Middle-east</b>	<b>0</b>	778
<b>News</b>	<b>0</b>	9050
<b>US_News</b>	<b>0</b>	783
<b>left-news</b>	<b>0</b>	4459
<b>politics</b>	<b>0</b>	6841
<b>politicsNews</b>	<b>1</b>	11272
<b>worldnews</b>	<b>1</b>	10145

We notice that only 2 subjects out of 8 are real news.

```
In [14]: ### dont forget plot
g = sns.catplot(x="subject", col="target",
                 data=data, kind="count",
                 height=4, aspect=2)

#Rotating the xlabels
g.set_xticklabels(rotation=45)
```

Out[14]: <seaborn.axisgrid.FacetGrid at 0x2043a23f130>



## Converting the date columns to datetime format

Because the "date" column is not formated correctly, To demonstrate how news is transmitted in real time

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

```
Out[15]: December 20, 2017
182
December 6, 2017
166
November 30, 2017
162
November 9, 2017
158
October 13, 2017
155

...
August 6, 2016
1
https://100percentfedup.com/12-yr-old-black-conservative-whose-video-to-obama-went-viral-do-you-really-love-america-receives-death-threats-from-left/
1
December 9, 2017
1
https://100percentfedup.com/video-hillary-asked-about-trump-i-just-want-to-eat-some-pie/
1
September 3, 2016
1
Name: date, Length: 2397, dtype: int64
```

Let's remove the links that are in the data columns

```
In [16]: #Only keeps the which has months in it.  
data=data[data.date.str.contains("Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec")]
```

```
In [17]: print('Here we can see that all links has been removed')  
data['date'].value_counts()
```

```
Here we can see that all links has been removed  
Out[17]:  
December 20, 2017      182  
December 6, 2017       166  
November 30, 2017     162  
November 9, 2017      158  
October 13, 2017      155  
...  
December 4, 2017        1  
December 25, 2016       1  
October 9, 2017         1  
October 22, 2017        1  
Jun 21, 2015            1  
Name: date, Length: 2391, dtype: int64
```

Now that the links won't caused any trouble, we can convert our data with "pd.datetime"

```
In [18]: #Converting argument to datetime  
data = data.copy()  
data['date'] = pd.to_datetime(data['date'])
```

```
In [19]: print("We can confirm that the date columns has been formated")  
data.sample(5)
```

We can confirm that the date columns has been formated

```
Out[19]:
```

		title	text	subject	date	target
19564		TALENTLESS Gigi Hadid Makes Ridiculous "Apolog..."	Here's Gigi Hadid's ridiculous apology to Mel...	left-news	2016-11-23	0
8896		Paul Ryan Attacks President Obama For Being '...	On Monday, President Obama announced that he w...	News	2016-01-07	0
5836		Tech companies to meet on legal challenge to T...	SAN FRANCISCO (Reuters) - A group of technolog...	politicsNews	2017-01-31	1
4830		Maine Gov. LePage Says He's 'Tired Of Being C...	Maine Governor Paul LePage has made quite a na...	News	2016-08-31	0
10534		Michigan governor issues appeal over Flint fun...	(Reuters) - Michigan Governor Rick Snyder has ...	politicsNews	2016-03-03	1

