REAL AND FACK NEWS DATA SET

Abstract:

Title: A Comparative Analysis of Real and Fake News Detection in Online Media

Module 1: Data Collection and Preprocessing

In Module 1 of our research project, we focus on the critical task of collecting and preprocessing a comprehensive dataset of news articles. This module encompasses the following key steps:

- Data Gathering: We employ web scraping techniques to acquire news articles from diverse online sources, including reputable news websites and social media platforms. This dataset comprises both real and potentially fake news articles to ensure a balanced representation.

- Data Cleaning: To enhance the quality of the collected data, we perform extensive data cleaning, including text normalization, removal of duplicates, and extraction of relevant features. Additionally, we assess and handle missing data points.

- Labeling: Human annotators are engaged to label the collected news articles as either "real" or "fake" based on factual accuracy and credibility. This ground truth labeling is crucial for supervised machine learning models.

Module 2: Feature Engineering and Selection

Module 2 focuses on feature engineering and selection techniques to extract meaningful information from the preprocessed dataset. The steps involved in this module are as follows:

- Text Representation: We explore various text representation methods such as TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings (e.g., Word2Vec, GloVe) to convert textual data into numerical features.

- Feature Extraction: We extract linguistic, semantic, and contextual features from the text, including n-grams, sentiment analysis scores, and topic modeling. These features are designed to capture patterns and nuances in news articles.

- Feature Selection: Employing feature selection algorithms, we identify and retain the most relevant features while eliminating noise. This helps improve the model's efficiency and interpretability.

Module 3: Machine Learning Models for Classification

Module 3 is dedicated to building and evaluating machine learning models for the classification of news articles into real and fake categories. This module encompasses the following steps:

- Model Selection: We experiment with a range of classification algorithms, including logistic

regression, random forests, and deep neural networks. Each model's performance is assessed using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.

- Model Training and Tuning: The selected models are trained on the preprocessed dataset, and hyperparameter tuning is performed to optimize their performance.

- Cross-Validation: To ensure the generalizability of our models, we employ cross-validation techniques to assess their robustness and reliability.

Module 4: Evaluation and Conclusion

In the final module, we evaluate the effectiveness of our real and fake news detection models using an independent test dataset. We analyze the results, discuss insights, and draw conclusions regarding the performance and limitations of the models.

Overall, our research project aims to provide a comprehensive framework for real and fake news detection, from data collection to model evaluation. Through this systematic approach, we contribute to the ongoing efforts to combat misinformation and enhance the credibility of online news sources.

INTRODUCTION

l identifying fake and real news data using nltk.

IMPORT LIBRARIES

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import nltk
from tqdm import tqdm
import torchtext.data as data
import torch.optim as optim
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from torchtext.data import get_tokenizer
import seaborn as sns
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import re
```

in [2]:

```
# CONFIG
TRUE_DATA_PATH = '/kaggle/input/fake-and-real-news-dataset/True.csv'
FALSE_DATA_PATH = '/kaggle/input/fake-and-real-news-dataset/Fake.csv'
```

LOAD DATA

In [3]:

```
true_df = pd.read_csv(TRUE_DATA_PATH)
false_df = pd.read_csv(FALSE_DATA_PATH)
```

In [4]:

```
true_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
dtypes: object(4)
memory usage: 669.4+ KB
```

In [5]:

```
false_df.info():
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB
```

In [6]:

```
true_df['category'] = np.ones(len(true_df), dtype=int)
false_df['category'] = np.zeros(len(false_df), dtype=int)

