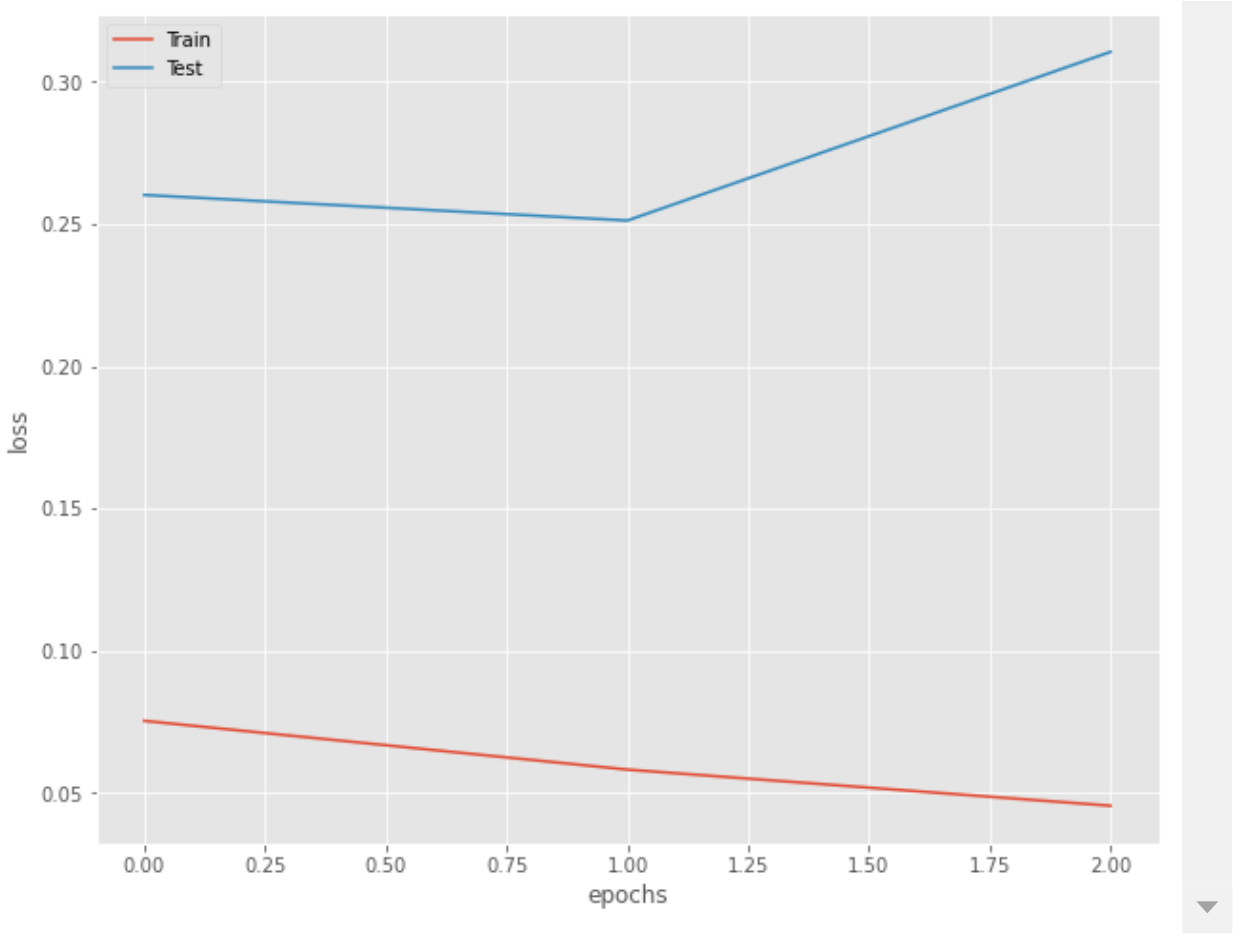


Bert Model

In [141]:
```
import pandas as pd
import numpy as np
```

In [150]:
```
true_path  = pd.read_csv("C:\\Users\\Laptop inn\\Desktop\Fake.csv")
fake_path  = pd.read_csv("C:\\Users\\Laptop inn\\Desktop\True.csv")
```

In [153]:
```
true_df = true_path  = pd.read_csv("C:\\Users\\Laptop inn\\Desktop\Fake.csv")
fake_df = fake_path  = pd.read_csv("C:\\Users\\Laptop inn\\Desktop\True.csv")
```