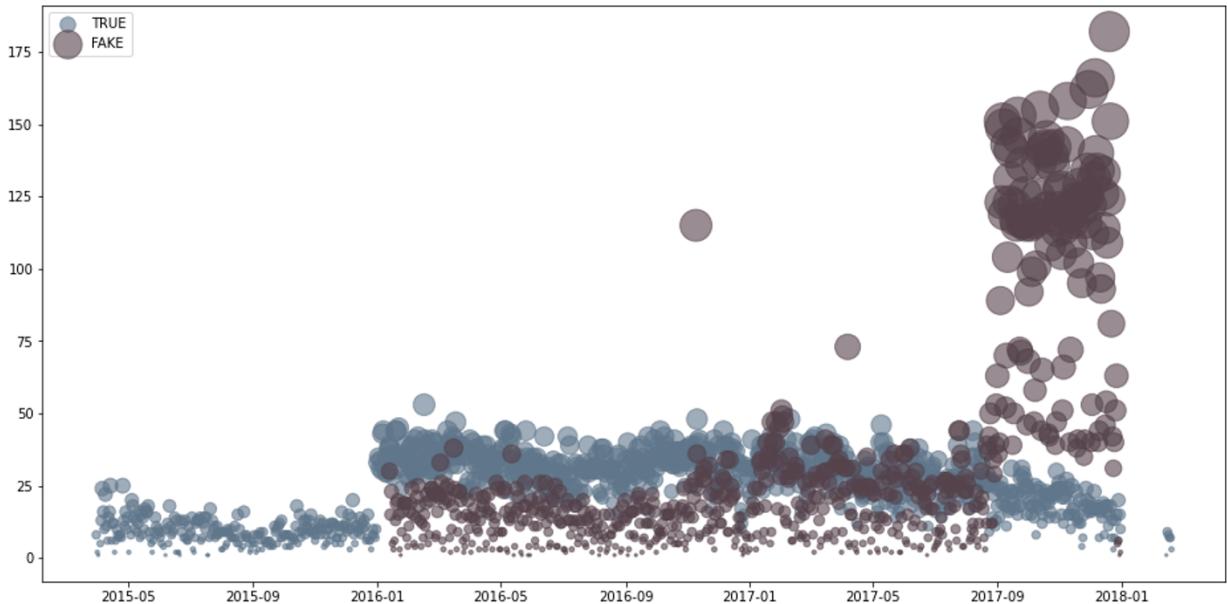
```
In [20]: subset_fake = data[data['target']==1].groupby('date').count().iloc[:, :1]  
subset_true = data[data['target']==0].groupby('date').count().iloc[:, :1]  
from pandas.plotting import register_matplotlib_converters  
  
register_matplotlib_converters()  
  
fig, ax = plt.subplots(figsize=(16,8))  
colours = ['#55424b', '#5f768b']  
sizes_true = subset_true.iloc[:, 0]*5  
sizes_fake = subset_fake.iloc[:, 0]*5  
plt.scatter(subset_true.index, subset_true.iloc[:, 0],  
           color = colours[1], label = "TRUE", alpha = 0.6,
```

```

    s = sizes_true)
plt.scatter(subset_fake.index, subset_fake.iloc[:, 0],
            color = colours[0], label = "FAKE", alpha = 0.6,
            s = sizes_fake)
plt.title("The quantity of news per day \n throughout 2015 - 2018 yy.\n")
plt.legend(loc = 'upper left')
plt.show()

```

The quantity of news per day  
throughout 2015 - 2018 yy.



## Cleaning

Let's combine the title and text of each news item into a single column.

```
In [21]: data['text'] = data['title'] + ' ' + data['text']
data.drop(['subject','title'], axis = 1, inplace = True)
data.sample(5)
```

		text	date	target
<b>12029</b>		Malaysian PM gets likely boost with electoral ...	2017-12-18	1
<b>2407</b>		Top Conservative Tries To Define 'Alt-Right' ...	2017-02-23	0
<b>3886</b>		Texas governor signs into law bill to punish '...	2017-05-08	1
<b>18112</b>		Mexico ex-first lady leaves opposition party f...	2017-10-06	1
<b>6397</b>		Internet Trolls Zimmerman, Top Bid Of \$65 Mil...	2016-05-13	0

## Lower Case

```
In [22]: data['text'] = data['text'].str.lower()
data.sample(5)
```

Out[22]:

		text	date	target
<b>6441</b>	exclusive: trump pulled out of project in ex-s...	2017-01-10	1	
<b>12047</b>	at least 61 dead after days of violence in eth...	2017-12-18	1	
<b>11004</b>	spineless gop proves majority means nothing ....s...	2017-05-01	0	
<b>4231</b>	north korean official says will continue missi...	2017-04-17	1	
<b>20125</b>	father of armed thug killed by milwaukee cop s...	2016-08-14	0	

## remove wrong confusing character

In [23]:

```
def clean_text(text):

    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 = re.sub('Reuters',' ',text)
    return text

data['text'] = data['text'].apply(lambda x:clean_text(x))
data.sample(5)
#source https://stackoverflow.com/questions/55187374/cleaning-text-with-python-and-re
```

Out[23]:

		text	date	target
<b>19214</b>	deadly aftershock volcanic ash spread alarm in...	2017-09-23	1	
<b>15576</b>	iran says saudi arabia facing results of its w...	2017-11-06	1	
<b>6</b>	fresh off the golf course trump lashes out at...	2017-12-23	0	
<b>23449</b>	boiler room – ep – cloppers or osmosis mk ult...	2016-02-11	0	
<b>5687</b>	ted cruz holds hearing on why obama won't use...	2016-06-28	0	

## Removing stop words:

**Stopwords:** A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. We would not want these words to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words. NLTK(Natural Language Toolkit) in python has a list of stopwords stored in 16 different languages.