true_df.head()
```

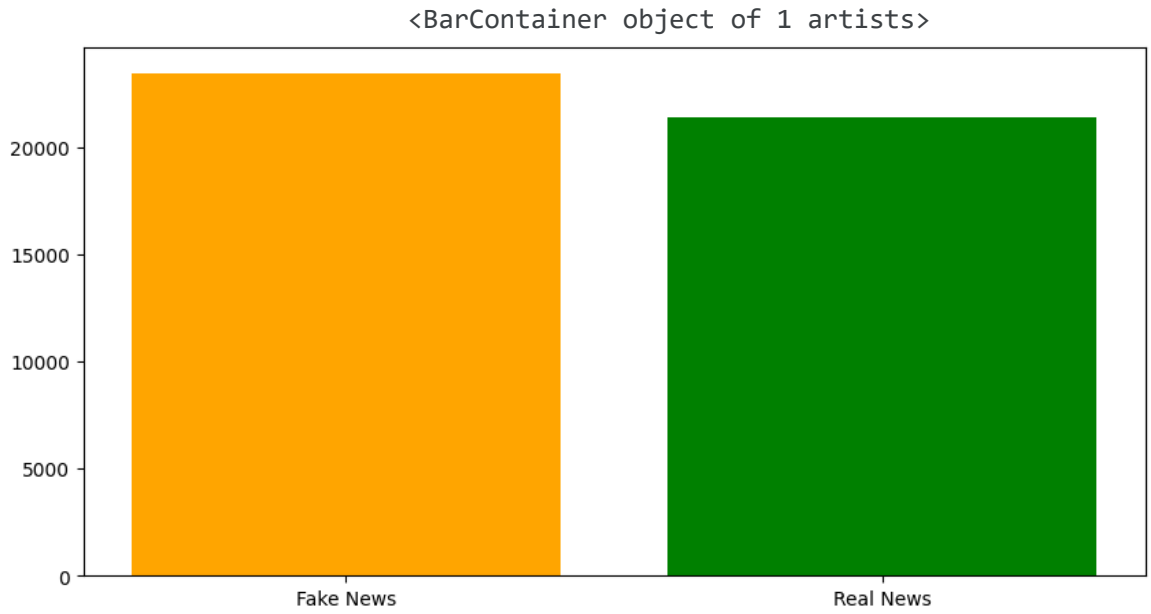
out[6]:

s.no	title	subject	subject	date	category
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

In [7]:

```
plt.figure(figsize=(10, 5))
plt.bar('Fake News', len(false_df), color='orange')
plt.bar('Real News', len(true_df), color='green')
```

out[7]:



In [8]:

```
# Difference of the Fake and Real News
print(f'Difference between Fake and Real News: {len(false_df) -
len(true_df)}')
```

```
Difference between Fake and Real News: 2064
```

In [9]:

```
# concat = merging datasets
news_df = pd.concat([true_df, false_df], axis=0)
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 0 to 23480
Data columns (total 5 columns):
 #   Column    Non-Null Count  Data type
---  ------    --------------  -----
 0   title     44898 non-null  object
 1   text      44898 non-null  object
 2   subject   44898 non-null  object
 3   date      44898 non-null  object
 4   category  44898 non-null  int64
Data types: int64(1), object(4)
memory usage: 2.1+ MB
```

In [10]:

```
news_df = news_df.sample(frac=1)
news_df.head(5)
```

out[10]:

	title	text	subject	date	category
880	These Charts Show Why We're All Screwed Under...	During his presidential campaign, Donald Trump...	News	July 11, 2017	0
597	U.S. towns, cities fear taxpayer revolt if Rep...	WASHINGTON (Reuters) - From Pataskala, Ohio, t...	politicsNews	November 17, 2017	1
15813	JEB BUSH WANTS CONGRESS TO APPROVE AMNESTY And...	Jeb Bush just unofficially placed himself on t...	politics	Apr 17, 2015	0
15407	DEMOCRAT PLAN TO INFILTRATE TRADITIONALLY RED ...	The fundamental transformation of America El S...	politics	Jul 27, 2015	0
18289	NEW YORK TIMES REFUSES To Publish Op-Ed By Lif...	The NYT allegedly wouldn t run Alan Dershowitz...	left-news	Jul 20, 2017	0

In [11]:

```
news_df['subject'].value_counts()
```

out[11]:

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

in[12]:

```
news_df = pd.get_dummies(news_df, columns=['subject'])
news_df.head()
```

in[13]:

```
news_df = news_df.drop('date', axis=1)
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 880 to 7431
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   title                   44898 non-null  object
 1   text                    44898 non-null  object
 2   category                44898 non-null  int64
 3   subject_Government News  44898 non-null  bool
 4   subject_Middle-east     44898 non-null  bool
 5   subject_News            44898 non-null  bool
 6   subject_US_News         44898 non-null  bool
 7   subject_left-news       44898 non-null  bool
 8   subject_politics        44898 non-null  bool
 9   subject_politicsNews    44898 non-null  bool
 10  subject_worldnews       44898 non-null  bool
dtypes: bool(8), int64(1), object(2)
memory usage: 1.7+ MB
```

in[14]