In [154]:
```
display(true_df)
```

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

23481 rows × 4 columns

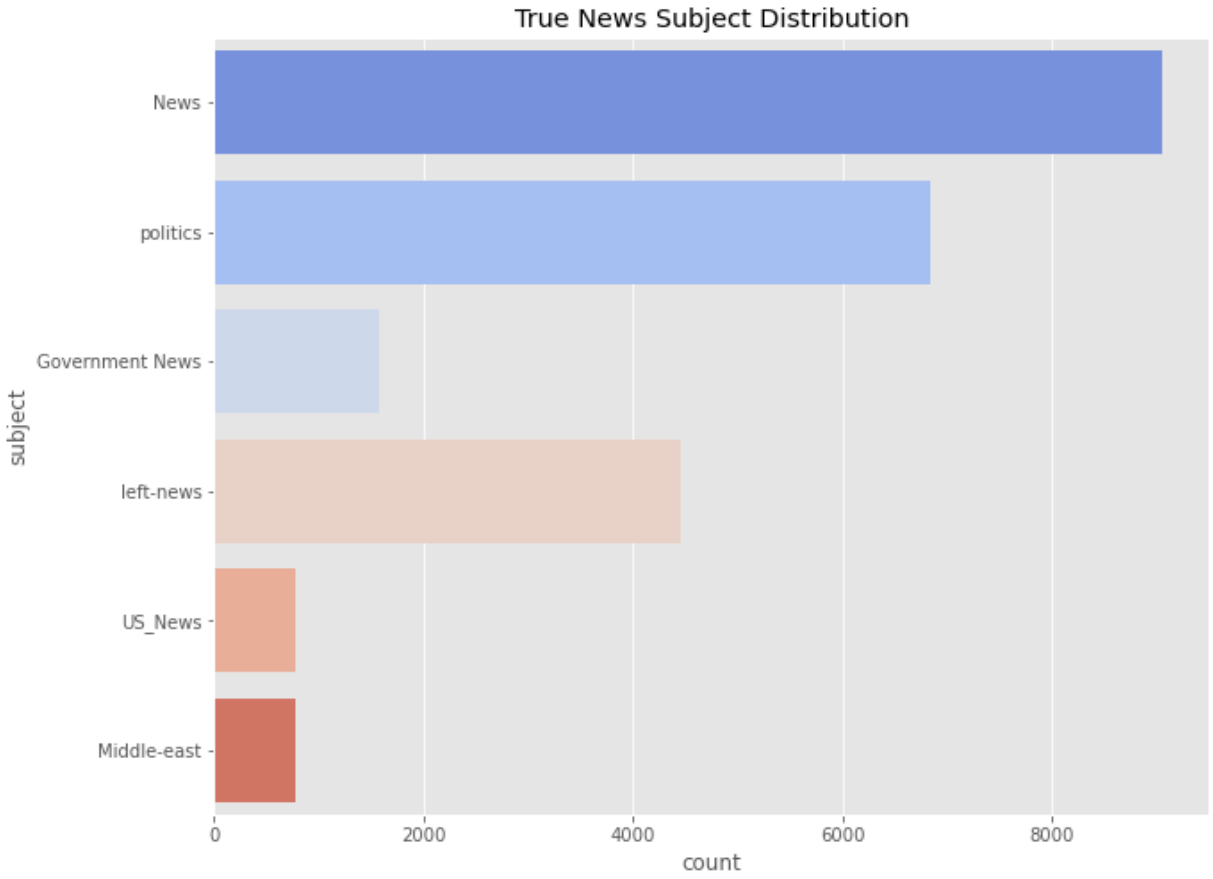
In [155]:
```
display(fake_df)
```

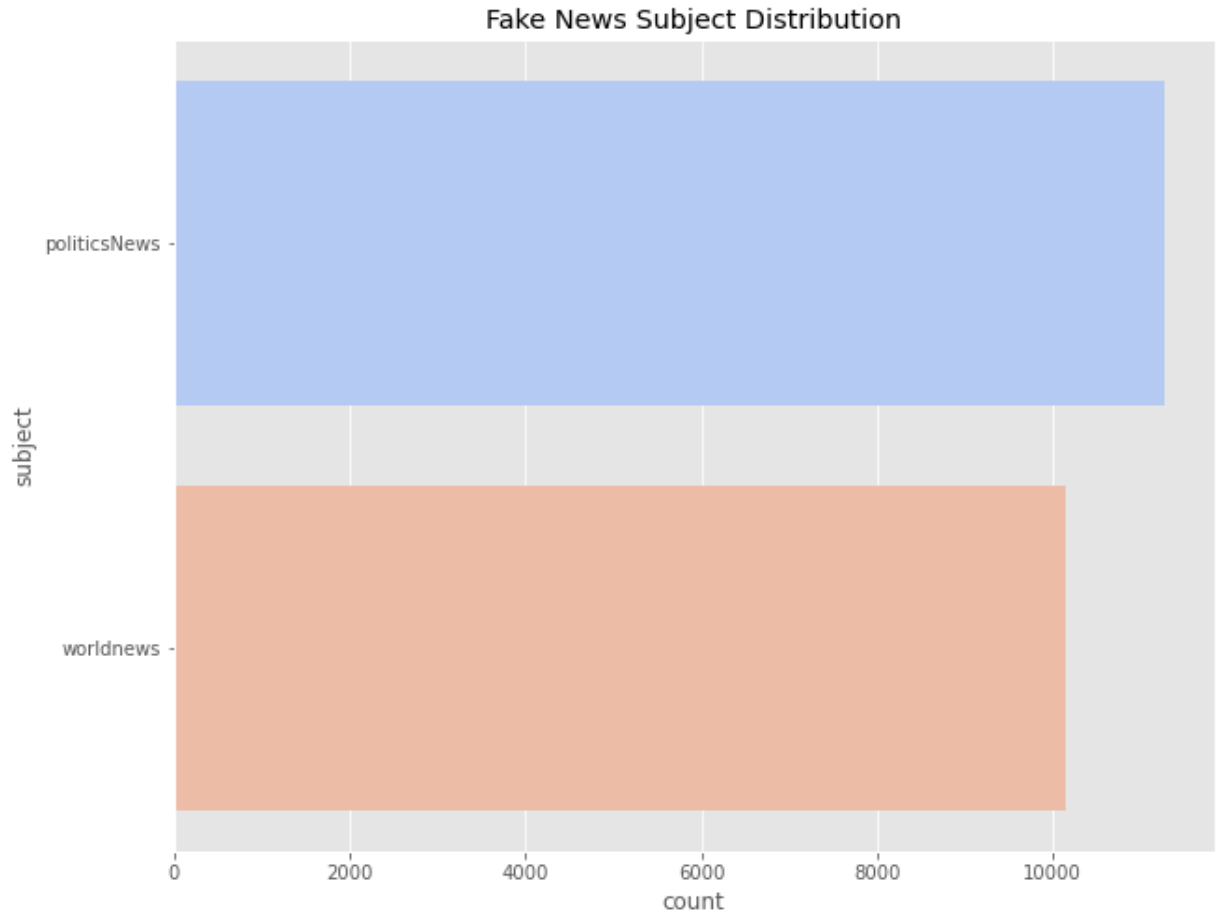
	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. 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
...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017

21417 rows × 4 columns

Data Visulization

In [156]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(y="subject", palette="coolwarm", data=true_df).set_title('True News
plt.show()