Source: <https://www.geeksforgeeks.org/removing-stop-words-nltk-python/>

In [24]:

```
from nltk.corpus import stopwords
stop = stopwords.words('english')
data['text'] = data['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in stop]))
data.sample(5)
```

Out[24]:

		text	date	target
6771	fox news says nonwhites depicted separate equa...	2016-04-22	0	
5996	reason latino democrat voting trump make rofl ...	2016-06-06	0	
485	trump tried distance 'senior advisor' promised...	2017-08-28	0	
7577	watch insider reveals trump university scammed...	2016-03-10	0	
5372	watch nicolle wallace declares republican part...	2016-07-22	0	

## Lemmatizing words:

Lemmatization was performed, so only root words were retained.

```
In [25]: wnl = WordNetLemmatizer()
data["text"] = data["text"].apply(lambda x: nltk.word_tokenize(x))
data["text"] = data["text"].apply(lambda x: " ".join([wnl.lemmatize(word) for word in
data.head()])
```

Out[25]:

		text	date	target
0	donald trump sends embarrassing new year ' s e...	2017-12-31	0	
1	drunk bragging trump staffer started russian c...	2017-12-31	0	
2	sheriff david clarke becomes internet joke thr...	2017-12-30	0	
3	trump obsessed even obama ' s name coded websi...	2017-12-29	0	
4	pope francis called donald trump christmas spe...	2017-12-25	0	

## Deeper analysis:

### Wordcloud

The size of each word represents its frequency or relevance in a **wordcloud**, which is a data visualisation tool for visualising text data.

```
In [26]: plt.figure(figsize = (20,20))

true_news = " ".join(data[data["target"]==1]["text"])

d = path.dirname(__file__) if "__file__" in locals() else os.getcwd()
# read the mask / color image taken from
coloring = np.array(Image.open(path.join(d, "Data/real.jpg")))
stopwords = set(STOPWORDS)

wc = WordCloud(background_color="white", max_words = 2000 , width = 1600 , height = 800
               stopwords=stopwords, max_font_size=80, random_state=42)

# generate word cloud
wc.generate(true_news)
```

```

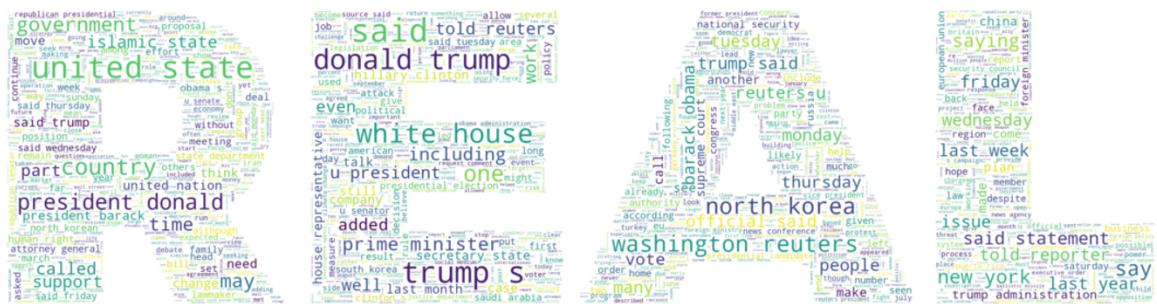
# create coloring from image
image_colors = ImageColorGenerator(coloring)

plt.imshow(wc, interpolation='bilinear')
plt.axis("off")

# # This bit of code is used to create a high quality exported image
# fig1 = plt.gcf()
# plt.show()
# plt.draw()
# fig1.savefig("real_news_nlp_word_cloud.png", bbox_inches='tight', dpi=600)

```

Out[26]: (-0.5, 3029.5, 1090.5, -0.5)