```
import nltk
import subprocess
import nltk
import subprocess

try:
    nltk.data.find('wordnet.zip')
except:
    nltk.download( download_dir='wornet')
    command = copora wornet’
    subprocess.run(command.split())
    nltk.data.path.append('working')

from nltk.corpus import wordnet
```

In[15]:

```
from nltk.corpus import wordnet

new_text = []
pattern = "[^a-zA-Z]"

lemma = nltk.WordNetLemmatizer()

for txt in tqdm(news_df.text):

    txt = re.sub(pattern," ",txt)
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt)
    txt = [lemma.lemmatize(word) for word in txt]
    txt = " ".join(txt)
    new_text.append(txt)

new_text[0]
```

```
100%|██████████| 44898/44898 [05:21<00:00, 139.84it/s]
```

```
Out[15]:
'during his presidential campaign donald trump constantly made reference to
repealing and replacing the disaster that is obamacare and democrat
collectively shuddered we all knew that nothing good could come of this now
after six month in office despite discovering that nobody knew healthcare
could be so difficult president trump is about to deliver on his campaign
promise a the senate return from a one week recess to get back to the task at
hand trying to come to an agreement on their new healthcare bill known a the
better care reconciliation act bcra one that they have predominately kept the
public in the dark about thing are looking bleak however a even the republican
party remains divided on a bill that is not only going to raise out of pocket
cost and restrict access to service for many but also cause ten of million to
lose their health insurance completely over the coming decade these chart
might be able to put the entire healthcare debacle into perspective we all
know the short term medicaid cut are going to suck the new health care bill
will save a ton of money but roughly a quarter of those saving or
approximately billion over the next decade come from cut to medicaid the
result is that million le 'during his presidential campaign donald trump
constantly made reference to repealing and replacing the disaster that is
obamacare and democrat collectively shuddered we all knew that nothing good
could come of this now after six month in office despite discovering that
nobody knew healthcare could be so difficult president trump is about to
deliver on his campaign promise a the senate return from a one week recess to
get back to the task at hand trying to come to an agreement on their new
healthcare bill known a the better care reconciliation act bcra one that they
have predominately kept the public in the dark about thing are looking bleak
however a even the republican party remains divided on a bill that is not only
going to raise out of pocket cost and restrict access to service for many but
also cause ten of million to lose their health insurance completely over the
coming decade these chart might be able to put the entire healthcare debacle
into perspective we all know the short term medicaid cut are going to suck the
new health care bill will save a ton of money but roughly a quarter of those
saving or approximately billion over the next decade come from cut to medicaid
the result is that million le including those whose overwhelming majority
voted for trump yes the new bill will drive up the uninsured rate by at least
and even up to in every state by a new study by the urban institute found the
older and poorer you are the more you will be paying for insurance premium if
an analysis by the center for budget and policy priority is to be believed
health insurance premium are going to go through the roof but those hit the
```

```
worst will be older american the older middle class will be hit pretty hard
too a their tax credit will go through the floor the center for budget and
policy priority analysis also found that the tax credit that are available to
help older people in the individual market afford health insurance are going
to do just the opposite and plummet even employer plan aren t immune the gop s
new bill cut to medicaid and individual market subsidy have given the million
american that receive their health insurance through their employer a false
sense of security but they re not safe either not only will the new
legislation bring back annual and lifetime limit in employer plan a well a end
penalty for company that don t provide health insurance to their worker but it
will also allow employer to shift much of the cost of copays deductible and
coinsurance onto their worker the center for american progress calculated how
many will feel the crunch hospital are going to feel the crunch a well
hospital aren t happy with the new bill and it is easy to see why when you
consider it will cause a large spike in uncompensated care for hospital across
all state finally the new bill will cause massive job loss particularly in the
health care sector by more than million job will be lost a a direct result of
the bcra go by the result of a report by the commonwealth fund and george
washington university in fact the report go a far a to say that every state
except hawaii would have fewer job and a weaker economy however it s not just
health care employment that will be affected but also retail and construction
a well so if you thought this latest rewrite of the gop s health care
legislation didn t affect you you more than likely thought wrongly even if it
isn t your health care that is directly affected chance are you will still
feel the ripple effect of the bill on the economy both on a state and national
level featured image via drew angerer getty image'.