sns.countplot(y="subject", palette="coolwarm", data=fake_df).set_title('Fake News
plt.show()
```



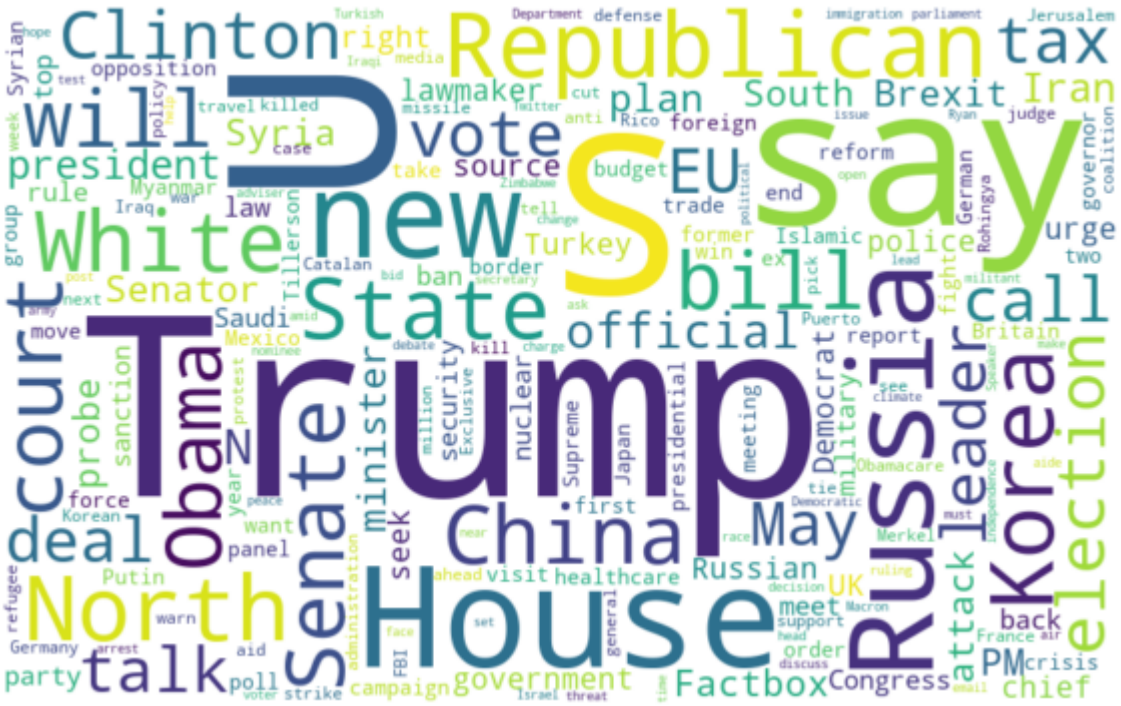


True News wordcloud

In [157]:
```
from wordcloud import WordCloud

real_titles = true_df.title
real_titles_ls = [text for text in real_titles]
# print(alls)
real_all_words = ' '.join(real_titles)
wordcloud_real = WordCloud(background_color='white',
    width= 800, height= 500,
    max_font_size = 180,
    collocations = False).generate(real_all_words)

plt.figure(figsize=(10,7))
plt.imshow(wordcloud_real, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Fake News Word Cloud

In [159]:
```
fake_titles = fake_df.title
fake_titles_ls = [text for text in fake_titles]
# print(alls)
fake_all_words = ' '.join(fake_titles)
wordcloud_fake = WordCloud(background_color='white',
    width= 800, height= 500,
    max_font_size = 180,
    collocations = False).generate(fake_all_words)

plt.figure(figsize=(10,7))
plt.imshow(wordcloud_fake, interpolation='bilinear')
plt.axis("off")
plt.show()
```



Data Preprocessing

Data Combination

In [161]:
```
# Add Labels to both df
true_df['true'] = 1
fake_df['true'] = 0

# Concat
df = pd.concat([true_df, fake_df])
display(df)
```

	title	text	subject	date	true
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	1
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	1
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017	1
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	1
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	1
...	...	...	...	...	...
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	0
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	0
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	0
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	0
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	0

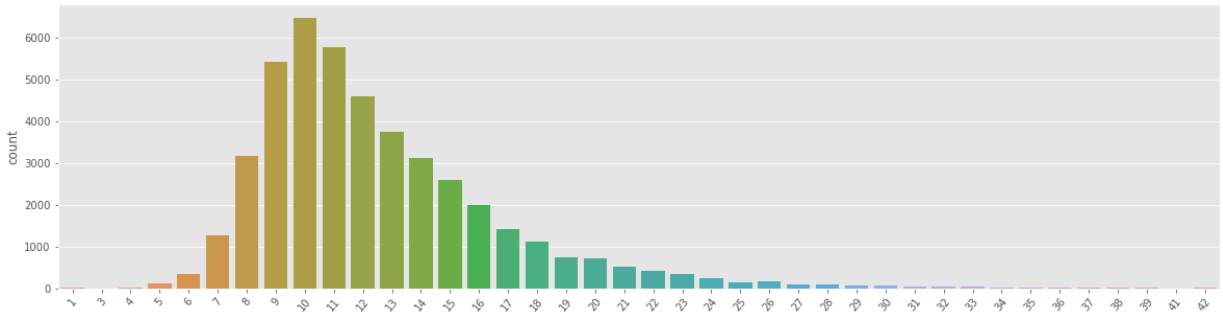
44898 rows × 5 columns

Inspect Lengths of News

In [163]:
```
titles = [text for text in df.title]

max_len = 0
titles_len = []
for title in titles:
    titles_len.append(len(title.split()))
    max_len = max(len(title.split()), max_len)

print('Number of titles:', len(titles))
print('Max length of the titles:', max_len)
print('Mean length of the titles:', np.mean(titles_len))
```

```
Number of titles: 44898
Max length of the titles: 42
Mean length of the titles: 12.453472315025168
```