In [27]: plt.figure(figsize = (20,20))

```

fake_news = " ".join(data[data["target"]==0]["text"])

d = path.dirname(__file__) if "__file__" in locals() else os.getcwd()
# read the mask / color image taken from
coloring = np.array(Image.open(path.join(d, "Data/fake.jpg")))
stopwords = set(STOPWORDS)

wc = WordCloud(background_color="white", max_words = 2000 , width = 1600 , height = 800,
               stopwords=stopwords, max_font_size=80, random_state=42)

# generate word cloud
wc.generate(fake_news)

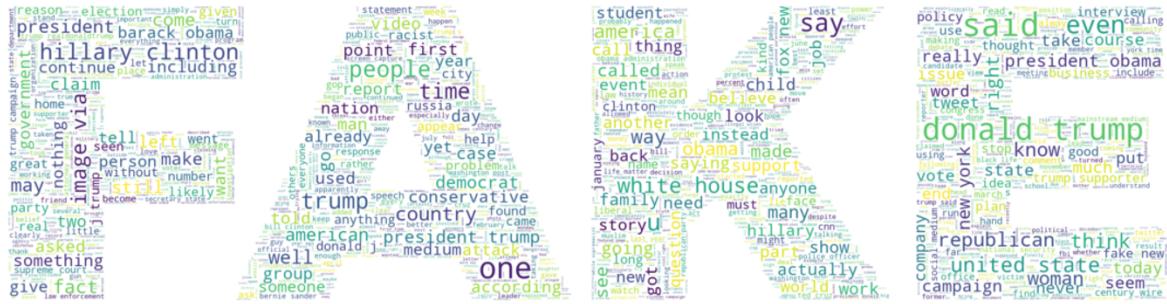
# create coloring from image
image_colors = ImageColorGenerator(coloring)

plt.imshow(wc, interpolation='bilinear')
plt.axis("off")

# # This bit of code is used to create a high quality exported image
# fig1 = plt.gcf()
# plt.show()
# plt.draw()
# fig1.savefig("real_news_nlp_word_cloud.png", bbox_inches='tight', dpi=600)

```

Out[27]: (-0.5, 3029.5, 1090.5, -0.5)



Because majority of the articles were classified as political, there are similar representations such as 'Donald Trump' and 'United States.'

## N-Grams

```
In [28]: def get_top_text_ngrams(corpus, n, g):
    vec = CountVectorizer(ngram_range=(g, g)).fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
    words_freq = sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:n]

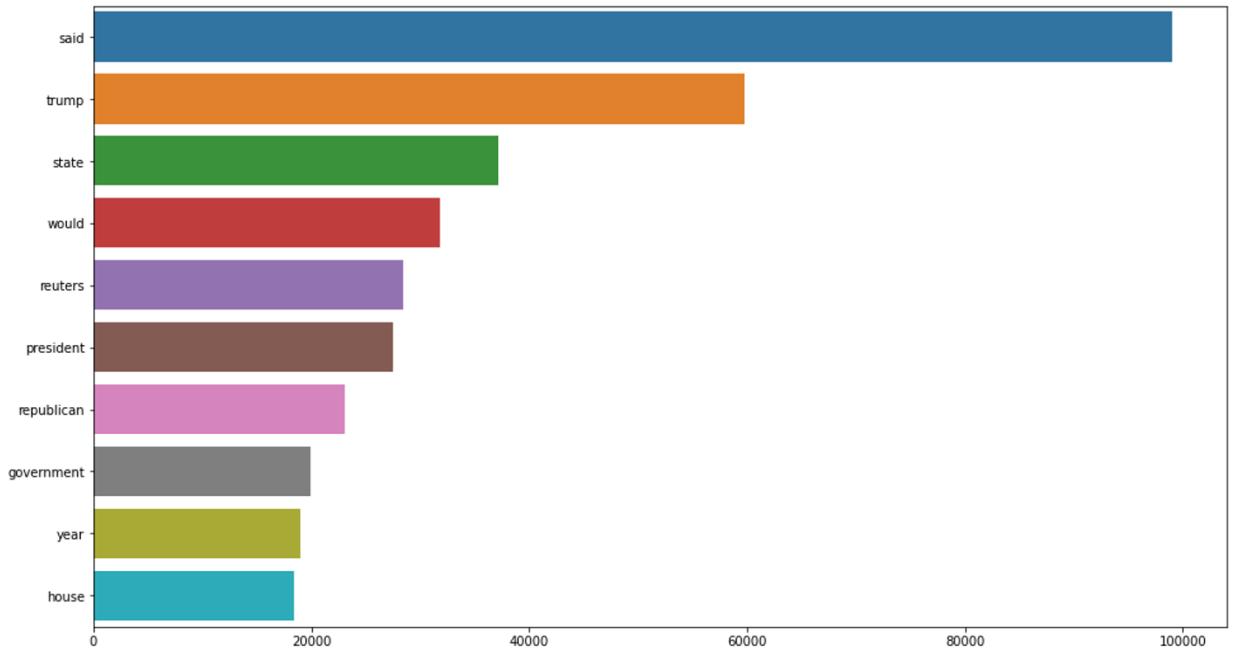
#source: https://www.guruguru.science/competitions/13/discussions/c73cb36a-6b8f-4a13-8
```