In[16]:
new_title = []
for txt in tqdm(news_df.title):

    txt = re.sub(pattern," ",txt)
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt)
    txt = [lemma.lemmatize(word) for word in txt]
    txt = " ".join(txt)
    new_title.append(txt)
new_title[0]

100%|██████████| 44898/44898 [00:15<00:00, 2941.90it/s]

Out [16]:
'these chart show why we re all screwed under the gop health care bill'

In[17]:
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_title = CountVectorizer(stop_words="english",max_features=1000)
vectorizer_text = CountVectorizer(stop_words="english",max_features=4000)

title_matrix = vectorizer_title.fit_transform(new_title).toarray()
text_matrix = vectorizer_text.fit_transform(new_text).toarray()

print("Finished")

Finished

In[18]
news_df.head(5)

in[19]:
news_df.drop(['title', 'text'], axis=1, inplace=True)
```

```
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 880 to 7431
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   category                44898 non-null  int64
 1   subject_Government News  44898 non-null  bool
 2   subject_Middle-east     44898 non-null  bool
 3   subject_News            44898 non-null  bool
 4   subject_US_News         44898 non-null  bool
 5   subject_left-news       44898 non-null  bool
 6   subject_politics        44898 non-null  bool
 7   subject_politicsNews    44898 non-null  bool
 8   subject_worldnews       44898 non-null  bool
dtypes: bool(8), int64(1)
memory usage: 1.0 MB
```

```
in [20]:
```

```
print(news_df.shape)
print(title_matrix.shape)
print(text_matrix.shape)
```

```
(44898, 9)
(44898, 1000)
(44898, 4000)
```

```
In[21]:
```

```
X = np.concatenate((np.array(news_df.drop('category', axis=1)), title_matrix,
                    text_matrix), axis=1)

y = news_df.category
```

```
(44898, 5008)
(44898,)
```

```
in [23]:
```

```
X_train, X_test, y_train, y_test = train_test_split(X, np.array(y),
                                                    test_size=0.25,
                                                    random_state=42)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(33673, 5008)
(11225, 5008)
(33673,)
(11225,)
```

BUILDING MODEL

```
In[24]:
```

INTRODUCTION

Hello and welcome to my first NLP project, identifying fake and real news data using pytorch and nltk.

IMPORT LIBRARIES

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import nltk
from tqdm import tqdm
import torchtext.data as data
import torch.optim as optim
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from torchtext.data import get_tokenizer
import seaborn as sns
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import re
```

In [2]:

```
# CONFIG
TRUE_DATA_PATH = 'input fake-and-real-news-dataset/True.csv'
FALSE_DATA_PATH = 'output fake-and-real-news-dataset/True.csv'
```

LOAD DATA

In [3]:

```
true_df = pd.read_csv(TRUE_DATA_PATH)
false_df = pd.read_csv(FALSE_DATA_PATH)
```

In [4]:

```
true_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
```

```
dtypes: object(4)
memory usage: 669.4+ KB
```

In [5]:

```
false_df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB
```

In [6]:

```
true_df['category'] = np.ones(len(true_df), dtype=int)
false_df['category'] = np.zeros(len(false_df), dtype=int)

true_df.head()
```

Out[6]:

	Title	text	subject	date	category
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

In [7]:

```
plt.figure(figsize=(10, 5))
plt.bar('Fake News', len(false_df), color='orange')
plt.bar('Real News', len(true_df), color='green')
```

Out[7]:

```
<BarContainer object of 1 artists>
```

In [8]:

```
# Difference of the Fake and Real News
print(f'Difference between Fake and Real News: {len(false_df) - len(true_df)}')
```

```
Difference between Fake and Real News: 2064
```

In [9]:

```
# concat = merging datasets
news_df = pd.concat([true_df, false_df], axis=0)
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 0 to 23480
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   title     44898 non-null  object
 1   text      44898 non-null  object
 2   subject   44898 non-null  object
 3   date      44898 non-null  object
 4   category  44898 non-null  int64 
dtypes: int64(1), object(4)
memory usage: 2.1+ MB
```

In [10]:

```
news_df = news_df.sample(frac=1)
news_df.head(5)
```

Out[10]:

	Title	text	subject	date	category
880	These Charts Show Why We’re All Screwed Under...	During his presidential campaign, Donald Trump...	News	July 11, 2017	0
597	U.S. towns, cities fear taxpayer revolt if Rep...	WASHINGTON (Reuters) - From Pataskala, Ohio, t...	politicsNews	November 17, 2017	1
15813	JEB BUSH WANTS CONGRESS TO APPROVE AMNESTY And...	Jeb Bush just unofficially placed himself on t...	politics	Apr 17, 2015	0
15407	DEMOCRAT PLAN TO INFILTRATE TRADITIONALLY RED ...	The fundamental transformation of America El S...	politics	Jul 27, 2015	0
18289	NEW YORK TIMES REFUSES To Publish Op-Ed By Lif...	The NYT allegedly wouldn t run Alan Dershowitz...	left-news	Jul 20, 2017	0

In [11]:

```
news_df['subject'].value_counts()
```

Out[11]:

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

In [12]:

```
news_df = pd.get_dummies(news_df, columns=['subject'])
news_df.head()
```

Out[12]:

	title	text	Date	category	subject_Government News	subject_Middle-east	subject_News	subject_US_News	subject_left-news	subject_politics	subject_politicsNews	subject_worldnews
880	These Charts Show Why We're All Screwed Under ...	During his presidential campaign, Donald Trump...	July 11, 2017	0	False	False	True	False	False	False	False	False
597	U.S. towns, cities fear taxpayer revolt if Rep...	WASHINGTON (Reuters) - From Pataskala, Ohio, t...	November 17, 2017	1	False	False	False	False	False	False	True	False
15813	JEB BUSH WANTS	Jeb Bush just unoffi	Apr 17, 201	0	False	False	False	False	False	True	False	False

	CONGRESS TO APPROVE AMNESTY And...	cially placed himself on t...	5									
15407	DEMOCRAT PLAN TO INFILTRATE TRADITIONALLY RED ...	The fundamental transformation of America El S...	Jul 27, 2015	0	False	False	False	False	False	True	False	False
18289	NEW YORK TIMES REFUSES To Publish Op-Ed By Lif...	The NYT allegedly wouldn t run Alan Dershowitz. ..	Jul 20, 2017	0	False	False	False	False	True	False	False	False

In [13]:

```
news_df = news_df.drop('date', axis=1)
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 880 to 7431
Data columns (total 11 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   title                   44898 non-null  object
 1   text                    44898 non-null  object
 2   category                44898 non-null  int64
 3   subject_Government News  44898 non-null  bool
 4   subject_Middle-east     44898 non-null  bool
 5   subject_News            44898 non-null  bool
 6   subject_US_News         44898 non-null  bool
 7   subject_left-news       44898 non-null  bool
 8   subject_politics        44898 non-null  bool
 9   subject_politicsNews    44898 non-null  bool
 10  subject_worldnews       44898 non-null  bool
dtypes: bool(8), int64(1), object(2)
memory usage: 1.7+ MB
```

In [14]:

```
import nltk
import subprocess

# Download and unzip wordnet
try:
    nltk.data.find('wordnet.zip')
except:
    nltk.download('wordnet', download_dir='fake news')
    command =
    subprocess.run(command.split())
    nltk.data.path.append('data')

# Now you can import the NLTK resources as usual
from nltk.corpus import wordnet
[nltk_data]
Archive:  working corpora wordnet.zip
```


```
In [15]:

from nltk.corpus import wordnet

new_text = []
pattern = "[^a-zA-Z]"

lemma = nltk.WordNetLemmatizer()

for txt in tqdm(news_df.text):

    txt = re.sub(pattern,"news ",txt)
    txt = txt.lower()
    txt = nltk.word_tokenize(txt)
    txt = [lemma.lemmatize(word) for word in txt]
    txt = " ".join(txt)
    new_text.append(txt)

new_text[0]

100%|██████████| 44898/44898 [05:21<00:00, 139.84it/s]

Out[15]:

'during his presidential campaign donald trump constantly made reference to
repealing and replacing the disaster that is obamacare and democrat
collectively shuddered we all knew that nothing good could come of this now
after six month in office despite discovering that nobody knew healthcare
could be so difficult president trump is about to deliver on his campaign
promise a the senate return from a one week recess to get back to the task at
hand trying to come to an agreement on their new healthcare bill known a the
better care reconciliation act bcra one that they have predominately kept the
public in the dark about thing are looking bleak however a even the republican
party remains divided on a bill that is not only going to raise out of pocket
cost and restrict access to service for many but also cause ten of million to
lose their health insurance completely over the coming decade these chart
might be able to put the entire healthcare debacle into perspective we all
know the short term medicaid cut are going to suck the new health care bill
will save a ton of money but roughly a quarter of those saving or
approximately billion over the next decade come from cut to medicaid the
```