In [164]:
```
plt.figure(figsize=(20,5))
g = sns.countplot(x=titles_len)
g.set_xticklabels(g.get_xticklabels(), rotation=50)
plt.show()
```



In [165]:
```
texts = [text for text in df.text]

max_len = 0
texts_len = []
for text in texts:
    texts_len.append(len(text.split()))
    max_len = max(len(text.split()), max_len)

# g = sns.countplot(x=texts_len)
print('Mean length of the texts:', np.mean(texts_len))
```

```
Mean length of the texts: 405.28228428883244
```

Purify & Shffle the DataFrame

In [167]:
```
from sklearn.utils import shuffle

# Purify
df = df.iloc[:,[0, -1]]

# Shuffle
df = shuffle(df).reset_index(drop=True)

display(df)
```

	title	true
0	MOCKINGBIRD REDUX? CNN’s Role in Peddling Fake...	1
1	Zakharova Slams CIA Chief Pompeo: Stop Making ...	1
2	Factbox: The companies making money from Illin...	0
3	McCain says time for Republicans, Democrats to...	0
4	This Anti-Government Oregon Terrorist Took Th...	1
...	...	...
44893	Pope says Colombia must confront inequality to...	0
44894	Trump: Being friends with North Korea's Kim is...	0
44895	Tillerson says never considered resigning	0
44896	Trump's Washington foreign policy speech	0
44897	Trump Tweets Deranged, Self-Congratulatory ‘F...	1

44898 rows × 2 columns

Split Data into Training, Validation, Test

In [169]:
```
train_val_df = df.sample(frac = 0.8)
test_df = df.drop(train_val_df.index)

train_df = train_val_df.sample(frac = 0.8)
val_df = train_val_df.drop(train_df.index)

# Reset Index
train_df = train_df.reset_index(drop=True)
val_df = val_df.reset_index(drop=True)
test_df = test_df.reset_index(drop=True)

print('trainset size:', train_df.shape)
print('valset size:', val_df.shape)
print('testset size:', test_df.shape)
```

```
trainset size: (28734, 2)
valset size: (7184, 2)
testset size: (8980, 2)
```

Dataframe to csv

In [171]:
```
train_df.to_csv('train.tsv', sep='\t', index=False)
val_df.to_csv('val.tsv', sep='\t', index=False)
test_df.to_csv('test.tsv', sep='\t', index=False)
```

Concatenate all dataframe

In [173]:
```
df = pd.concat([train_df, val_df, test_df])
df
```

Out[173]:

	title	true
0	WATCH: Powerful Video PERFECTLY Explains How ...	1
1	TOP DETECTIVES GIVE REASONS They Are Shocked N...	1
2	South Africa's Zuma rejects reports his office...	0
3	BRAIN FREEZE! HILLARY CLINTON Goes Blank…Forge...	1
4	HA! THEY'RE NOT FORGETTING About Hillary At Th...	1
...	...	...
8975	"YOU'RE HIRED!" Trump Pulls Unemployed Vet Fro...	1
8976	WHOA: Russians Hacked Voting Systems In 39 St...	1
8977	Unwavering Trump voters say they will not miss...	0
8978	U.S. says identified target for sanctions over...	0
8979	U.N. discussing resolution after Syrian govern...	0

44898 rows × 2 columns

Performing Data Cleaning

In [175]:
```
import nltk
# Downloading Stopwords
nltk.download("stopwords")
```

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

Out[175]: True

In [176]:
```
# Obtaining Additional Stopwords From nltk
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
```

In [177]:
```
# Removing Stopwords And Remove Words With 2 Or Less Characters
def preprocess(text):
    result = []
    for token in gensim.utils.simple_preprocess(text):
        if token not in gensim.parsing.preprocessing.STOPWORDS and len(token) > 3
            result.append(token)

    return result
```