### 1. Unigram Analysis

True news Unigram

```
In [29]: plt.figure(figsize = (16,9))
True_ngram = data[data["target"]==1]["text"]
top_10_uni = get_top_text_ngrams(True_ngram,10,1)
top_10_uni = dict(top_10_uni)
sns.barplot(x=list(top_10_uni.values()),y=list(top_10_uni.keys()))
```

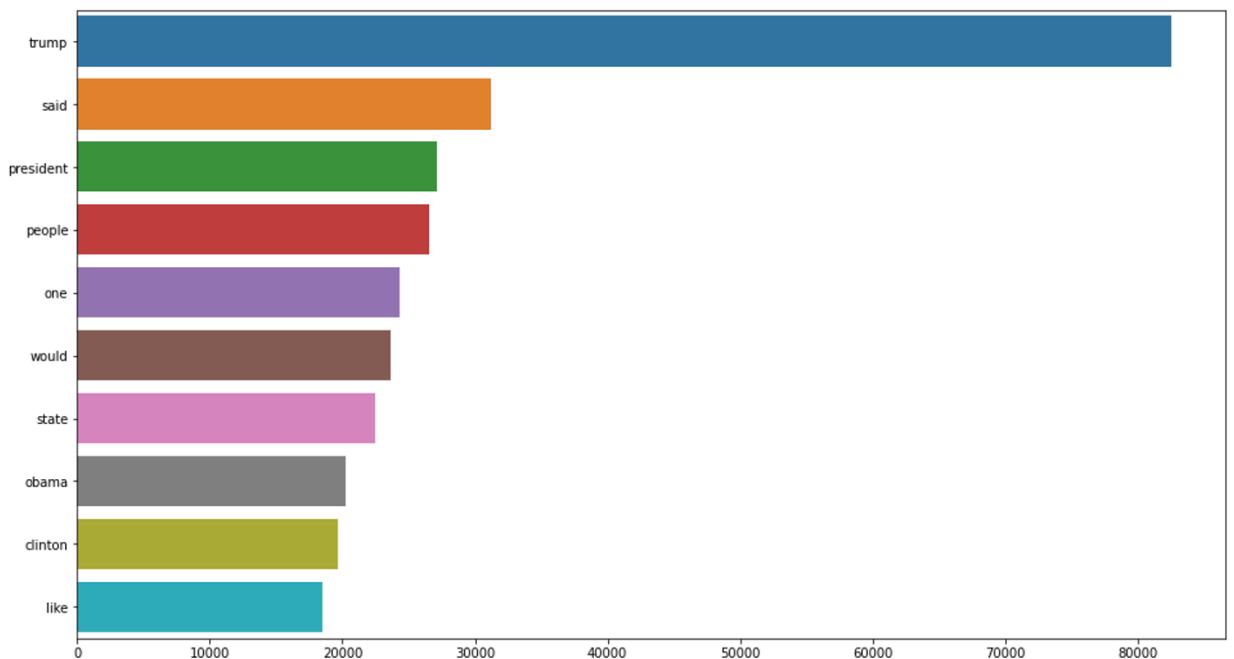
Out[29]: <AxesSubplot:>



Fake news Unigram

```
In [30]: plt.figure(figsize = (16,9))
Fake_ngram = data[data["target"]==0]["text"]
top_10_uni = get_top_text_ngrams(Fake_ngram,10,1)
top_10_uni = dict(top_10_uni)
sns.barplot(x=list(top_10_uni.values()),y=list(top_10_uni.keys()))
```