result is that million le people will be enrolled in medicaid under the new gop bill than compared to obamacare if you thought that wa bad the long term effect are even worse the inflation rate for medicaid spending beginning in is much slower affecting those who rely on it the most mainly child the disabled and the elderly in fact by federal medicaid spending on child will be reduced by almost a third and by a quarter for the disabled and the elderly when compared to the current law according to an analysis by the health consulting firm avalere health the percentage of those uninsured will rise in every single age bracket that s right under the bcra million people will lose their insurance compared to million under the version passed by the house and every single age group will be affected according to an assessment by the congressional budget office it will also rise in every state including those whose overwhelming majority voted for trump yes the new bill will drive up the uninsured rate by at least and even up to in every state by a new study by the urban institute found the older and poorer you are the more you will be paying for insurance premium if an analysis by the center for budget and policy priority is to be believed health insurance premium are going to go through the roof but those hit the worst will be older american the older middle class will be hit pretty hard too a their tax credit will go through the floor the center for budget and policy priority analysis also found that the tax credit that are available to help older people in the individual market afford health insurance are going to do just the opposite and plummet even employer plan aren t immune the gop s new bill cut to medicaid and individual market subsidy have given the million american that receive their health insurance through their employer a false sense of security but they re not safe either not only will the new legislation bring back annual and lifetime limit in employer plan a well a end penalty for company that don t provide health insurance to their worker but it will also allow employer to shift much of the cost of copays deductible and coinsurance onto their worker the center for american progress calculated how many will feel the crunch hospital are going to feel the crunch a well hospital aren t happy with the new bill and it is easy to see why when you consider it will cause a large spike in uncompensated care for hospital across all state finally the new bill will cause massive job loss particularly in the health care sector by more than million job will be lost a a direct result of the bcra go by the result of a report by the commonwealth fund and george washington university in fact the report go a far a to say that every state except hawaii would have fewer job and a weaker economy however it s not just health care employment that will be affected but also retail and construction a well so if you thought this latest rewrite of the gop s health care legislation didn t affect you you more than likely thought wrongly even if it isn t your health care that is directly affected chance are you will still feel the ripple effect of the bill on the economy both on a state and national level featured image via drew angerer getty image'

In [16]:

```
new_title = []
for txt in tqdm(news_df.title):

    txt = re.sub(pattern," ",txt) # Cleaning
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt) # Tokenizing
    txt = [lemma.lemmatize(word) for word in txt] # Lemmatizing
    txt = " ".join(txt)
    new_title.append(txt)
new_title[0]
```

100%|██████████| 44898/44898 [00:15<00:00, 2941.90it/s]

Out[16]:

'these chart show why we re all screwed under the gop health care bill'

In [17]:

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_title = CountVectorizer(stop_words="english",max_features=1000)
vectorizer_text = CountVectorizer(stop_words="english",max_features=4000)

title_matrix = vectorizer_title.fit_transform(new_title).toarray()
text_matrix = vectorizer_text.fit_transform(new_text).toarray()

print("Finished")
```

```
Finished
```

In [18]:

```
news_df.head(5)
```

Out[18]:

	title	text	Category	subject_Government News	subject_Middle-east	subject_News	subject_US_News	subject_left-news	subject_politics	subject_politicsNews	subject_worldnews
880	These Charts Show Why We're All Screwed Under...	During his presidential campaign, Donald Trump ...	0	False	False	True	False	False	False	False	False
597	U.S. towns, cities fear taxpayer revolt if Rep...	WASHINGTON (Reuters) - From Pataskala, Ohio, t...	1	False	False	False	False	False	False	True	False
15813	JEB BUSH WANTS CONGRESS TO APPROVE AMNESTY	Jeb Bush just unofficially placed himself on t...	0	False	False	False	False	False	True	False	False

	And...										
15407	DEMOCRAT PLAN TO INFILTRATE TRADITIONALLY RED ...	The fundamental transformation of America El S...	0	False	False	False	False	False	True	False	False
18289	NEW YORK TIMES REFUSES To Publish Op-Ed By Lif...	The NYT allegedly would n t run Alan Dershowitz.. .	0	False	False	False	False	True	False	False	False

In [19]:

```
news_df.drop(['title', 'text'], axis=1, inplace=True)
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 880 to 7431
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   category                 44898 non-null  int64
 1   subject_Government News  44898 non-null  bool
 2   subject_Middle-east      44898 non-null  bool
 3   subject_News             44898 non-null  bool
 4   subject_US_News          44898 non-null  bool
 5   subject_left-news        44898 non-null  bool
 6   subject_politics         44898 non-null  bool
 7   subject_politicsNews     44898 non-null  bool
 8   subject_worldnews        44898 non-null  bool
dtypes: bool(8), int64(1)
memory usage: 1.0 MB
```

In [20]:

```
print(news_df.shape)
print(title_matrix.shape)
print(text_matrix.shape)
```

```
(44898, 9)
```

```
(44898, 1000)
(44898, 4000)
```

In [21]:

```
X = np.concatenate((np.array(news_df.drop('category', axis=1)), title_matrix,
                    text_matrix), axis=1)

y = news_df.category
```

In [22]:

```
print(X.shape)
print(y.shape)
```

```
(44898, 5008)
(44898,)
```

In [23]:

```
X_train, X_test, y_train, y_test = train_test_split(X, np.array(y),
                                                    test_size=0.25,
                                                    random_state=42)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(33673, 5008)
(11225, 5008)
(33673,)
(11225,)
```

BUILDING MODEL

In [24]:

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class NewsClassifier(nn.Module):
    def __init__(self):
        super(NewsClassifier, self).__init__()
        self.linear1 = nn.Linear(5008, 2000)
        self.relu1 = nn.ReLU()
        self.linear2 = nn.Linear(2000, 500)
        self.relu2 = nn.ReLU()
        self.linear3 = nn.Linear(500, 100)
        self.relu3 = nn.ReLU()
        self.dropout = nn.Dropout(0.1)
```

```
        self.linear4 = nn.Linear(100, 20)
        self.relu4 = nn.ReLU()
        self.linear5 = nn.Linear(20, 2)

    def forward(self, x):
        out = self.linear1(x)
        out = self.relu1(out)
        out = self.linear2(out)
        out = self.relu2(out)
        out = self.linear3(out)
        out = self.relu3(out)
        out = self.dropout(out)
        out = self.linear4(out)
        out = self.relu4(out)
        out = self.linear5(out)

        return out
```

In [25]:

```
model = NewsClassifier()
optimizer = torch.optim.Adam(model.parameters(), lr=0.012)
criterion = nn.CrossEntropyLoss()
```

In [26]:

```
import torch
from tqdm import tqdm

X_train = torch.Tensor(X_train)
y_train = torch.Tensor(y_train).type(torch.LongTensor)

X_test = torch.Tensor(X_test)
y_test = torch.Tensor(y_test).type(torch.LongTensor)

EPOCHS = 30

for epoch in tqdm(range(EPOCHS)):
    optimizer.zero_grad()

    # Forward pass
    outputs = model(X_train)

    # Calculate loss
    loss = criterion(outputs, y_train)
    loss.backward()
    optimizer.step()

    # Calculate accuracy
    _, predicted = torch.max(outputs, 1)
    correct = (predicted == y_train).sum().item()
    accuracy = correct / len(y_train) * 100.0

    print(f'Epoch [{epoch+1}/{EPOCHS}], Loss: {loss.item():.4f}, Accuracy:
{accuracy:.2f}%')
```

```
  3%|▎         | 1/30 [00:12<05:59, 12.41s/it]
Epoch [1/30], Loss: 0.6984, Accuracy: 47.69%
  7%|▋         | 2/30 [00:33<08:17, 17.79s/it]
Epoch [2/30], Loss: 11.2522, Accuracy: 52.31%
 10%|█         | 3/30 [00:46<06:56, 15.43s/it]
Epoch [3/30], Loss: 2.9311, Accuracy: 52.22%
 13%|█▎        | 4/30 [00:58<06:09, 14.21s/it]
Epoch [4/30], Loss: 1.1631, Accuracy: 52.31%
 17%|█▋        | 5/30 [01:12<05:45, 13.81s/it]
```

```
Epoch [5/30], Loss: 2.0339, Accuracy: 48.20%
 20%|██        | 6/30 [01:24<05:20, 13.36s/it]
Epoch [6/30], Loss: 1.0975, Accuracy: 47.75%
 23%|██        | 7/30 [01:37<05:05, 13.29s/it]
Epoch [7/30], Loss: 0.6547, Accuracy: 48.29%
 27%|███       | 8/30 [01:49<04:45, 12.97s/it]
Epoch [8/30], Loss: 0.5906, Accuracy: 65.16%
 30%|███       | 9/30 [02:02<04:28, 12.80s/it]
Epoch [9/30], Loss: 0.5159, Accuracy: 85.44%
 33%|███       | 10/30 [02:15<04:18, 12.91s/it]
Epoch [10/30], Loss: 0.4104, Accuracy: 88.48%
 37%|███       | 11/30 [02:27<04:02, 12.76s/it]
Epoch [11/30], Loss: 0.2363, Accuracy: 94.74%
 40%|████      | 12/30 [02:41<03:53, 12.98s/it]
Epoch [12/30], Loss: 0.2008, Accuracy: 94.57%
 43%|████      | 13/30 [02:53<03:37, 12.81s/it]
Epoch [13/30], Loss: 0.1622, Accuracy: 95.46%
 47%|████      | 14/30 [03:06<03:23, 12.72s/it]
Epoch [14/30], Loss: 0.1197, Accuracy: 96.86%
 50%|█████     | 15/30 [03:19<03:14, 12.96s/it]
Epoch [15/30], Loss: 0.1070, Accuracy: 97.40%
 53%|█████     | 16/30 [03:32<02:59, 12.84s/it]
Epoch [16/30], Loss: 0.0793, Accuracy: 98.13%
 57%|█████     | 17/30 [03:45<02:48, 12.93s/it]
Epoch [17/30], Loss: 0.0550, Accuracy: 98.66%
 60%|██████    | 18/30 [03:58<02:34, 12.85s/it]
Epoch [18/30], Loss: 0.0440, Accuracy: 98.82%
 63%|██████    | 19/30 [04:10<02:20, 12.76s/it]
Epoch [19/30], Loss: 0.0373, Accuracy: 98.99%
 67%|██████    | 20/30 [04:24<02:09, 12.94s/it]
Epoch [20/30], Loss: 0.0289, Accuracy: 99.16%
 70%|███████   | 21/30 [04:36<01:55, 12.85s/it]
Epoch [21/30], Loss: 0.0243, Accuracy: 99.39%
 73%|███████   | 22/30 [04:49<01:43, 12.95s/it]
Epoch [22/30], Loss: 0.0211, Accuracy: 99.46%
 77%|███████   | 23/30 [05:02<01:29, 12.80s/it]
Epoch [23/30], Loss: 0.0153, Accuracy: 99.62%
 80%|████████  | 24/30 [05:14<01:16, 12.73s/it]
Epoch [24/30], Loss: 0.0105, Accuracy: 99.74%
 83%|████████  | 25/30 [05:28<01:04, 12.90s/it]
Epoch [25/30], Loss: 0.0073, Accuracy: 99.82%
 87%|████████  | 26/30 [05:40<00:51, 12.78s/it]
Epoch [26/30], Loss: 0.0055, Accuracy: 99.85%
 90%|█████████ | 27/30 [05:54<00:38, 12.99s/it]
Epoch [27/30], Loss: 0.0039, Accuracy: 99.89%
 93%|█████████ | 28/30 [06:06<00:25, 12.86s/it]
Epoch [28/30], Loss: 0.0024, Accuracy: 99.94%
 97%|█████████| 29/30 [06:19<00:12, 12.83s/it]
Epoch [29/30], Loss: 0.0016, Accuracy: 99.96%
100%|██████████| 30/30 [06:33<00:00, 13.11s/it]
Epoch [30/30], Loss: 0.0011, Accuracy: 99.97%
```

Evaluating

In [27]:

```
model.eval()
with torch.no_grad():
    test_outputs = model(X_test)
    _, predicted = torch.max(test_outputs, 1)
    correct = (predicted == y_test).sum().item()
    test_accuracy = correct / len(y_test) * 100.0
    test_loss = criterion(test_outputs, y_test)

print(f'Test Accuracy: {test_accuracy:.2f}%')
print(f'Test Loss: {test_loss:.2f}%')
```

```
Test Accuracy: 99.21%
Test Loss: 0.04%
```