In [178]:
```
import gensim

# Applying The Function To The Dataframe
df['clean'] = df['title'].apply(preprocess)
```

Obtaining The Total Words Present In The Dataset

In [180]:
```
list_of_words = []
for i in df.clean:
    for j in i:
        list_of_words.append(j)

total_words = len(list(set(list_of_words)))
total_words
```

Out[180]: 19174

Preparing The Data By Performing Tokenization And Padding

In [182]:
```
from nltk import word_tokenize
```

In [183]:
```
from tensorflow.keras.preprocessing.text import one_hot, Tokenizer

# Creating A Tokenizer To Tokenize The Words And Create Sequences Of Tokenized Wo
tokenizer = Tokenizer(num_words = total_words)
tokenizer.fit_on_texts(train_df['title'])

train_sequences = tokenizer.texts_to_sequences(train_df['title'])
val_sequences = tokenizer.texts_to_sequences(val_df['title'])
test_sequences = tokenizer.texts_to_sequences(test_df['title'])
```

In [184]:
```
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Adding Padding
padded_train = pad_sequences(train_sequences,maxlen = 42, padding = 'post', trunc
padded_val = pad_sequences(val_sequences,maxlen = 42, padding = 'post', truncatin
padded_test = pad_sequences(test_sequences,maxlen = 42, padding = 'post', truncat
```

In [188]:
```
!pip install transformers

import torch
from transformers import BertTokenizer
```

```
Requirement already satisfied: transformers in c:\users\laptop inn\anaconda3\ne
w folder\lib\site-packages (4.20.1)
Requirement already satisfied: tokenizers!=0.11.3,<0.13,>=0.11.1 in c:\users\la
ptop inn\anaconda3\new folder\lib\site-packages (from transformers) (0.12.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in c:\users\laptop i
nn\anaconda3\new folder\lib\site-packages (from transformers) (0.8.1)
Requirement already satisfied: numpy>=1.17 in c:\users\laptop inn\anaconda3\new
folder\lib\site-packages (from transformers) (1.20.3)
Requirement already satisfied: requests in c:\users\laptop inn\anaconda3\new fo
lder\lib\site-packages (from transformers) (2.26.0)
Requirement already satisfied: packaging>=20.0 in c:\users\laptop inn\anaconda3
\new folder\lib\site-packages (from transformers) (21.0)
Requirement already satisfied: filelock in c:\users\laptop inn\anaconda3\new fo
lder\lib\site-packages (from transformers) (3.3.1)
Requirement already satisfied: regex!=2019.12.17 in c:\users\laptop inn\anacond
a3\new folder\lib\site-packages (from transformers) (2021.8.3)
Requirement already satisfied: tqdm>=4.27 in c:\users\laptop inn\anaconda3\new
folder\lib\site-packages (from transformers) (4.62.3)
Requirement already satisfied: pyyaml>=5.1 in c:\users\laptop inn\anaconda3\new
folder\lib\site-packages (from transformers) (6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\laptop in
n\anaconda3\new folder\lib\site-packages (from huggingface-hub<1.0,>=0.1.0->tra
nsformers) (3.10.0.2)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\laptop inn\anaconda
3\new folder\lib\site-packages (from packaging>=20.0->transformers) (3.0.4)
Requirement already satisfied: colorama in c:\users\laptop inn\anaconda3\new fo
lder\lib\site-packages (from tqdm>=4.27->transformers) (0.4.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\laptop inn\anacon
da3\new folder\lib\site-packages (from requests->transformers) (2022.6.15)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\laptop inn
\anaconda3\new folder\lib\site-packages (from requests->transformers) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\laptop inn\ana
conda3\new folder\lib\site-packages (from requests->transformers) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\laptop inn\anaconda3\ne
w folder\lib\site-packages (from requests->transformers) (3.2)
```

In [189]:
```
PRETRAINED_MODEL_NAME = 'bert-base-uncased'
tokenizer = BertTokenizer.from_pretrained(PRETRAINED_MODEL_NAME)
```