Out[30]: <AxesSubplot:>



## 2. Bigram Analysis

True news Bigram

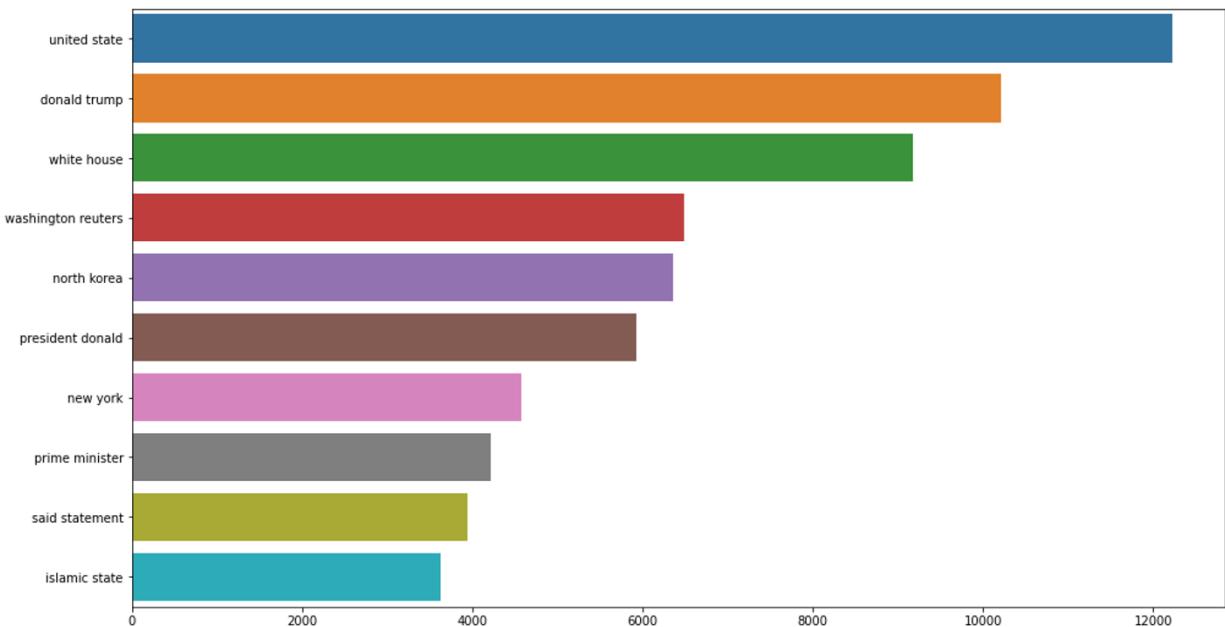
```
In [31]: plt.figure(figsize = (16,9))
```

```

True_ngram = data[data["target"]==1]["text"]
top_10_bi = get_top_text_ngrams(True_ngram,10,2)
top_10_bi = dict(top_10_bi)
sns.barplot(x=list(top_10_bi.values()),y=list(top_10_bi.keys()))

```

Out[31]: <AxesSubplot:>



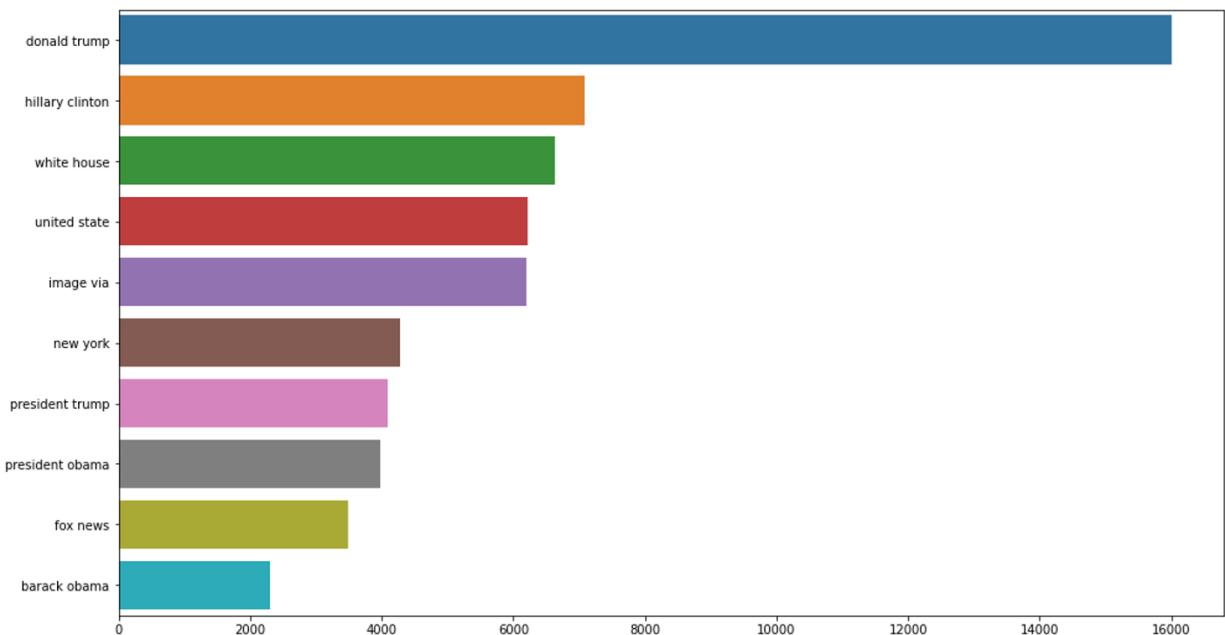
Fake news Bigram

```

In [32]: plt.figure(figsize = (16,9))
Fake_ngram = data[data["target"]==0]["text"]
top_10_bi = get_top_text_ngrams(Fake_ngram,10,2)
top_10_bi = dict(top_10_bi)
sns.barplot(x=list(top_10_bi.values()),y=list(top_10_bi.keys()))