Load Dataset Class

In [191]:
```
from torch.utils.data import Dataset

class FakeNewsDataset(Dataset):
    def __init__(self, mode, tokenizer):
        assert mode in ['train', 'val', 'test']
        self.mode = mode
        self.df = pd.read_csv(mode + '.tsv', sep='\t').fillna("")
        self.len = len(self.df)
        self.tokenizer = tokenizer  # BERT tokenizer

    # Define a function that returns a training/test data
    def __getitem__(self, idx):
        if self.mode == 'test':
            statement, label = self.df.iloc[idx, :].values
            label_tensor = torch.tensor(label)
        else:
            statement, label = self.df.iloc[idx, :].values
            label_tensor = torch.tensor(label)


# Create BERT tokens for the first sentence and add separators [SEP
        word_pieces = ['[CLS]']
        statement = self.tokenizer.tokenize(statement)
        word_pieces += statement + ['[SEP]']
        len_st = len(word_pieces)


        ids = self.tokenizer.convert_tokens_to_ids(word_pieces)
        tokens_tensor = torch.tensor(ids)

        # Set the token position of the first sentence containing [SEP] to 0
        segments_tensor = torch.tensor([0] * len_st, dtype=torch.long)

        return (tokens_tensor, segments_tensor, label_tensor)

    def __len__(self):
        return self.len


# Initialize Datasets for Transformation
trainset = FakeNewsDataset('train', tokenizer=tokenizer)
valset = FakeNewsDataset('val', tokenizer=tokenizer)
testset = FakeNewsDataset('test', tokenizer=tokenizer)

print('trainset size:' ,trainset.__len__())
print('valset size:',valset.__len__())
print('testset size: ',testset.__len__())
```

```
trainset size: 28734
valset size: 7184
testset size:  8980
```

Sampling and Observing Tensors

In [193]:
```
# select the first sample
sample_idx = 0


# take out the original text for comparison
statement, label = trainset.df.iloc[sample_idx].values


# Use the Dataset just created to take out the converted id tensors
tokens_tensor, segments_tensor, label_tensor = trainset[sample_idx]

# restore tokens_tensor to text
tokens = tokenizer.convert_ids_to_tokens(tokens_tensor.tolist())
combined_text = " ".join(tokens)
# The difference before and after rendering, no response is a print. You can see
print(f"""
original_statement:
{statement}

tokens:
{tokens}

label: {label}

--------------------

tokens_tensor:
{tokens_tensor}

segments_tensor:
{segments_tensor}

label_tensor:
{label_tensor}

""")
```

```
original_statement:
 WATCH: Powerful Video PERFECTLY Explains How And Why Trump Is Manipulating Whi
te People

tokens:
['[CLS]', 'watch', ':', 'powerful', 'video', 'perfectly', 'explains', 'how', 'a
nd', 'why', 'trump', 'is', 'manipulating', 'white', 'people', '[SEP]']

label: 1

--------------------

tokens_tensor:
tensor([  101,  3422,  1024,  3928,  2678,  6669,  7607,  2129,  1998,  2339,
         8398,  2003, 26242,  2317,  2111,   102])

segments_tensor:
tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
label_tensor:
1
```

In [194]:
```
# Reforming the Dataset to Fit the Model
```

In [195]:
```
from torch.utils.data import DataLoader
from torch.nn.utils.rnn import pad_sequence

# The input to this function `samples` is a list where each element is
# A sample returned by the just defined `FakeNewsDataset`, each containing 3 tens
# - tokens_tensor
# - segments_tensor
# - label_tensor
# It will zero-pad the first two tensors and generate the masks_tensors describe
def create_mini_batch(samples):
    tokens_tensors = [s[0] for s in samples]
    segments_tensors = [s[1] for s in samples]
# test set has labels
    if samples[0][2] is not None:
        label_ids = torch.stack([s[2] for s in samples])
    else:
        label_ids = None

    # Zero Padding
    tokens_tensors = pad_sequence(tokens_tensors, batch_first=True)
    segments_tensors = pad_sequence(segments_tensors, batch_first=True)

# attention masks, make tokens_tensors not zero padding
    # positions are set to 1 to make BERT only focus on tokens in these positions
    masks_tensors = torch.zeros(tokens_tensors.shape, dtype=torch.long)
    masks_tensors = masks_tensors.masked_fill(tokens_tensors != 0, 1)

    return tokens_tensors, segments_tensors, masks_tensors, label_ids


# Initialize a DataLoader that returns 16 training samples each time
# Using `collate_fn` to combine the list of samples into a mini-batch is the key
BATCH_SIZE = 16
trainloader = DataLoader(trainset, batch_size=BATCH_SIZE, collate_fn=create_mini_
valloader = DataLoader(valset, batch_size=BATCH_SIZE, collate_fn=create_mini_batc
testloader = DataLoader(testset, batch_size=BATCH_SIZE,collate_fn=create_mini_bat
```