```

Out[32]: <AxesSubplot:>

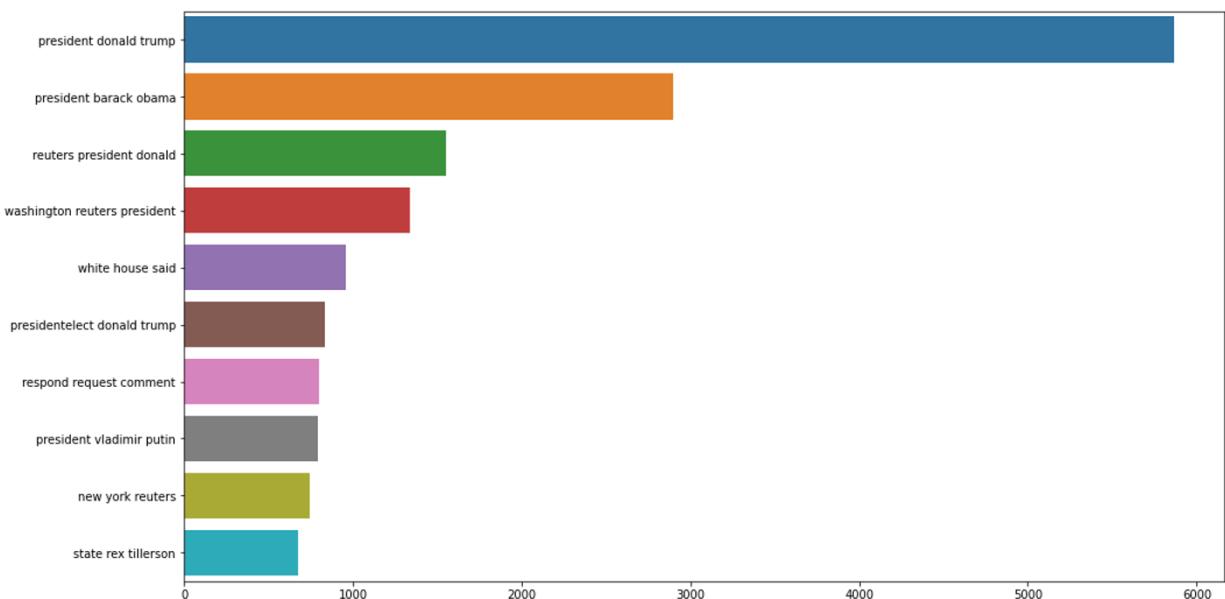


### 3. Trigram Analysis

### True news Trigram

```
In [33]: plt.figure(figsize = (16,9))
True_ngram = data[data["target"]==1]["text"]
top_10_tri = get_top_text_ngrams(True_ngram,10,3)
top_10_tri = dict(top_10_tri)
sns.barplot(x=list(top_10_tri.values()),y=list(top_10_tri.keys()))
```

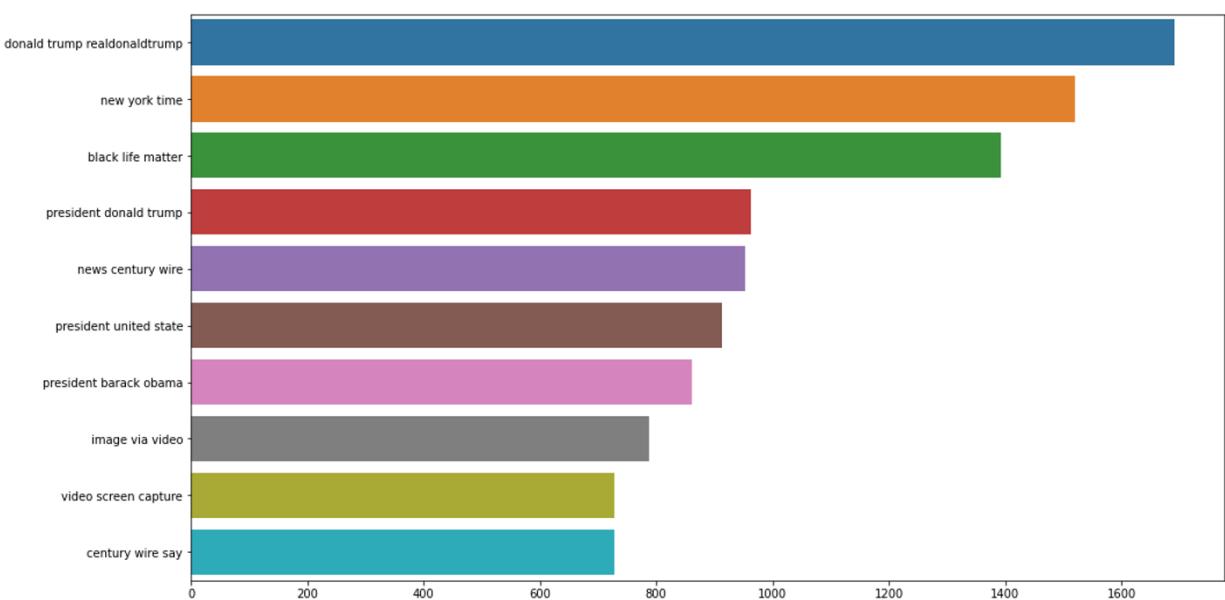
Out[33]: <AxesSubplot:>



### Fake news Trigram

```
In [34]: plt.figure(figsize = (16,9))
Fake_ngram = data[data["target"]==0]["text"]
top_10_tri = get_top_text_ngrams(Fake_ngram,10,3)
top_10_tri = dict(top_10_tri)
sns.barplot(x=list(top_10_tri.values()),y=list(top_10_tri.keys()))
```

Out[34]: <AxesSubplot:>



## Naive Bayes

Splitting dataset into training and test sets.

```
In [35]: y = data['target']
x = data['text']
X_train, X_test, y_train, y_test = train_test_split(x, y)
```

```
In [36]: count_vectorizer = CountVectorizer(stop_words='english')
count_train = count_vectorizer.fit_transform(X_train.values)
count_test = count_vectorizer.transform(X_test.values)
print(count_train.shape)

(33666, 166380)
```

```
In [37]: nb_classifier1 = MultinomialNB()
nb_classifier1.fit(count_train, y_train)

pred1 = nb_classifier1.predict(count_test)

print(classification_report(y_test, pred1, target_names = ['Fake', 'True']))
```

	precision	recall	f1-score	support
Fake	0.96	0.95	0.95	5791
True	0.95	0.96	0.95	5431
accuracy			0.95	11222
macro avg	0.95	0.95	0.95	11222
weighted avg	0.95	0.95	0.95	11222

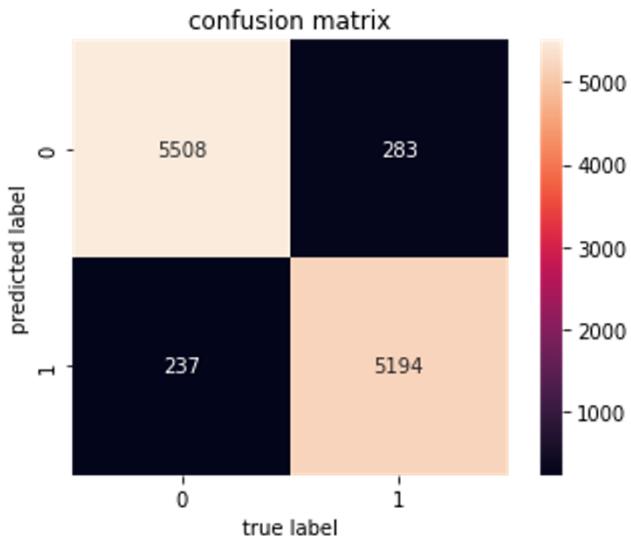
We end up with a pretty good score (95%) on this dataset, but of course there is always room for improvement.

## CONCLUSION

### Evaluation

#### Visualize the performance using a confusion matrix

```
In [38]: mat = confusion_matrix(y_test, pred1)
sns.heatmap(mat, square=True, annot=True, fmt='d')
plt.title('confusion matrix')
plt.xlabel('true label')
plt.ylabel('predicted label');
```



This is pretty helpful since it shows us where our model struggles. For both classes mispredictions is approximately 4%:

237 fake news predicted as true

283 real news predicted as fake

## Summary and conclusions

So we ended up with a fairly accurate (95%) model that shows how some techniques can produce decent results without much tuning. I hope to improve it further in the future, as we talked briefly about misinformation and how society is very sensitive to information online, so sometimes misinforming people may create actions with bad consequences.