In [196]:
```
data = next(iter(trainloader))

tokens_tensors, segments_tensors, masks_tensors, label_ids = data

print(f"""
tokens_tensors.shape   = {tokens_tensors.shape}
{tokens_tensors}
------------------------
segments_tensors.shape = {segments_tensors.shape}
{segments_tensors}
------------------------
masks_tensors.shape    = {masks_tensors.shape}
{masks_tensors}
------------------------
label_ids.shape        = {label_ids.shape}
{label_ids}
""")
```

```
tokens_tensors.shape   = torch.Size([16, 36])
tensor([[  101,  3422,  1024,  3928,  2678,  6669,  7607,  2129,  1998,  233
9,
          8398,  2003, 26242,  2317,  2111,   102,     0,     0,     0,
0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,
0,
             0,     0,     0,     0,     0,     0],
        [  101,  2327, 18145,  2507,  4436,  2027,  2024,  7135,  2053, 2453
4,
          2097,  2022,  2864,  2006,  3425,  8040, 22786,  1024,  1523,  104
5,
          2165,  1037,  2298,  2012,  1996,  3189,  1998,  1045,  2471,  306
2,
          2041,  1997,  2026,  3242,  1524,   102],
        [  101,  2148,  3088,  1005,  1055, 16950,  2863, 19164,  4311,  201
0,
          2436,  2003, 21168,  5057,  4277,   102,     0,     0,     0,
```

Model Construction

In [198]:
```
from transformers import BertForSequenceClassification
from IPython.display import display, clear_output

PRETRAINED_MODEL_NAME = "bert-base-uncased"
NUM_LABELS = 2

model = BertForSequenceClassification.from_pretrained(
    PRETRAINED_MODEL_NAME, num_labels=NUM_LABELS)

clear_output()
# high-level show modules in this model
print("""
name             module
----------------------""")
for name, module in model.named_children():
    if name == "bert":
        for n, _ in module.named_children():
            print(f"{name}:{n}")
    else:
        print("{:16} {}".format(name, module))
```

```
name             module
----------------------
bert:embeddings
bert:encoder
bert:pooler
dropout          Dropout(p=0.1, inplace=False)
classifier       Linear(in_features=768, out_features=2, bias=True)
```

In [199]:
```
model.config
```

Out[199]:

```
BertConfig {
  "_name_or_path": "bert-base-uncased",
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "position_embedding_type": "absolute",
  "transformers_version": "4.20.1",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 30522
}
```


Fine-Tuning of BERT

In [*]:
```
# %%time
from sklearn.metrics import accuracy_score
from tqdm.notebook import tqdm

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print("device:", device)
model = model.to(device)

model.train()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-5)
NUM_EPOCHS = 3

for epoch in range(NUM_EPOCHS):
    train_loss = 0.0
    train_acc = 0.0

    loop = tqdm(trainloader)
    for batch_idx, data in enumerate(loop):
        tokens_tensors, segments_tensors, masks_tensors, labels = [t.to(device) f

       # zero the parameter gradient
        optimizer.zero_grad()

        outputs = model(input_ids=tokens_tensors,
                        token_type_ids=segments_tensors,
                        attention_mask=masks_tensors,
                        labels=labels)

        loss = outputs[0]
        loss.backward()
        optimizer.step()

        logits = outputs[1]
        _, pred = torch.max(logits.data, 1)
        train_acc = accuracy_score(pred.cpu().tolist() , labels.cpu().tolist())

       # Record the current batch loss
        train_loss += loss.item()

        # if batch_idx == len(trainloader)-1:
        #     _, acc = get_predictions(model, trainloader, compute_acc=True)

        loop.set_description(f"Epoch [{epoch+1}/{NUM_EPOCHS}]")
        loop.set_postfix(acc = train_acc, loss = train_loss)
```

```
device: cpu
```

Epoch [1/3]: 13% 233/1796 [24:02<3:15:04, 7.49s/it, acc=1, loss=53.5]

In [ ]: