FAKE NEWS DETECTION USING BERT

Project Presentation
20.05.2022

Presented by:
Sreethu P R

Introduction

- The fake news may be on different domains such as political domain, entertainment domain, sports domain etc.
- Classifying fake news using models based on a natural language processing framework, **B**idirectional **E**ncoder **R**epresentations from **T**ransformers, also known as BERT is the most competitive study happened now.
- BERT is pre-trained on a large corpus of unlabelled text including the entire Wikipedia(that's 2,500 million words) and Book Corpus (800 million words).
- BERT is a deeply bidirectional model. Bidirectional means that BERT learns information from both the left and the right side of a token's context during the training phase.
- We can fine-tune it by adding just a couple of additional output layers to create state-of-the-art models for a variety of NLP tasks.

Related works

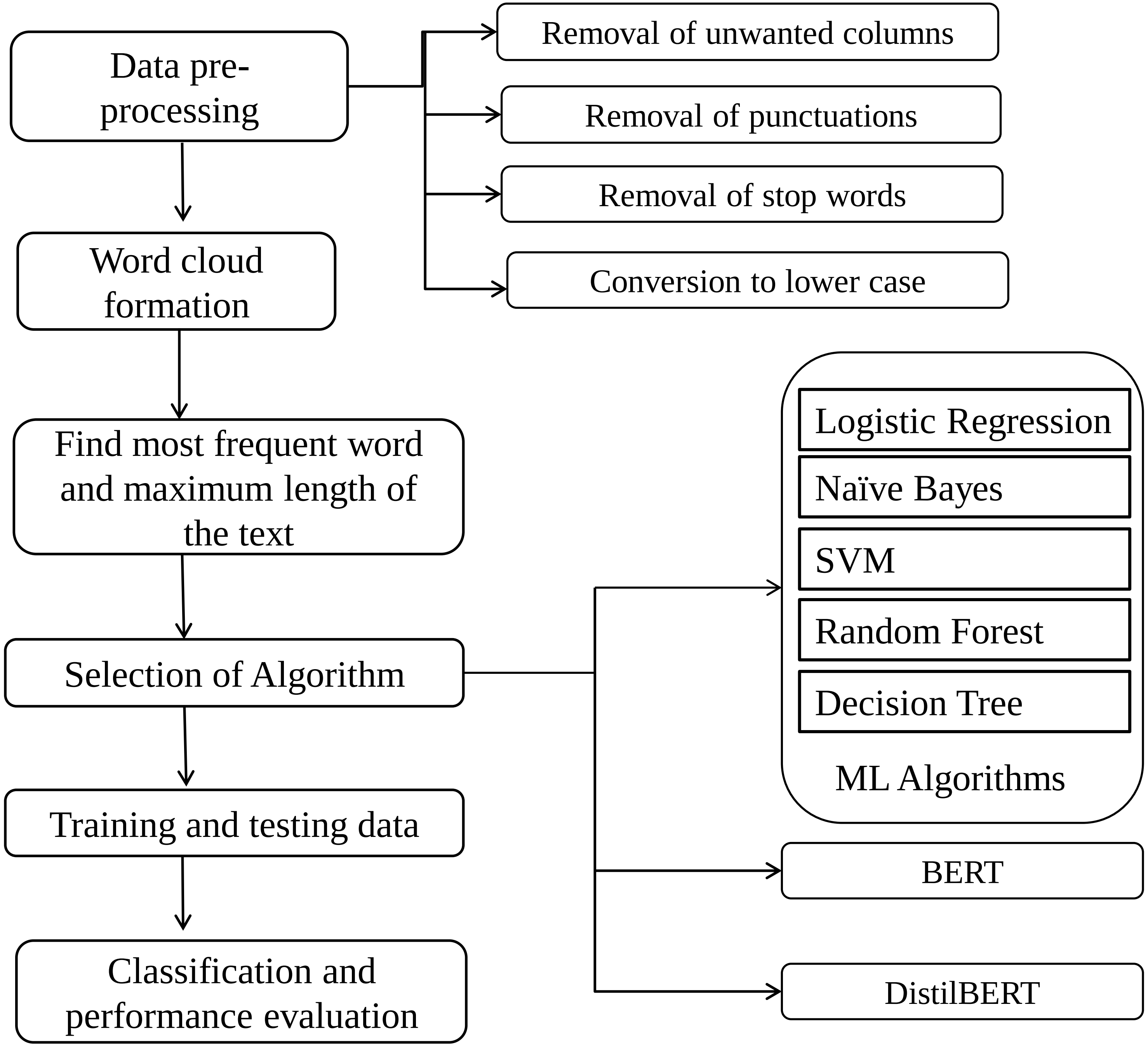
No	Reference	Title	Technique used	Advantage	Disadvantage/ Future work
1	[1] Springer (2021)	Implementation of the BERT-derived architectures to tackle disinformation challenges	BERT, RoBERTa, RNN	• This model is solid and reliable, ready to use in real-time fake news detection systems.	• Retrain the model effectively and can be used in various domains.
2	[2] arxiv (2021)	Multimodal Fusion with BERT and Attention Mechanism for Fake News Detection	BERTweet model and VGG-19 network.	• Scale dot product attention mechanism to capture the relationship between text features and visual features	• Very ambiguous when using the picture to express the content of the tweet.

Objective

A generalised model for fake news detection using BERT and DistilBERT, also do a comparison using different ML algorithms.

Architecture

- Data pre-processing helps to align the entire data in same format.
- Word Clouds are visual displays of text data – simple text analysis.
- It display the most prominent or frequent words in a body of text , ignore the most common words in the language (“a”, “an”, “the” etc.).
- Train and test the dataset on the created models.
- Obtain the accuracy and compare it with different models.



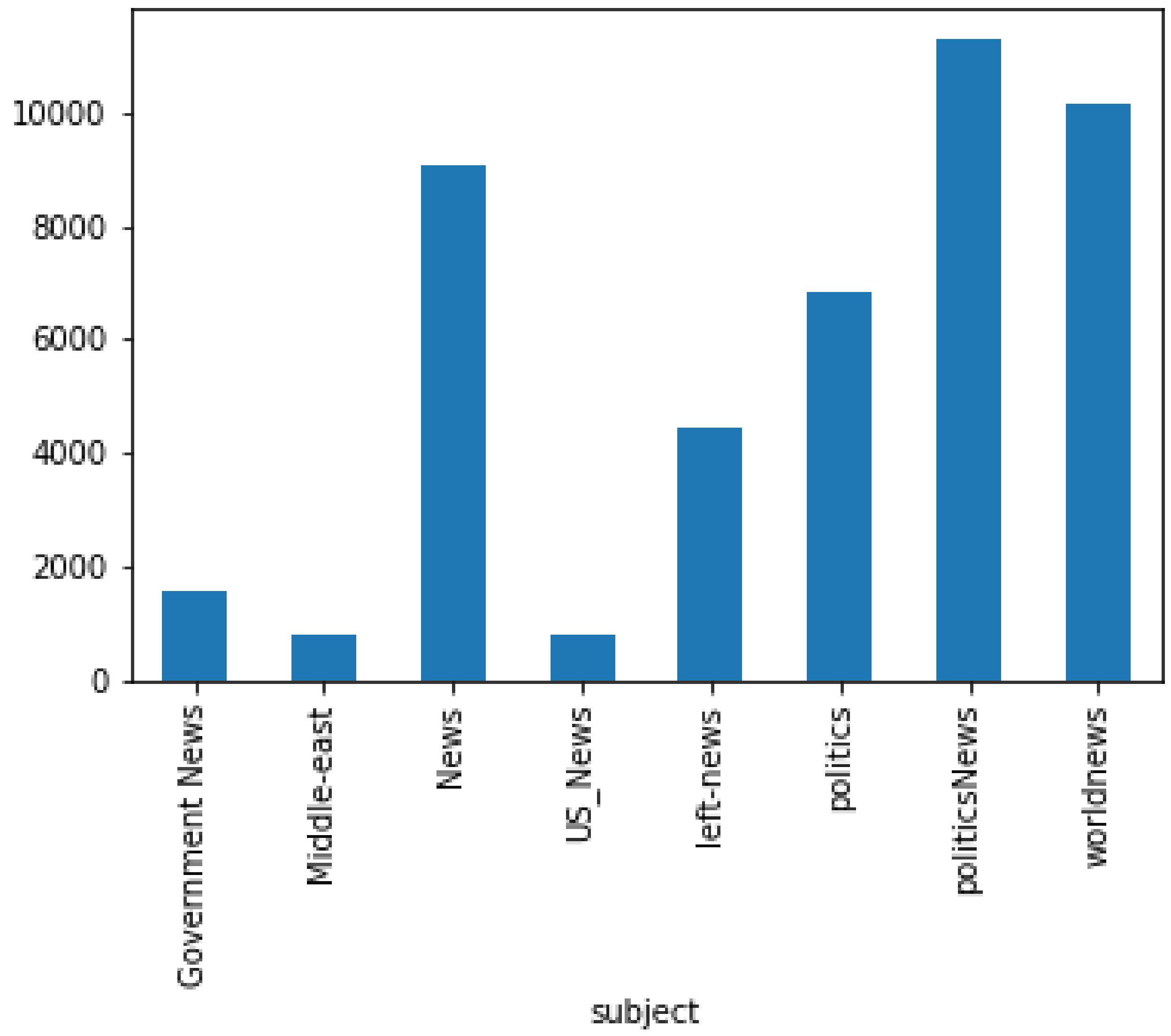
Frameworks

- Framework: - Keras with Tensorflow, pytorch as background in the Google Colab and Power edge server with NVDIA TESLA V100 GPU
- Library: - HuggingfaceTransformer, NumPy, Pandas, Matplotlib, NLTK

Datasets

- The dataset contains two sets of CSV files (true and fake) in 8 subjects such as Government news, Middle-east, News, US_news, left-news, politics, politicsNews, worldnews.
- true contains 21417 (politicsNews and worldnews).
- fake contains 23481 (Government news, Middle-east, News, US_news, left-news and politics).

ISOT Dataset

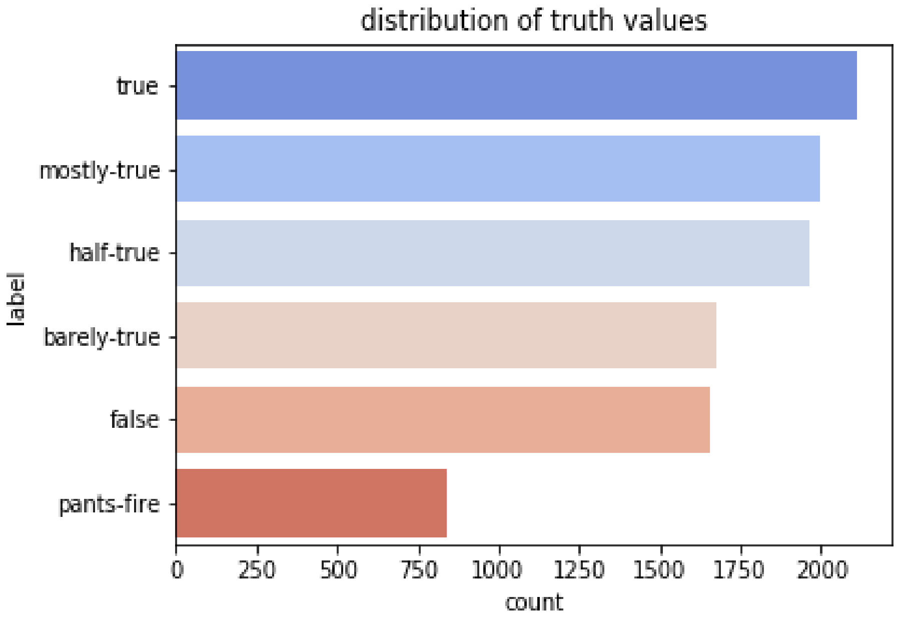


Datasets

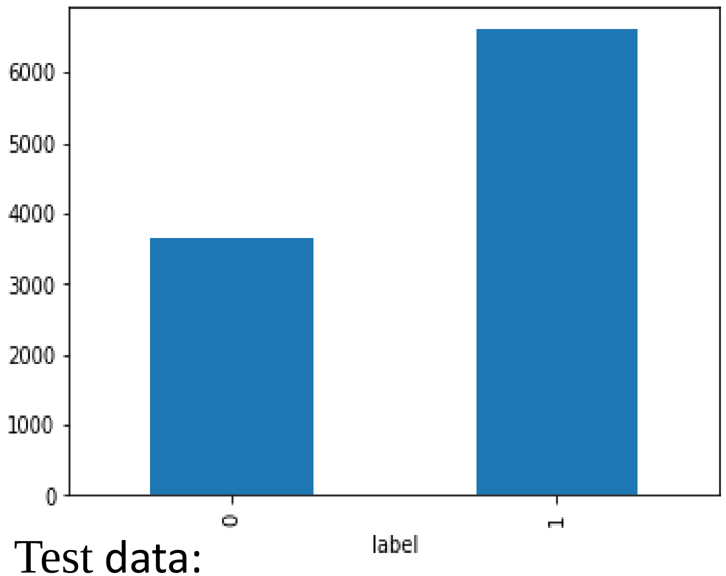
- The LIAR dataset was published by William Yang in July 2017.
- He retrieved the data from PolitiFact's API.
- This website collects statements made by US 'speakers' and assigns a truth value to them ranging from 'True' to 'Pants on Fire'.

File	Statements
train.tsv	10,240
valid.tsv	1,284
test.tsv	1,267

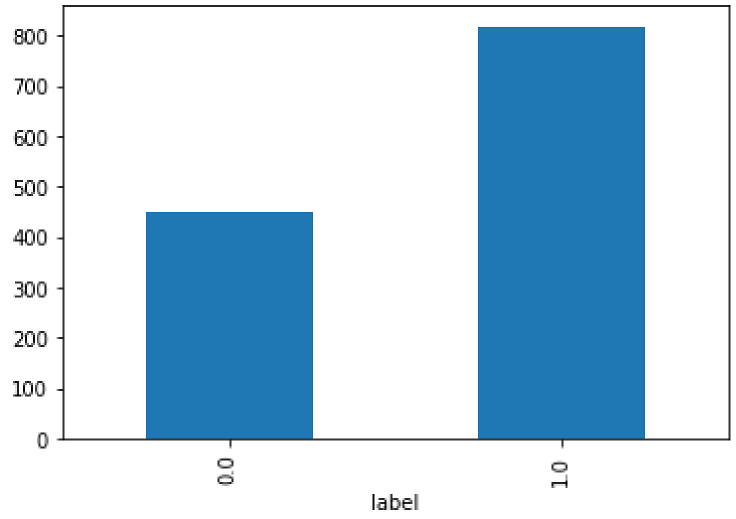
LIAR Dataset



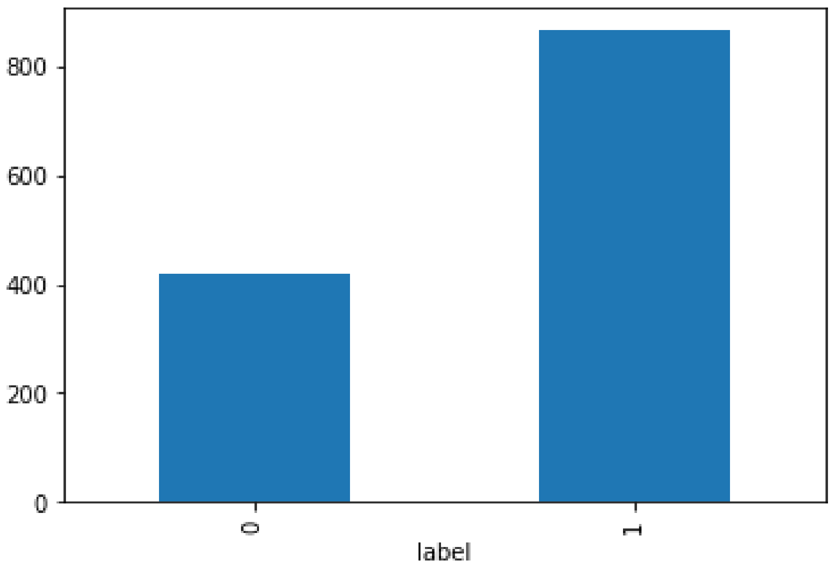
Train data:



Test data:



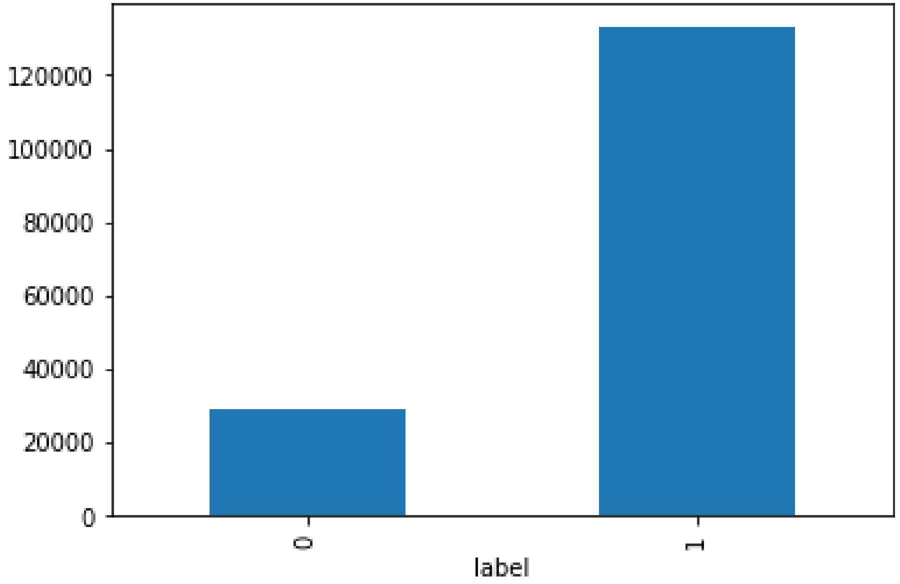
Validation data:



Datasets

- Contains 161743 data related to Covid-19 pandemic
- Disadvantage: Class imbalance

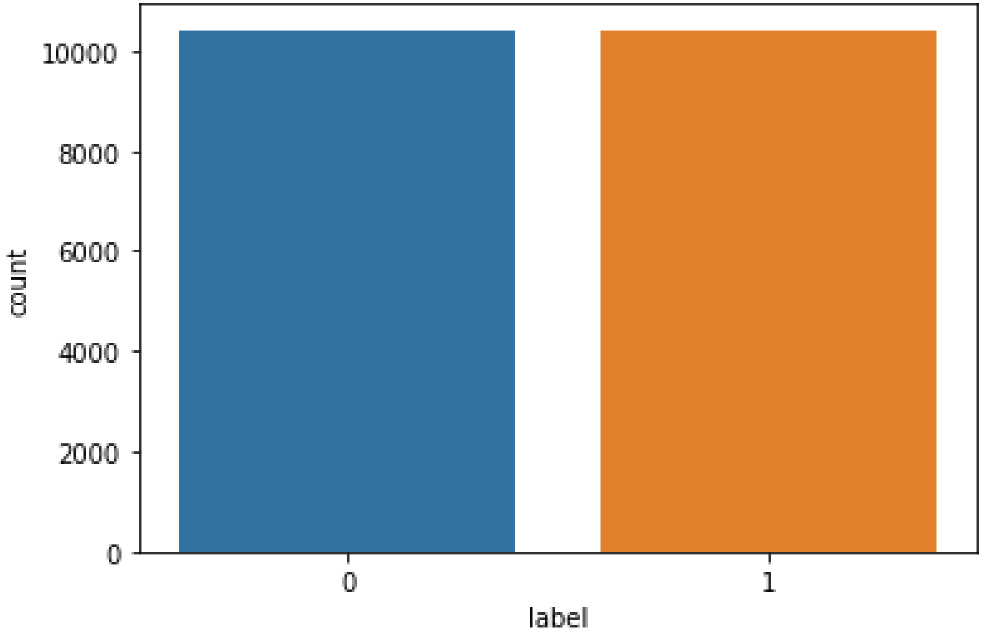
Twitter Dataset



- Contains two sets of CSV files: Train(20776) and Test(5201)
- Training and testing dataset have the attributes:

  id: unique id for a news article
  title: the title of a news article
  author: author of the news article
  text: the text of the article; could be incomplete
  label: a label that marks the article as potentially unreliable

Kaggle Dataset



Data Preprocessing

Pre-processing on LIAR dataset

- It contains so many unwanted columns so that first step is the elimination of them.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	12134.json	barely-true	We have less Americans working now than in the...	economy,jobs	vicky-hartzler	U.S. Representative	Missouri	republican	1	0	1	0	0	an interview with ABC17 News
1	238.json	pants-fire	When Obama was sworn into office, he DID NOT u...	obama-birth-certificate,religion	chain-email	NaN	NaN	none	11	43	8	5	105	NaN
2	7891.json	false	Says Having organizations parading as being so...	campaign-finance,congress,taxes	earl-blumenauer	U.S. representative	Oregon	democrat	0	1	1	1	0	a U.S. Ways and Means hearing
3	8169.json	half-true	Says nearly half of Oregons children are poor.	poverty	jim-francesconi	Member of the State Board of Higher Education	Oregon	none	0	1	1	1	0	an opinion article
4	929.json	half-true	On attacks by Republicans that various program...	economy,stimulus	barack-obama	President	Illinois	democrat	70	71	160	163	9	interview with CBS News

	1	2
0	barely-true	We have less Americans working now than in the...
1	pants-fire	When Obama was sworn into office, he DID NOT u...
2	false	Says Having organizations parading as being so...
3	half-true	Says nearly half of Oregons children are poor.
4	half-true	On attacks by Republicans that various program...

Data Preprocessing (Contd.)

Pre-processing on LIAR dataset (Contd.)

- The news were classified into 6 categories: True, half-true, Mostly-true, Barely-true, False, Pants-fire.
- Here we do binary classification, so map true and Mostly true into True (0) category and others into fake(1).
- Also give name to the columns like text and label.

	text	label
0	Says the Annies List political group supports ...	1
1	When did the decline of coal start? It started...	1
2	Hillary Clinton agrees with John McCain "by vo...	0
3	Health care reform legislation is likely to ma...	1
4	The economic turnaround started at the end of ...	1
...	...	...
10235	There are a larger number of shark attacks in ...	0
10236	Democrats have now become the party of the [At...	0
10237	Says an alternative to Social Security that op...	1
10238	On lifting the U.S. Cuban embargo and allowing...	1
10239	The Department of Veterans Affairs has a manua...	1

10240 rows × 2 columns

- Remove stop words, punctuation and also convert all the text into lower case.

	text	label
0	say annies list political group support thirdt...	1
1	decline coal start started natural gas took st...	1
2	hillary clinton agrees john mccain voting give...	0
3	health care reform legislation likely mandate ...	1
4	economic turnaround started end term	1

Data Preprocessing (Contd.)

Pre-processing on ISOT dataset

- Combine fake and true data files and shuffle the rows and put fake=1 and true=0

Out[6]:

	title	text	subject	date	target
0	Republican ex-Treasury chief Paulson slams Tru...	WASHINGTON (Reuters) - Henry Paulson, a Republ...	politicsNews	June 25, 2016	0
1	SHOCK POLL In MUST WIN State Of FLORIDA: Hispa...	Apparently the Black Lives Matter terror group...	left-news	Jul 11, 2016	1
2	MEDALS OF VALOR: President Trump Honored Agent...	It s great to have a president who appreciates...	politics	Jul 27, 2017	1
3	Newsweek Just Made Their BEST Cover Ever And ...	Newsweek has never been a publication to shy a...	News	November 9, 2017	1
4	Trump says he believes Cuba responsible for at...	WASHINGTON (Reuters) - President Donald Trump ...	politicsNews	October 16, 2017	0

- Then remove punctuation, stop words and unwanted columns from the dataset, also convert the text into lower case.

Out[17]:

	title	text	target
0	republican extreasury chief paulson slam trump...	washington reuters henry paulson republican u ...	0
1	shock poll must win state florida hispanic tur...	apparently black life matter terror group mana...	1
2	medal valor president trump honored agent offi...	great president appreciates special agent poli...	1
3	newsweek made best cover ever people freaking	newsweek never publication shy away controvers...	1
4	trump say belief cuba responsible attack hurt ...	washington reuters president donald trump said...	0

Data Preprocessing (Contd.)

Pre-processing on Kaggle dataset

- Remove the columns of author and id

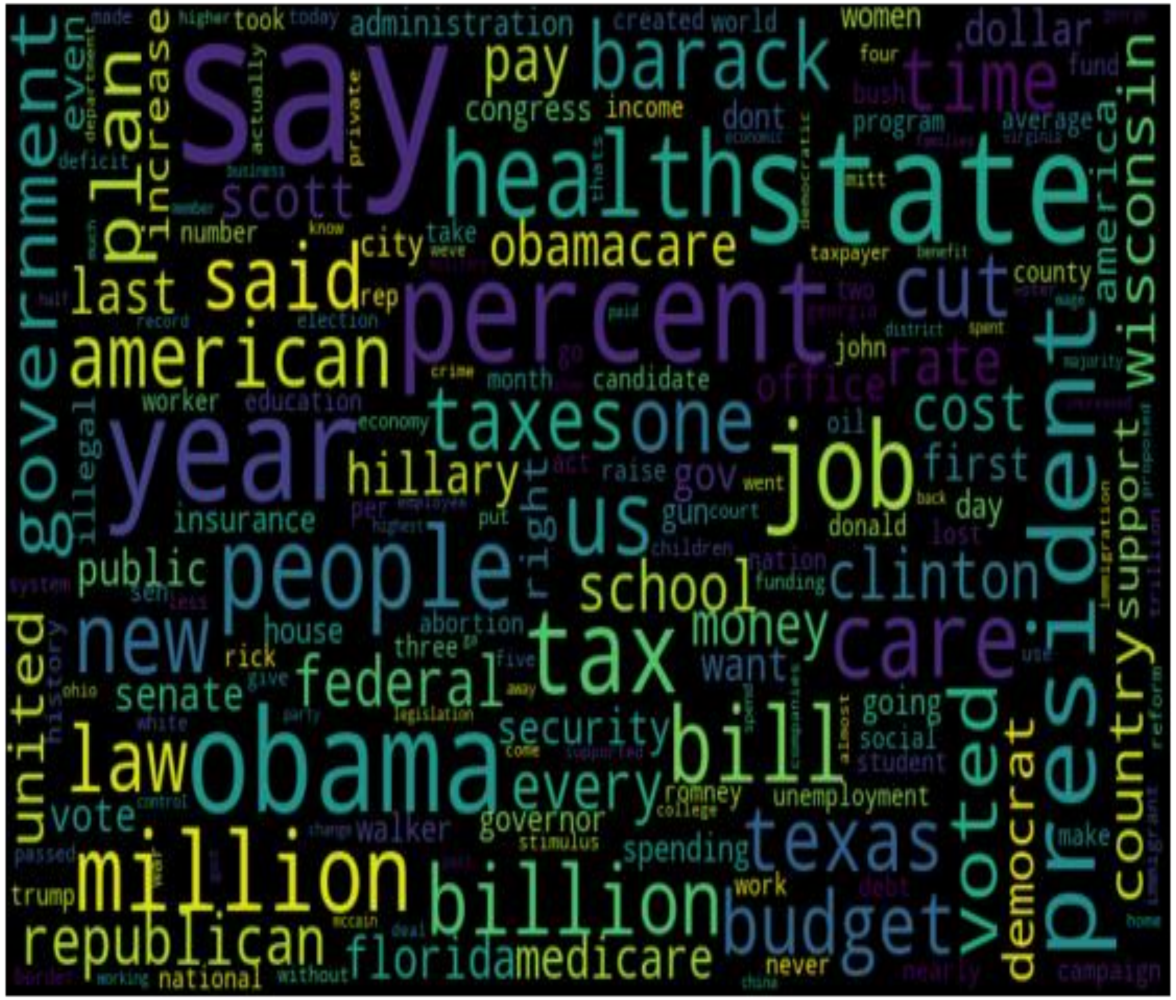
	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

- Then remove punctuation, stop words and unwanted columns from the dataset, also convert the text into lower case.

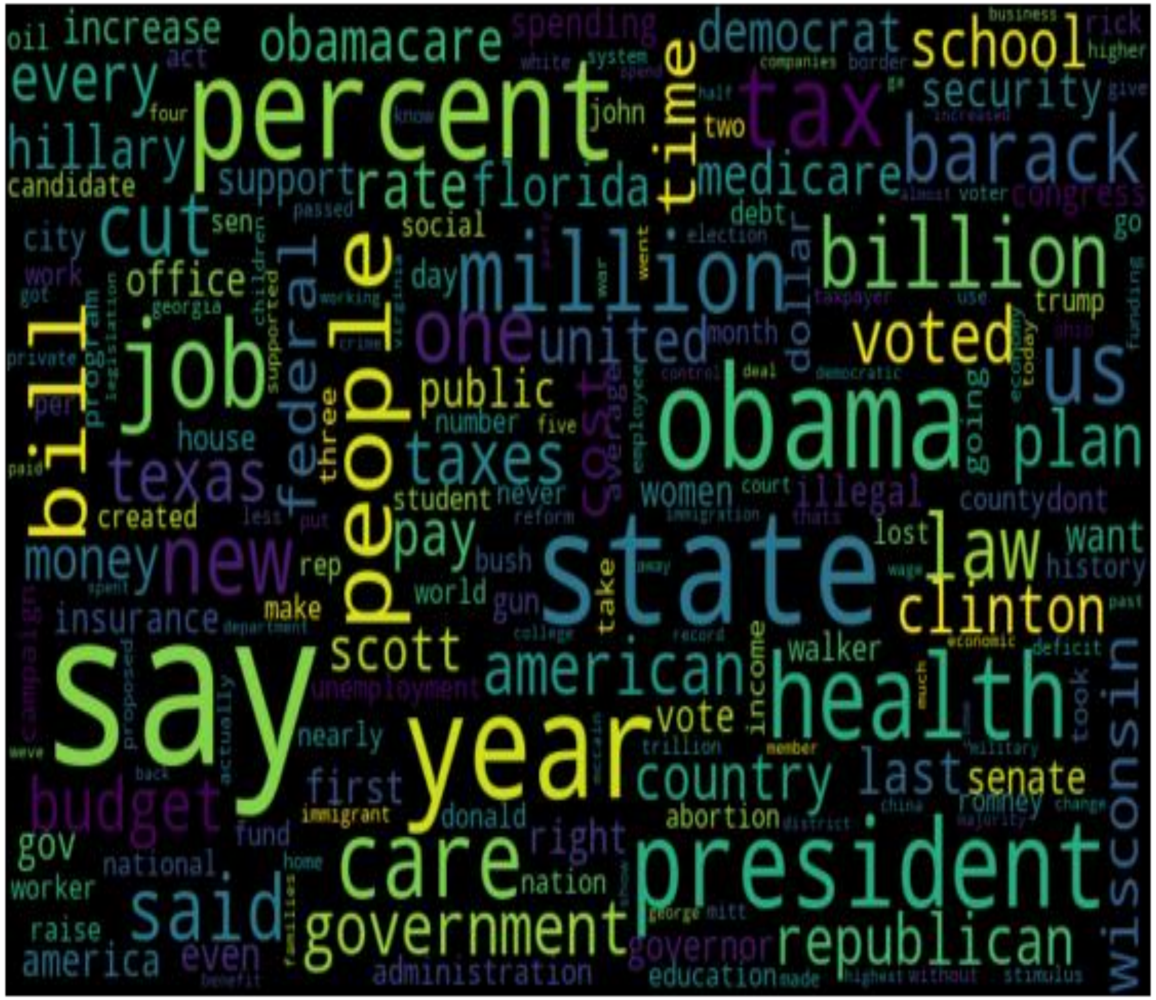
	title	text	label
0	house dem aide didnt even see comeys letter ja...	house dem aide didnt even see comeys letter ja...	1
1	flynn hillary clinton big woman campus breitbart	ever get feeling life circle roundabout rather...	0
2	truth might get fired	truth might get fired october 29 2016 tension ...	1
3	15 civilian killed single u airstrike identified	video 15 civilian killed single u airstrike id...	1
4	iranian woman jailed fictional unpublished sto...	print iranian woman sentenced six year prison ...	1

Word cloud - LIAR

Fake

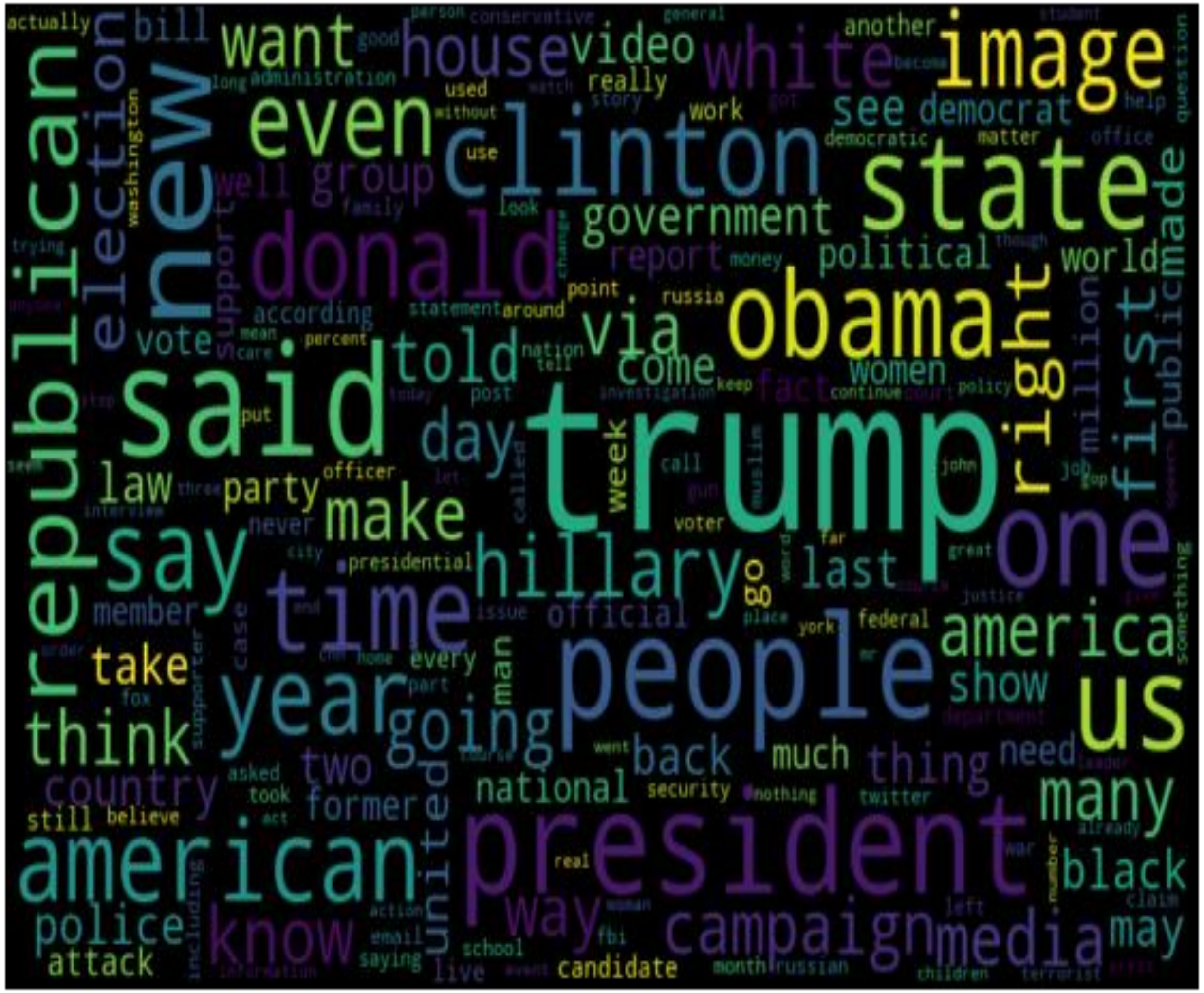


Real

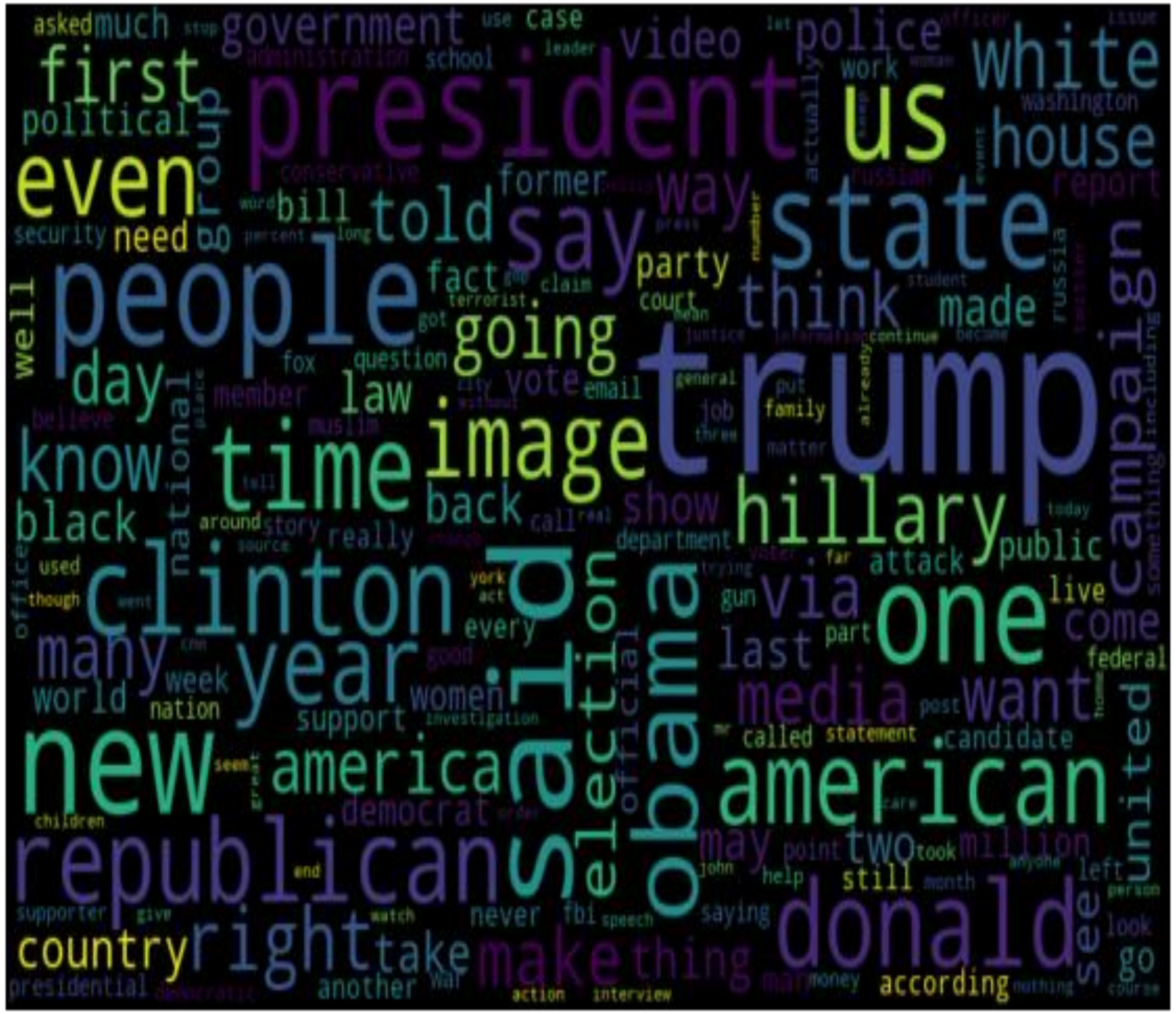


Word cloud - ISOT

Fake

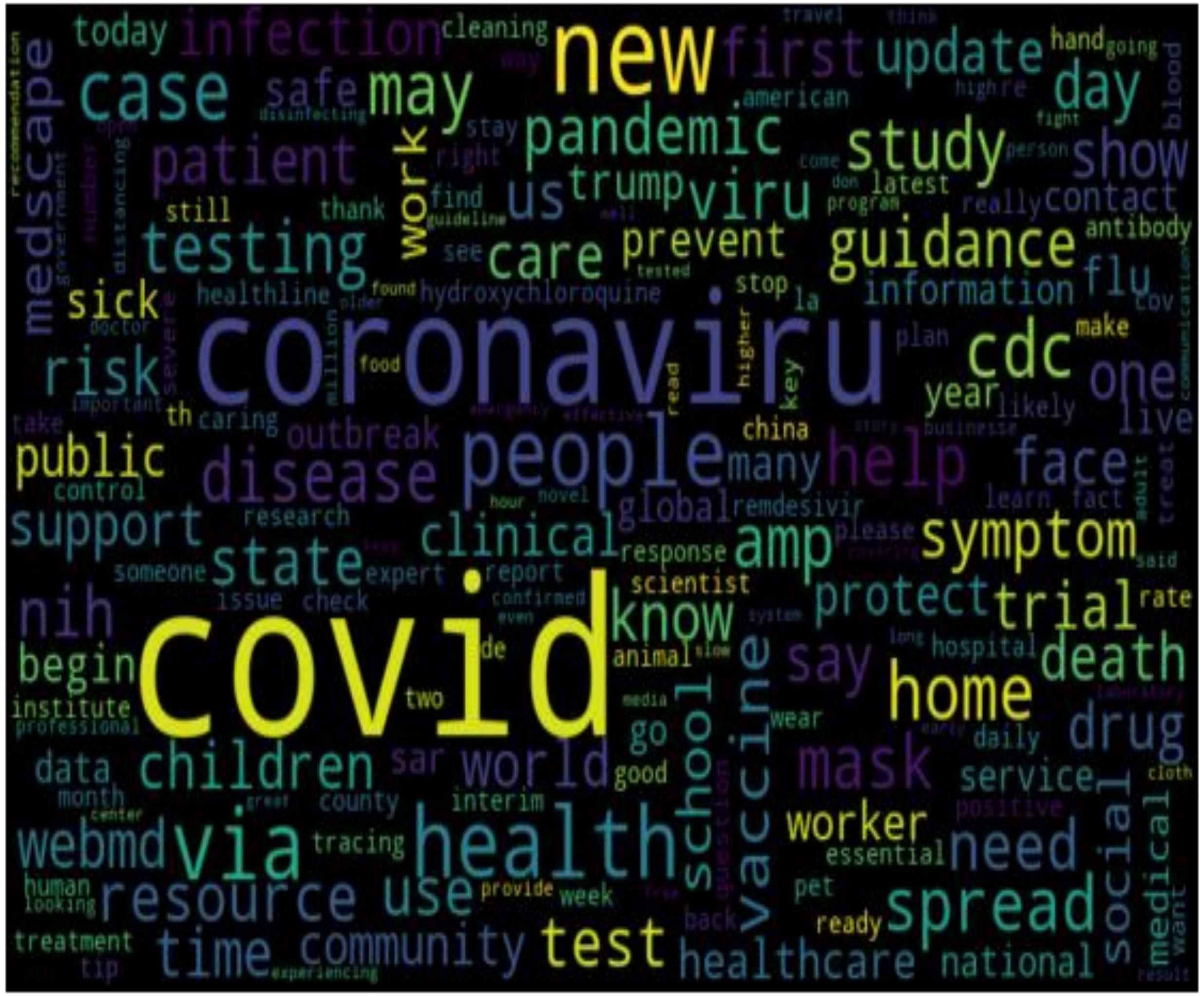


Real

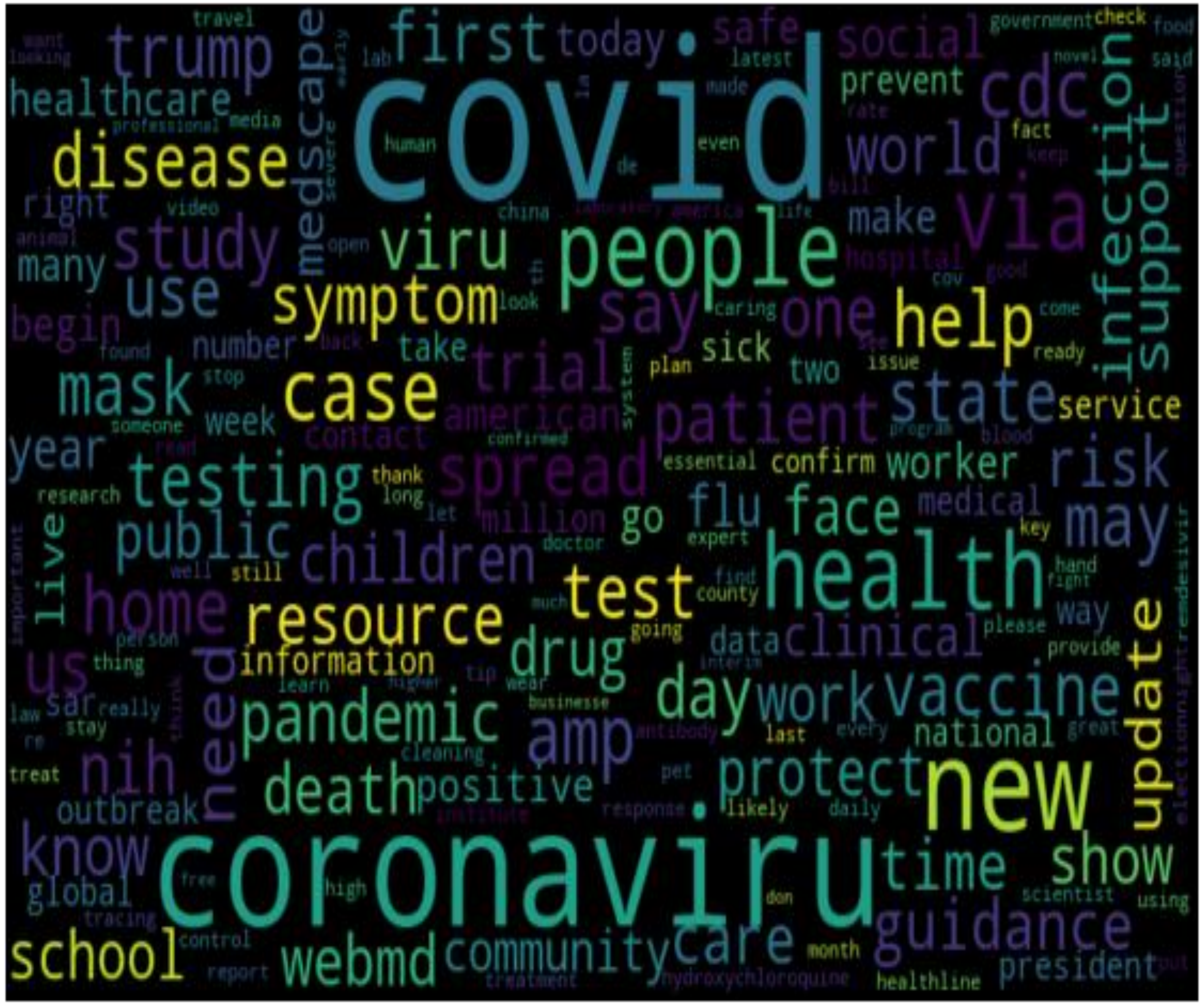


Word cloud - Twitter

Fake

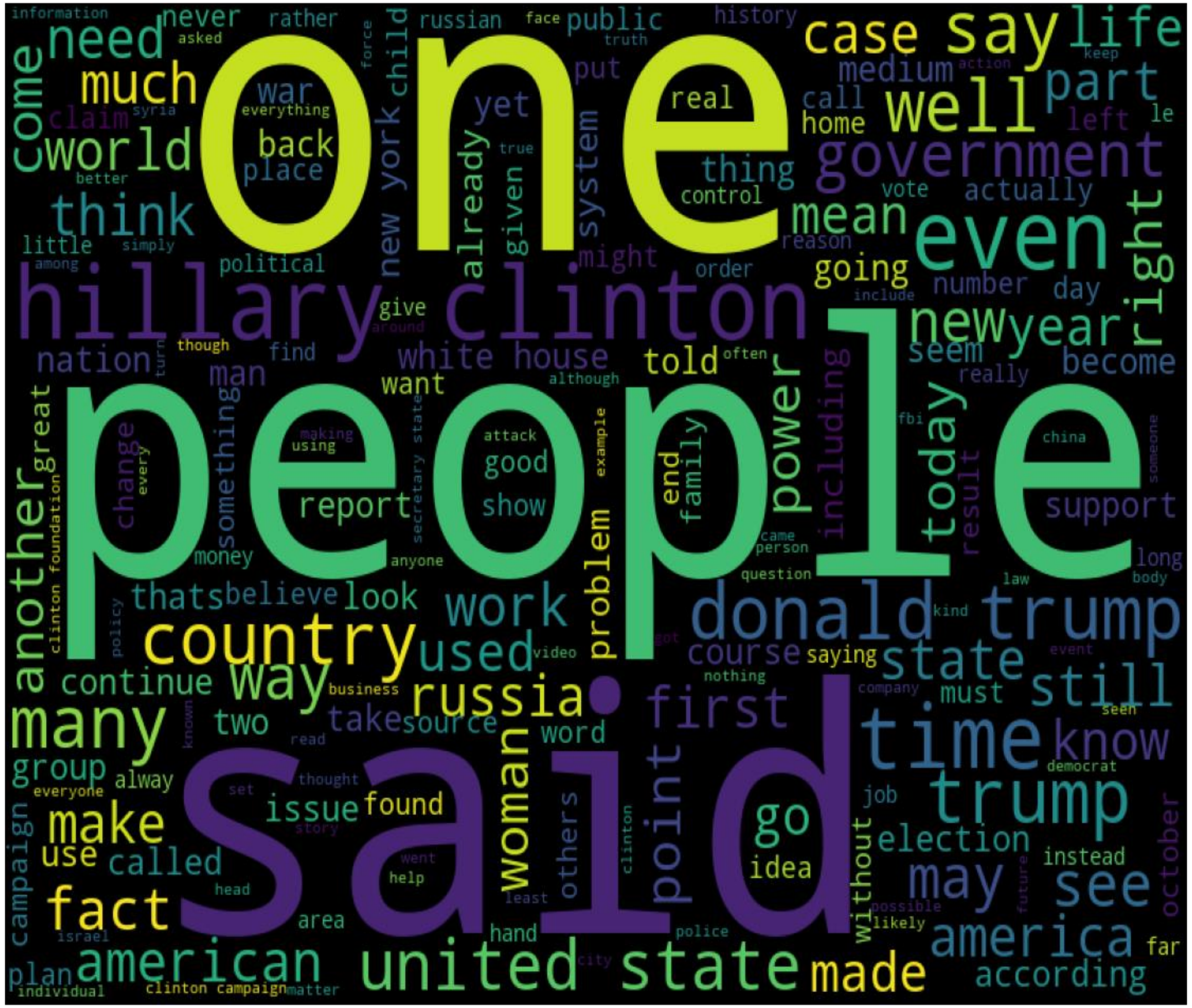


Real

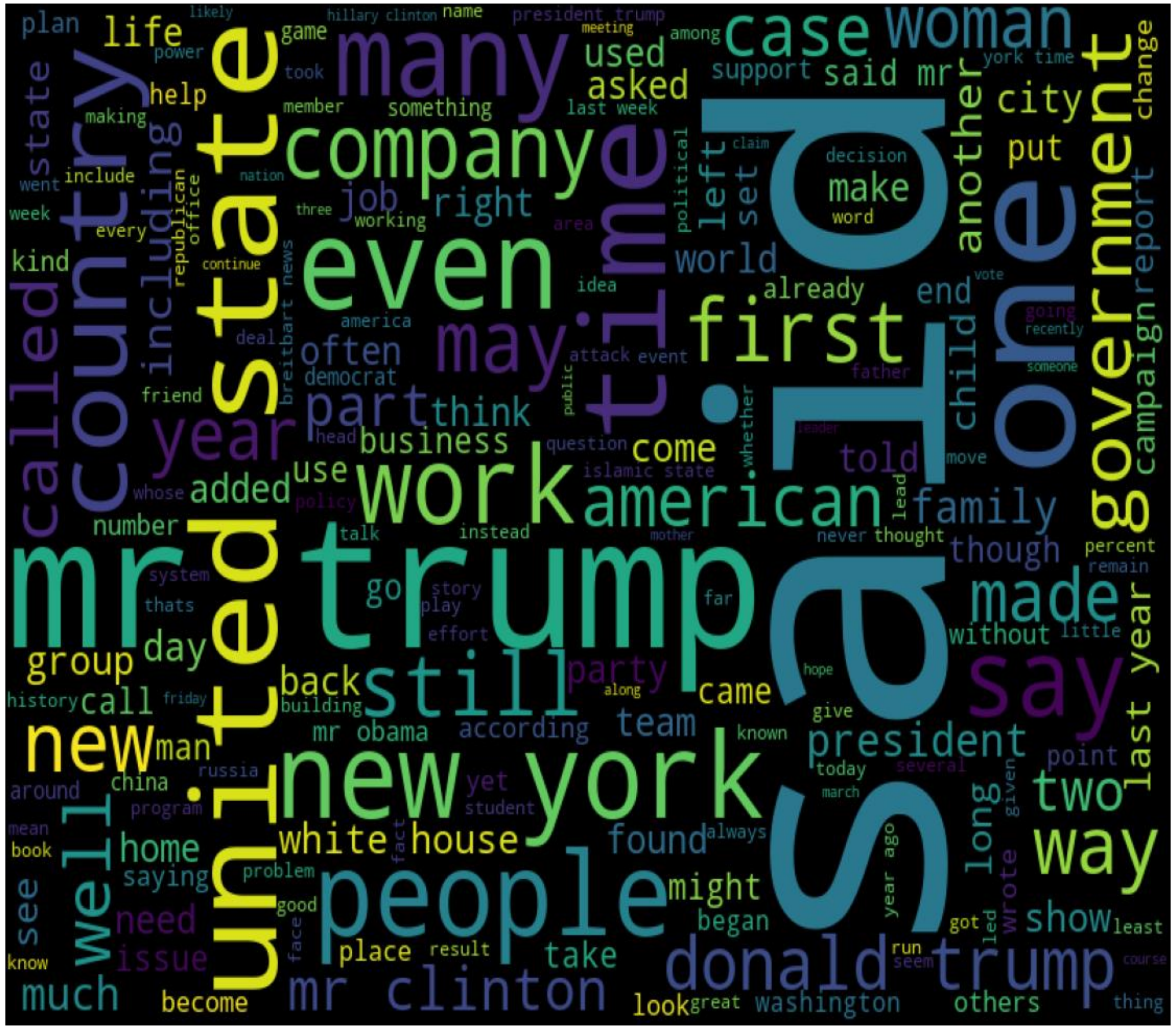


Word cloud - Kaggle

Fake

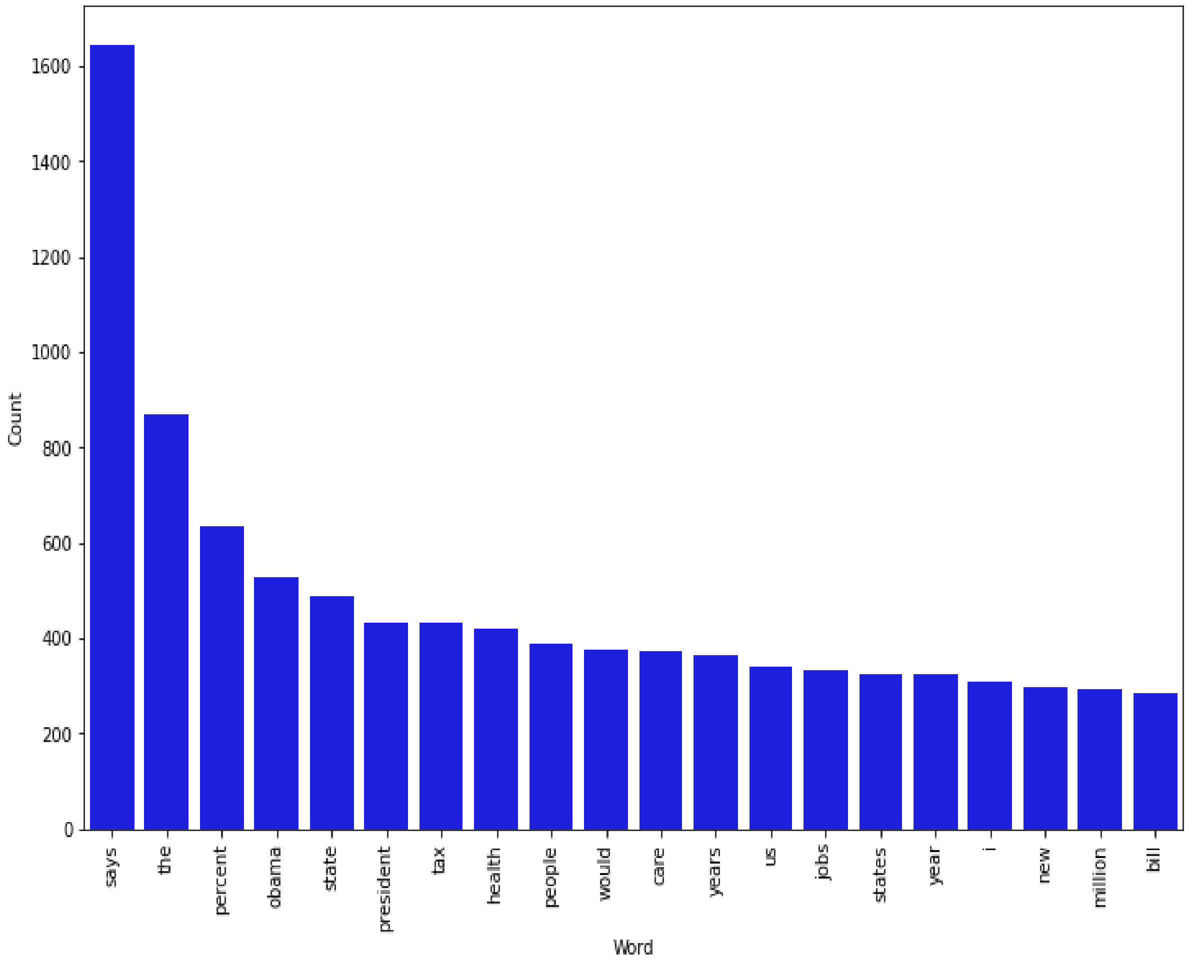


Real

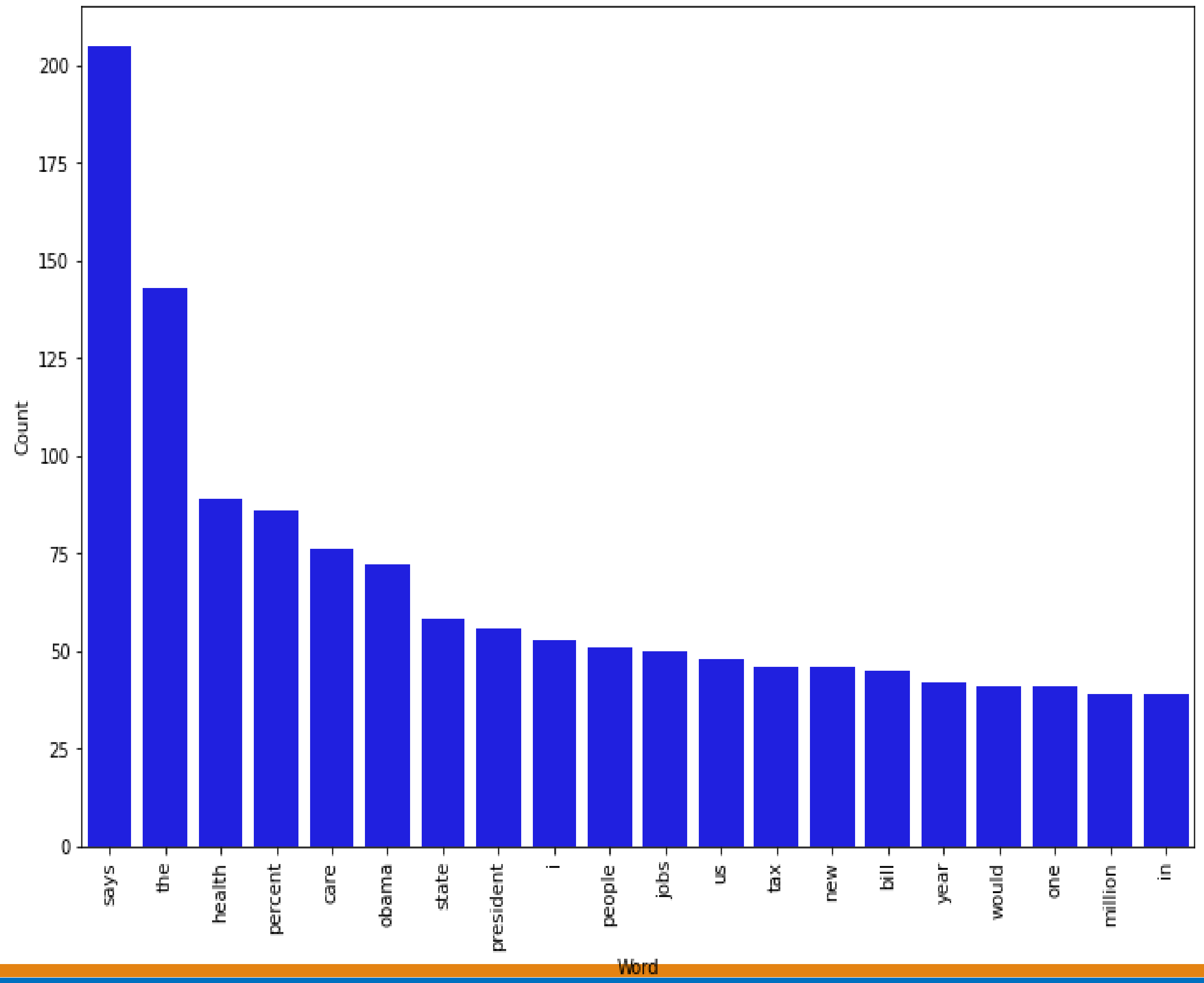


Finding of Frequent words - LIAR

Fake

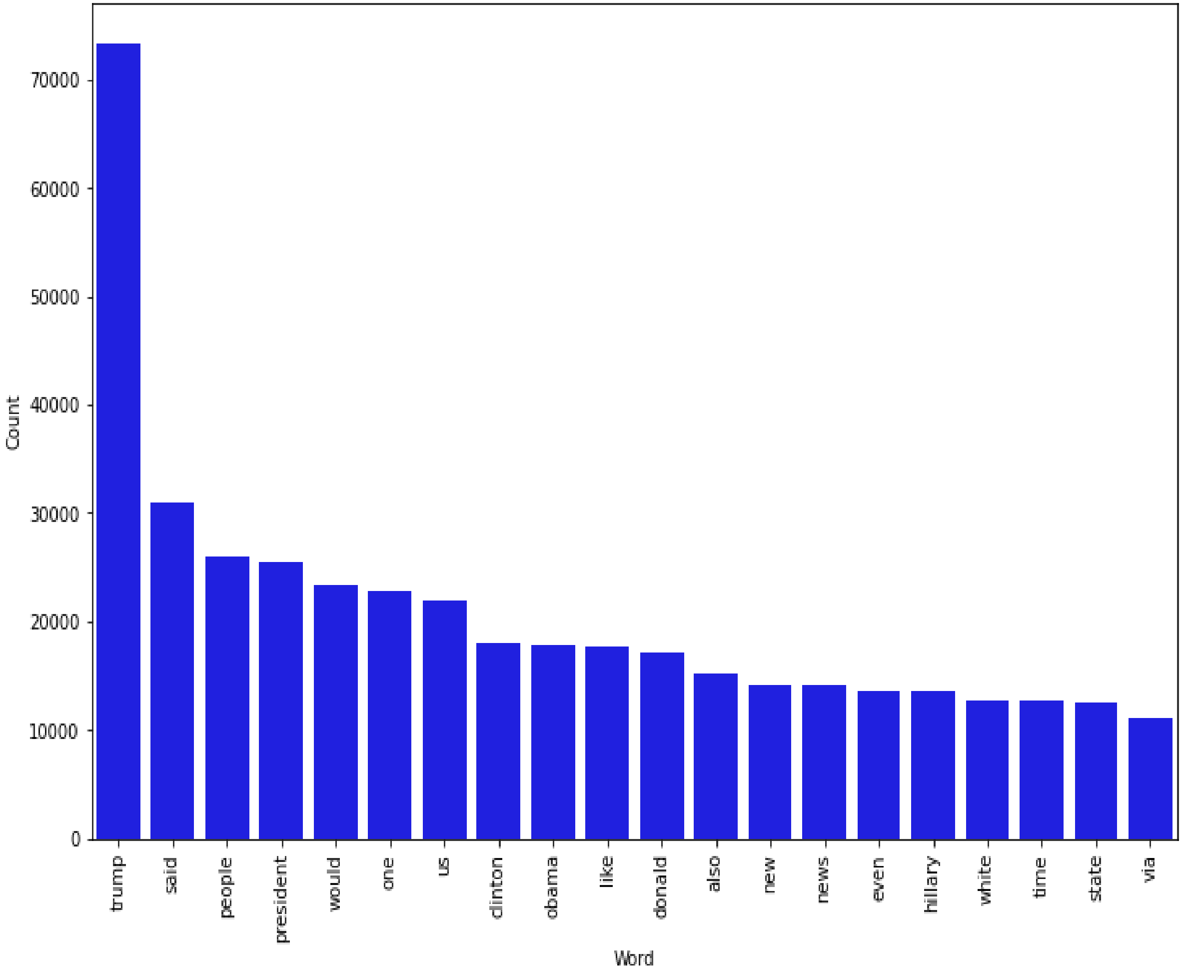


Real

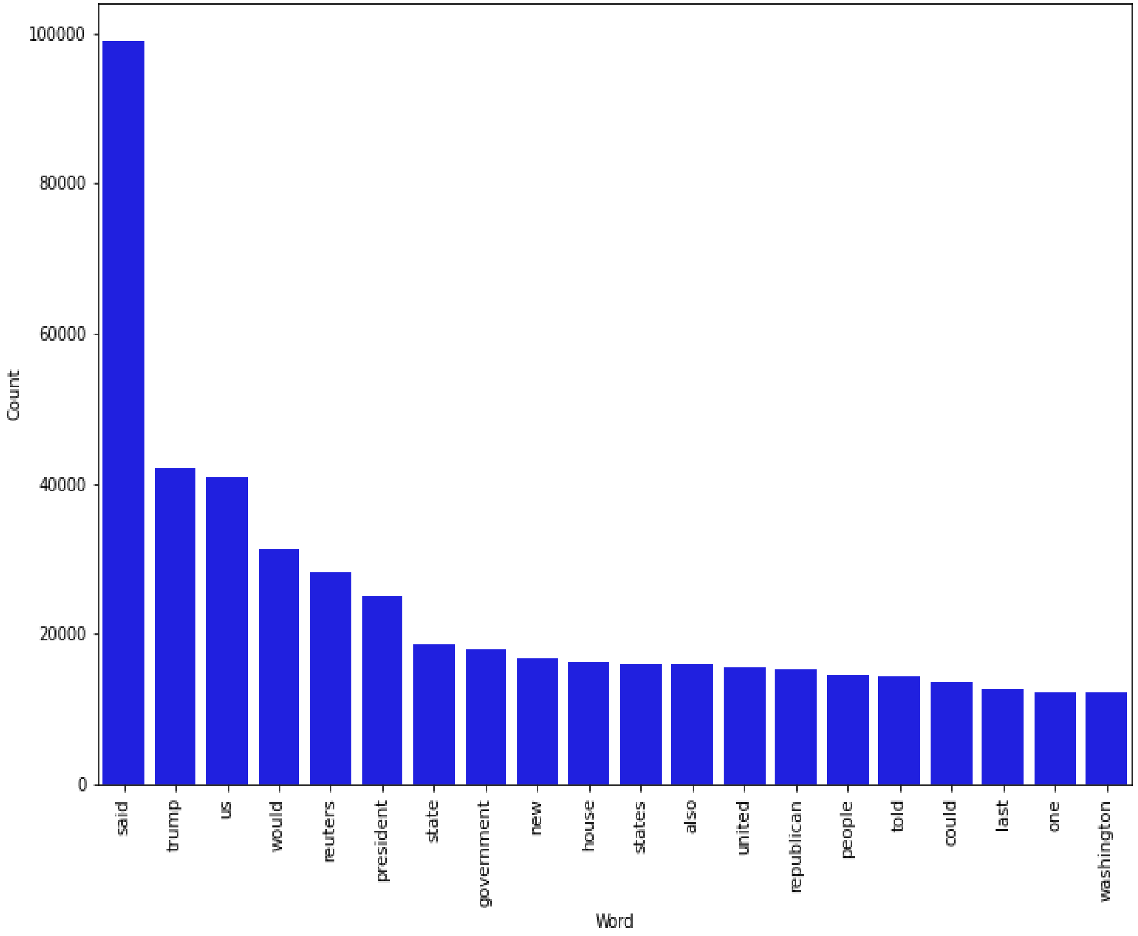


Finding of Frequent words - ISOT

Fake

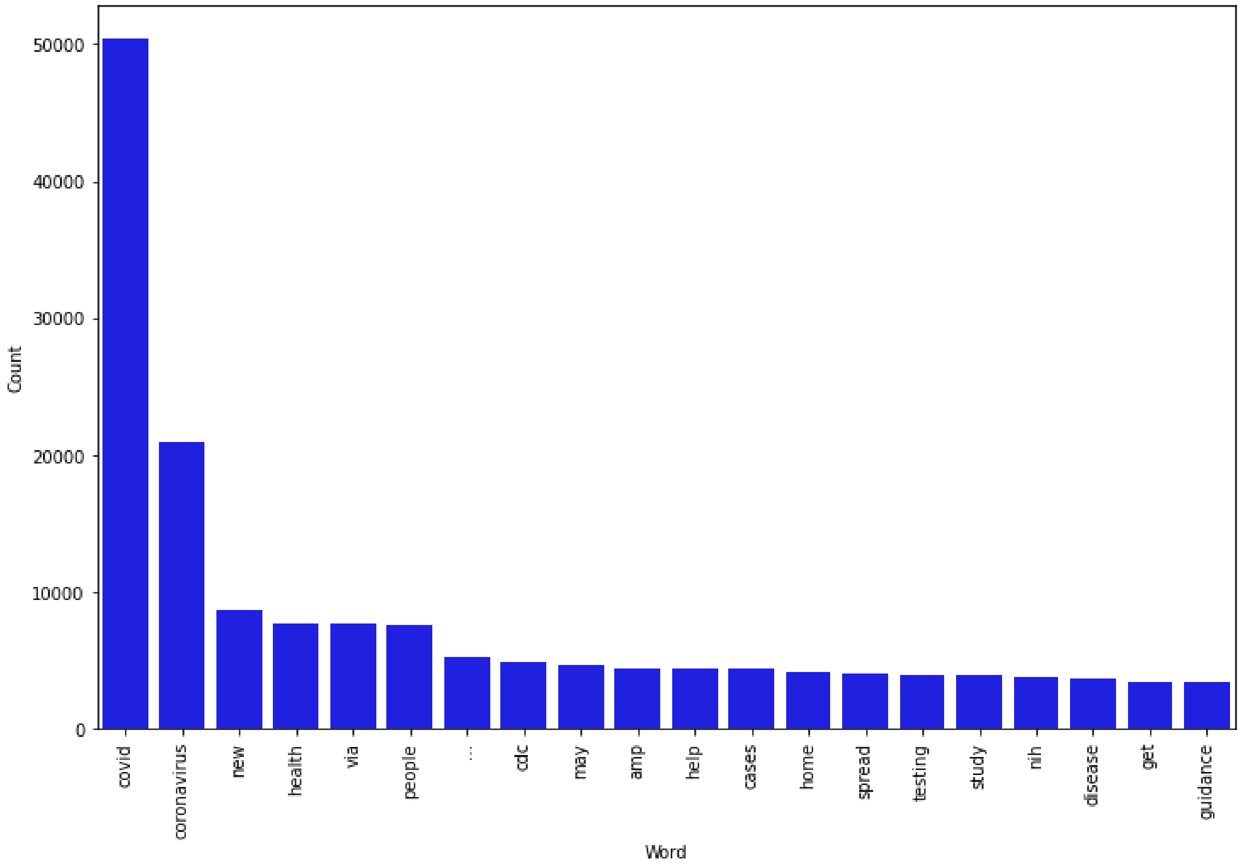


Real

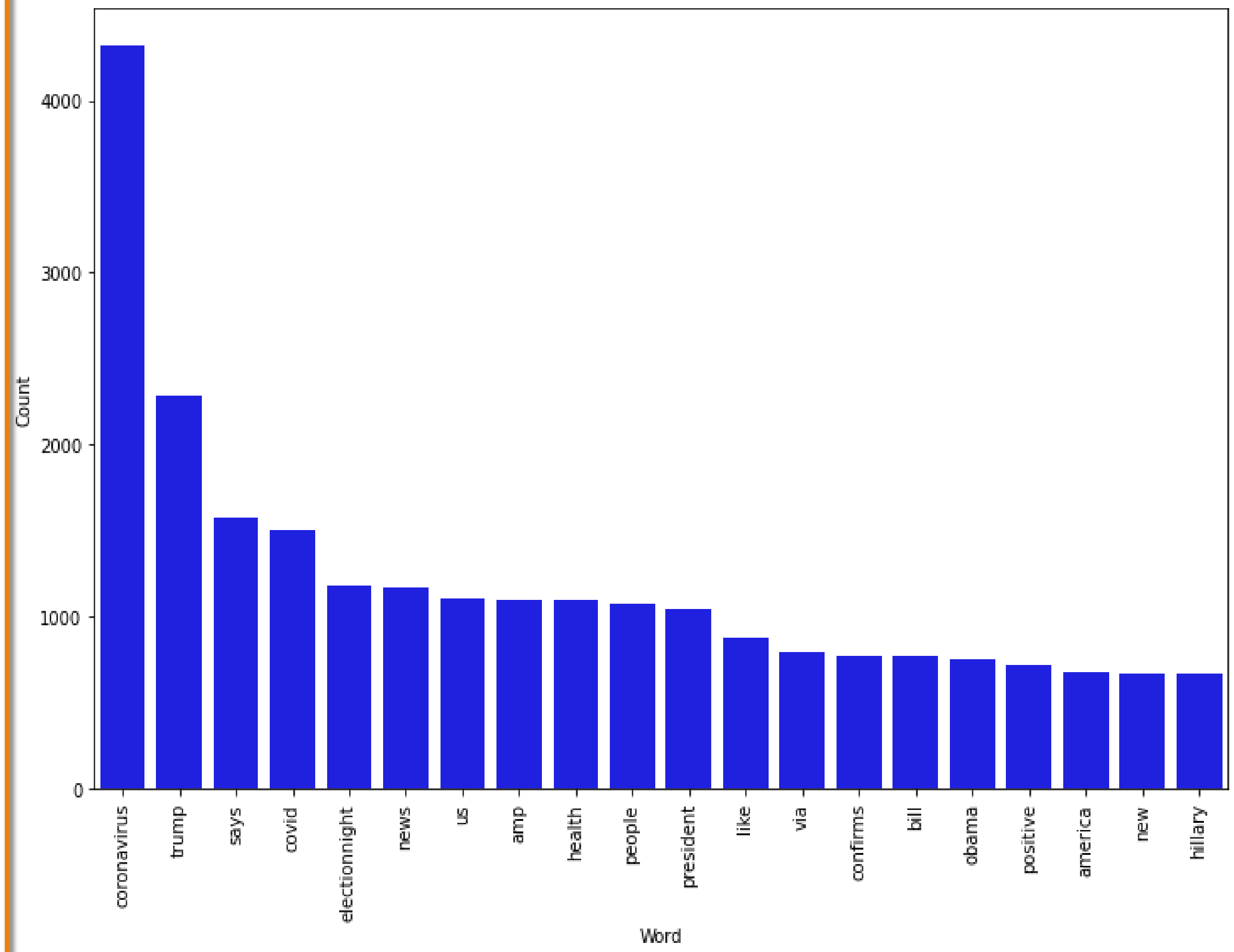


Finding of Frequent words - Twitter

Fake

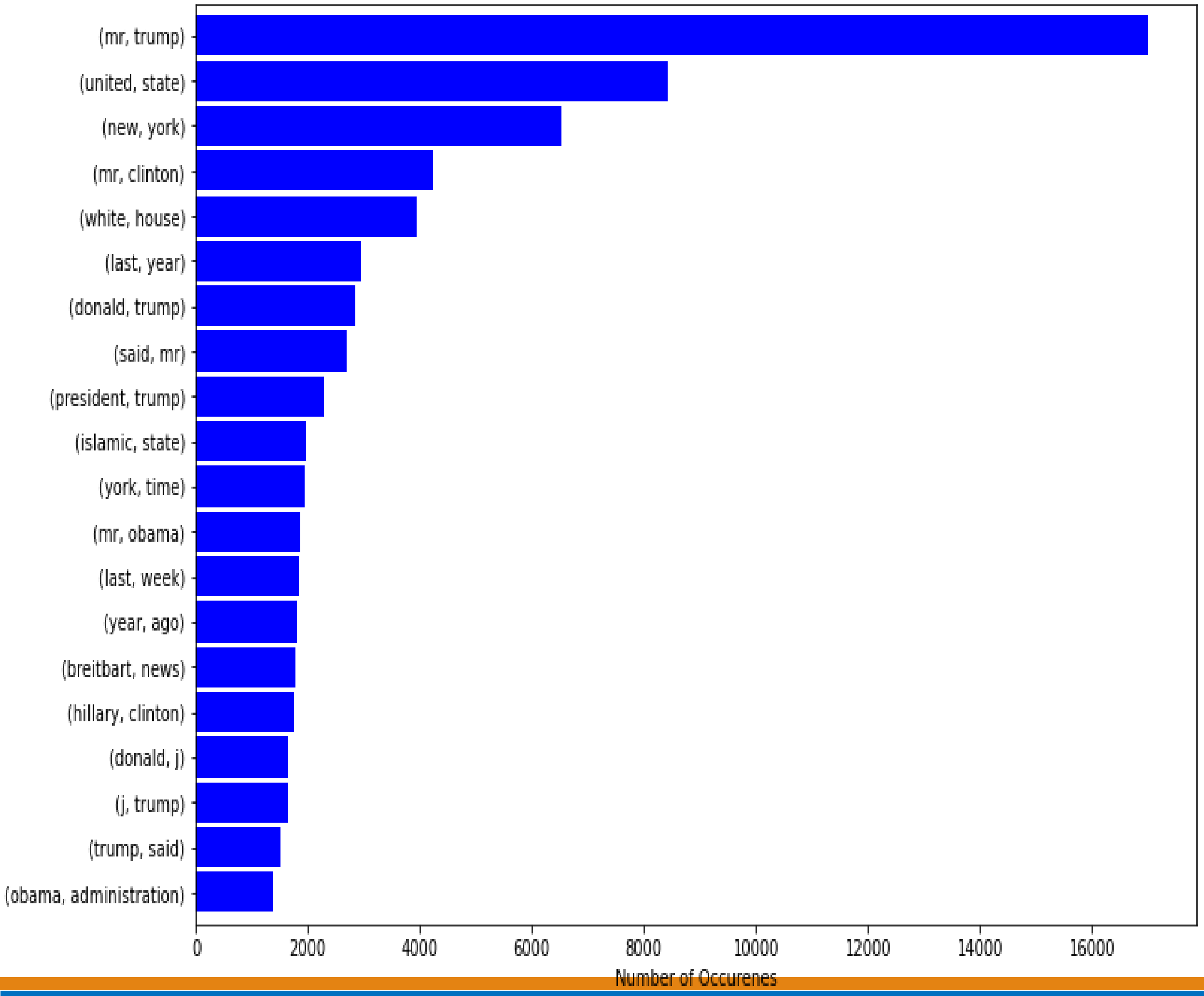


Real

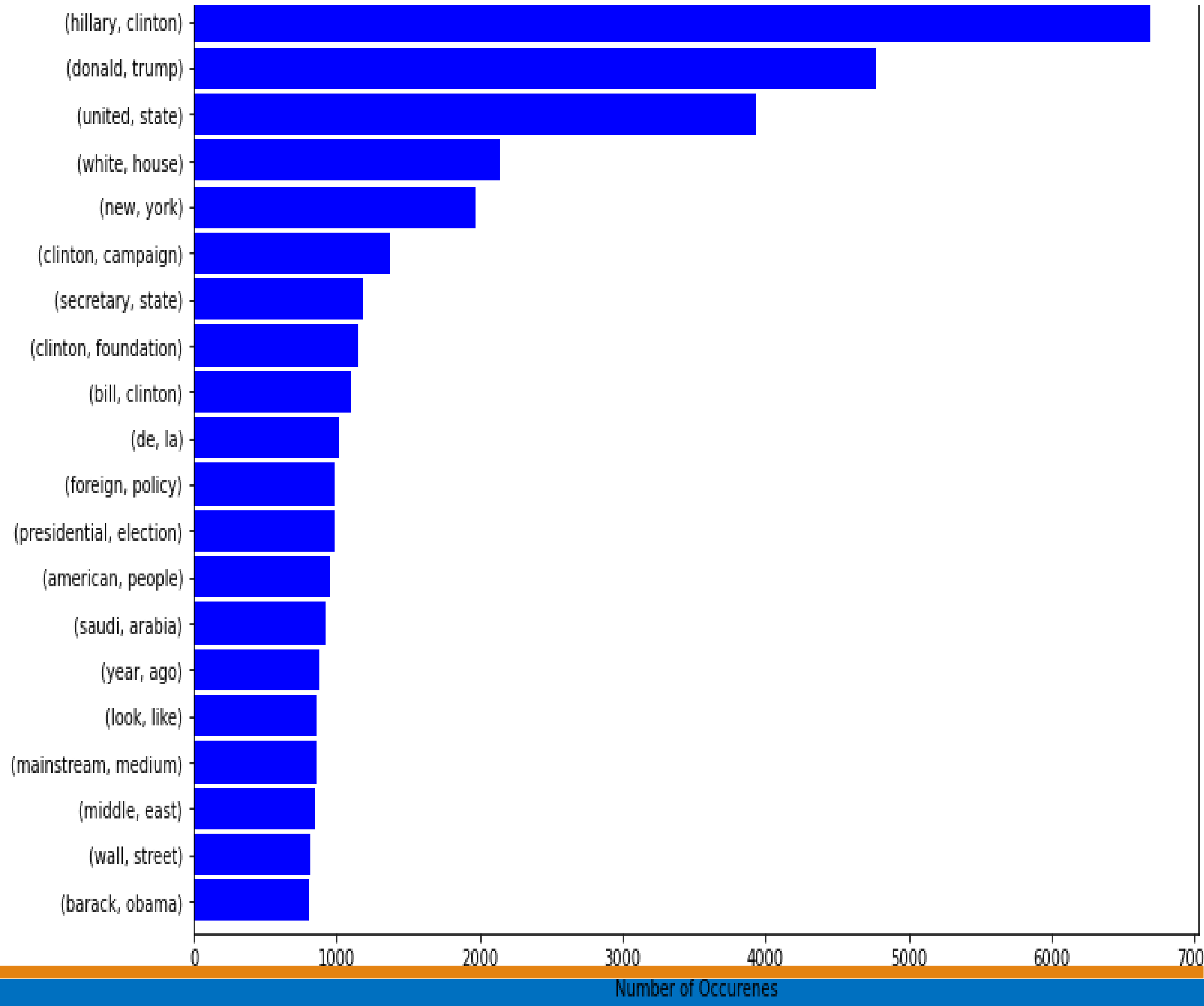


Finding of Frequent words - Kaggle

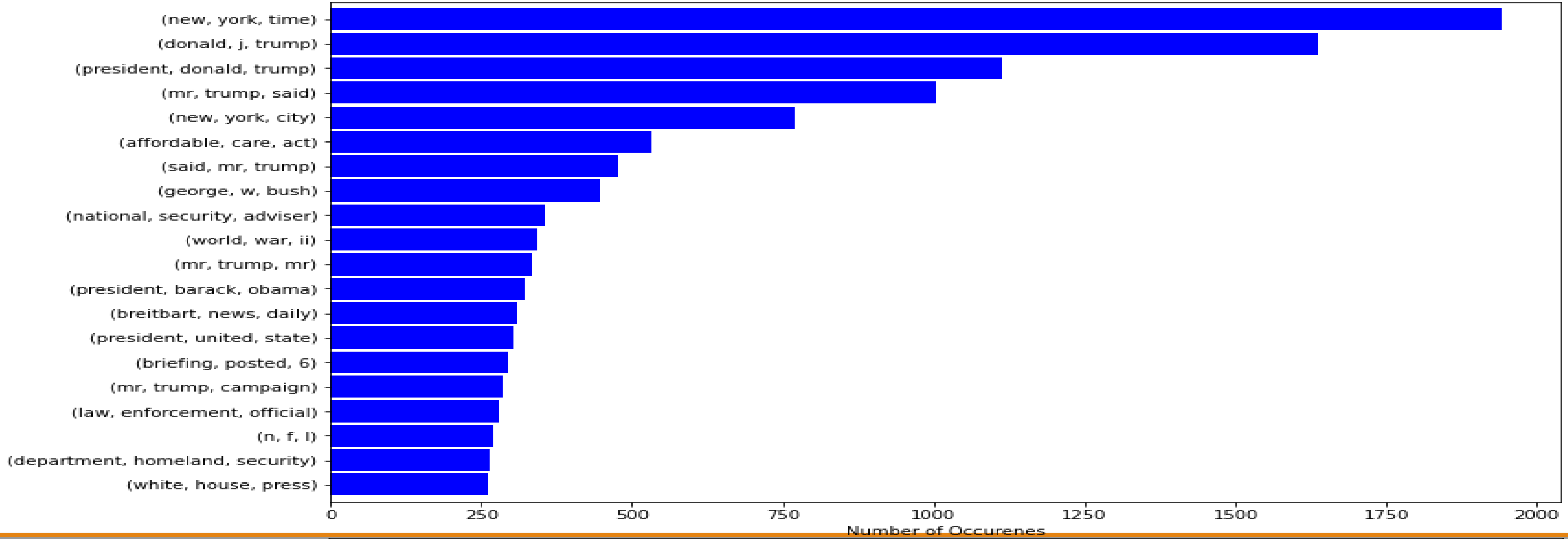
Bigram - real



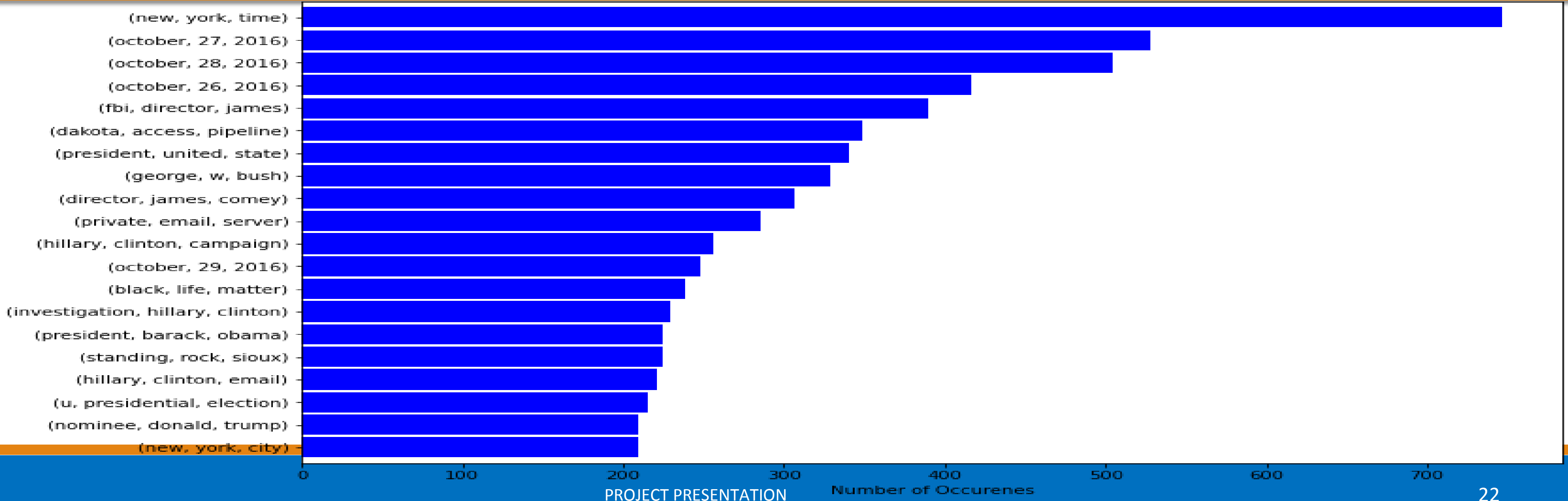
Bigram - fake



Trigram - real



Trigram - fake



Classification models

BERT

- BERT is a new language representation model released by Google.
- It works by randomly masking word tokens and representing each masked word with a vector based on its context.
- The two applications of BERT : Pre-training and Fine tuning.
  - Pre-training: Trained two unsupervised learning task; Masked language model & Next sentence prediction.
  - Fine tuning BERT: Encoding concatenated text pairs with self attention.
- BERT is basically a trained Transformer Encoder stack.
- There are two variants available: BERT Base and BERT Large.
  - BERT Base: 12 layers (transformer blocks), 12 attention heads, and 110 million parameters.
  - BERT Large: 24 layers (transformer blocks), 16 attention heads and, 340 million parameters.

DistilBERT

- DistilBERT is a small, fast, cheap and light transformer model based on the BERT architecture.
- Knowledge distillation is performed during the pre-training phase to reduce the size of a BERT model by 40%.
- It does not has token-type embedding, pooler and retains only half of the layers from Google's BERT.

```
Model: "tf_distil_bert_for_sequence_classification_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 distilbert (TFDistilBertMai  multiple                 66362880
 nLayer)

 pre_classifier (Dense)      multiple                  590592

 classifier (Dense)          multiple                  1538

 dropout_39 (Dropout)        multiple                  0

=================================================================
Total params: 66,955,010
Trainable params: 66,955,010
Non-trainable params: 0
_________________________________________________________________
None
```

- DistilBERT uses a technique called distillation, which approximates the Google's BERT, i.e. the large neural network by a smaller one.

ML Algorithms

- Naïve Bayes Classifier : One of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, it predicts on the basis of the probability of an object.
- Logistic regression : It is a supervised learning algorithm which is mostly used to solve binary "classification" tasks. The basis of logistic regression is the logistic function, also called the sigmoid function, which takes in any real valued number and maps it to a value between 0 and 1.
- Decision trees : A rule-based approach to classification and regression problems. They use the values in each feature to split the dataset to a point where all data points that have the same class are grouped together.
- Random Forest: It is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
- Support Vector Machine (SVM): It is a classification technique used for the classification of linear as well as non-linear data. SVM is the margin based classifier.

Design steps - BERT

1. Import Libraries – Transformer, numpy, pandas, matplotlib, NLTK, Word cloud, etc.
2. Load Dataset
3. Data preparation – Remove unwanted columns
4. Clean the entire dataset –Remove stop words, punctuation, convert uppercase letters into lowercase
5. Word cloud formation – Create word clod for real and fake data separately
6. Find most frequent words in the dataset- using Bigram, Trigram or simply by a bar graph
7. Split the dataset into train and test set
8. Tokenize the sentences
9. Convert tokenize dataset into torch dataset
10. Call the fine-tuned BERT model for text classification from pre-trained model - BertForSequenceClassification
11. Train the model and evaluate on test dataset
12. Save the fine-tuned model and tokenizer
13. Get prediction by adding a softmax layer for obtaining the output probabilities

Design steps - DistilBERT

1. Import Libraries – Transformer, numpy, pandas, matplotlib, NLTK, Word cloud, etc.
2. Load Dataset
3. Data preparation – Remove unwanted columns
4. Clean the entire dataset –Remove stop words, punctuation, convert uppercase letters into lowercase
5. Split the dataset into train and test set
6. Encode with DistilBERT
7. Find the maximum length
8. Tokenize the sentences
9. Convert labels and encodings to Tensorflow dataset
10. Fine tune with native Tensorflow- TFDistilBertForSequenceClassification
11. Train the model and evaluate on test dataset
12. Get prediction by adding a softmax layer for obtaining the output probabilities

Results

Experimentation using BERT in ISOT dataset

Step	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
200	0.146000	0.004917	0.999220	0.999253	0.998721	0.999787
400	0.014800	0.006016	0.999109	0.999147	0.998295	1.000000
600	0.009300	0.003091	0.999332	0.999360	0.998721	1.000000
800	0.003900	0.000694	0.999777	0.999787	0.999573	1.000000
1000	0.002100	0.001011	0.999777	0.999787	0.999573	1.000000
1200	0.000100	0.000736	0.999889	0.999893	0.999787	1.000000
1400	0.005600	0.000840	0.999777	0.999787	0.999573	1.000000
1600	0.004200	0.000091	1.000000	1.000000	1.000000	1.000000

Training

```
***** Running Evaluation *****
  Num examples = 8980
  Batch size = 20
/home/administrator/anaconda3/lib/python3.9/site-packages/
to gather along dimension 0, but all input tensors were sc
  warnings.warn('Was asked to gather along dimension 0, bu
```

```
Attempted to log scalar metric eval_loss:
9.067665814654902e-05
Attempted to log scalar metric eval_accuracy:
1.0
Attempted to log scalar metric eval_f1:
1.0
Attempted to log scalar metric eval_precision:
1.0
Attempted to log scalar metric eval_recall:
1.0
Attempted to log scalar metric eval_runtime:
56.0933
Attempted to log scalar metric eval_samples_per_second:
160.09
Attempted to log scalar metric eval_steps_per_second:
4.011
Attempted to log scalar metric epoch:
1.0
```

Testing

Results (Contd.)

Experimentation using BERT in LIAR dataset

Training

Step	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
200	0.672900	0.679516	0.644531	0.771787	0.656183	0.936834
400	0.665300	0.635513	0.640625	0.739745	0.690885	0.796043
600	0.656800	0.661908	0.641602	0.781678	0.641602	1.000000
800	0.654200	0.652114	0.641602	0.781678	0.641602	1.000000

```
Out[40]: {'eval_loss': 0.6355127692222595,
 'eval_accuracy': 0.640625,
 'eval_f1': 0.7397454031117398,
 'eval_precision': 0.6908850726552179,
 'eval_recall': 0.7960426179604262,
 'eval_runtime': 5.2521,
 'eval_samples_per_second': 389.937,
 'eval_steps_per_second': 19.611,
 'epoch': 1.0}
```

Testing

Results (Contd.)

Experimentation using BERT in Twitter dataset

Step	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
200	0.165000	0.566074	0.870904	0.906618	0.932110	0.882484
400	0.307200	0.239885	0.897515	0.926883	0.939344	0.914749
600	0.259000	0.212299	0.909824	0.936550	0.935920	0.937181
800	0.249300	0.200490	0.911169	0.938490	0.923199	0.954297
1000	0.233900	0.229428	0.912695	0.939261	0.928206	0.950583

…………………………………………………………………………………………………

…………………………………………………………………………………………………

6600	0.144100	0.136385	0.946624	0.962404	0.962755	0.962054
6800	0.142900	0.138613	0.947684	0.963349	0.958539	0.968208
7000	0.138800	0.141027	0.948046	0.963365	0.964791	0.961945
7200	0.138400	0.137223	0.948072	0.963426	0.963742	0.963110
7400	0.150100	0.132515	0.948589	0.963967	0.959584	0.968390
7600	0.133100	0.133747	0.948124	0.963421	0.964828	0.962017

Training

```
Out[48]: {'eval_loss': 0.13251514732837677,
          'eval_accuracy': 0.9485893092658202,
          'eval_f1': 0.963967229754223l,
          'eval_precision': 0.9595842956120092,
          'eval_recall': 0.9683903860160233,
          'eval_runtime': 98.2727,
          'eval_samples_per_second': 393.487,
          'eval_steps_per_second': 9.84,
          'epoch': 1.0}
```

Testing

Results (Contd.)

Experimentation using BERT in Kaggle dataset

Training

Step	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
200	0.010400	0.029362	0.996172	0.995628	0.997497	0.993766
400	0.039100	0.118688	0.973476	0.968840	0.999337	0.940150
600	0.067200	0.020631	0.996445	0.995944	0.996877	0.995012
800	0.021100	0.015175	0.996992	0.996581	0.993800	0.999377
1000	0.006400	0.014516	0.998086	0.997819	0.997508	0.998130
1200	0.029000	0.009180	0.998086	0.997819	0.997508	0.998130
1400	0.014100	0.005111	0.998906	0.998755	0.997512	1.000000

```
{'epoch': 1.0,
 'eval_accuracy': 0.9989062072737216,
 'eval_f1': 0.9987546699875467,
 'eval_loss': 0.005110885016620159,
 'eval_precision': 0.9975124378109452,
 'eval_recall': 1.0,
 'eval_runtime': 241.0123,
 'eval_samples_per_second': 15.174,
 'eval_steps_per_second': 0.759}
```

Testing

Results (Contd.)

Experimentation using DistilBERT in ISOT dataset

Training

```
Epoch 1/10
1965/1965 [==============================] - 625s 314ms/step - loss: 0.0196 - accuracy: 0.9938
Epoch 2/10
1965/1965 [==============================] - 617s 314ms/step - loss: 0.0038 - accuracy: 0.9991
Epoch 3/10
1965/1965 [==============================] - 618s 314ms/step - loss: 0.0011 - accuracy: 0.9997
Epoch 4/10
 155/1965 [=>............................] - ETA: 9:29 - loss: 1.3566e-04 - accuracy: 1.0000
```

Testing

```
In [32]: #Model Evaluation
         model.evaluate(test_dataset.shuffle(len(X_test)).batch(BATCH_SIZE), return_dict=True, batch_size=BATCH_SIZE)

         842/842 [==============================] - 96s 112ms/step - loss: 3.2334e-04 - accuracy: 0.9999

Out[32]: {'loss': 0.00032334256684407592, 'accuracy': 0.9998515248298645}
```

Results (Contd.)

Experimentation using DistilBERT in LIAR dataset

Training

```
All model checkpoint layers were used when initializing TFDistilBertForSequenceClassification.

All the layers of TFDistilBertForSequenceClassification were initialized from the model checkpoin
If your task is similar to the task the model of the checkpoint was trained on, you can already us
Epoch 1/10
1726/1726 [==============================] - 313s 175ms/step - loss: 0.6538 - accuracy: 0.6411
Epoch 2/10
1726/1726 [==============================] - 302s 175ms/step - loss: 0.6205 - accuracy: 0.6589
Epoch 3/10
1726/1726 [==============================] - 302s 175ms/step - loss: 0.5426 - accuracy: 0.7386
Epoch 4/10
1726/1726 [==============================] - 302s 175ms/step - loss: 0.3891 - accuracy: 0.8330
Epoch 5/10
1726/1726 [==============================] - 302s 175ms/step - loss: 0.2744 - accuracy: 0.8988
Epoch 6/10
1726/1726 [==============================] - 302s 175ms/step - loss: 0.2042 - accuracy: 0.9253
Epoch 7/10
1726/1726 [==============================] - 302s 175ms/step - loss: 0.1734 - accuracy: 0.9341
Epoch 8/10
1726/1726 [==============================] - 302s 175ms/step - loss: 0.1549 - accuracy: 0.9390
Epoch 9/10
1726/1726 [==============================] - 302s 175ms/step - loss: 0.1504 - accuracy: 0.9405
Epoch 10/10
1726/1726 [==============================] - 301s 175ms/step - loss: 0.1224 - accuracy: 0.9523
<keras.callbacks.History at 0x7ff520a7f610>
```

```
[ ] #Model Evaluation
    model.evaluate(test_dataset.shuffle(len(X_test)).batch(BATCH_SIZE), return_dict=True, batch_size=BATCH_SIZE)

    1151/1151 [==============================] - 100s 86ms/step - loss: 1.6099 - accuracy: 0.6018
    {'accuracy': 0.60178142786026, 'loss': 1.6099258661270142}
```

Testing

Results (Contd.)

Experimentation using DistilBERT in Twitter dataset

Training

```
Downloading: 100% 256M/256M [00:07<00:00, 36.3MB/s]
All model checkpoint layers were used when initializing TFDistilBertForSequenceClassification.

All the layers of TFDistilBertForSequenceClassification were initialized from the model checkpoin
If your task is similar to the task the model of the checkpoint was trained on, you can already u
Epoch 1/3
6066/6066 [==============================] - 2571s 421ms/step - loss: 0.1220 - accuracy: 0.9529
Epoch 2/3
6066/6066 [==============================] - 2557s 422ms/step - loss: 0.0666 - accuracy: 0.9748
Epoch 3/3
6066/6066 [==============================] - 2556s 421ms/step - loss: 0.0423 - accuracy: 0.9845
<keras.callbacks.History at 0x7feb9e2067d0>
```

Testing

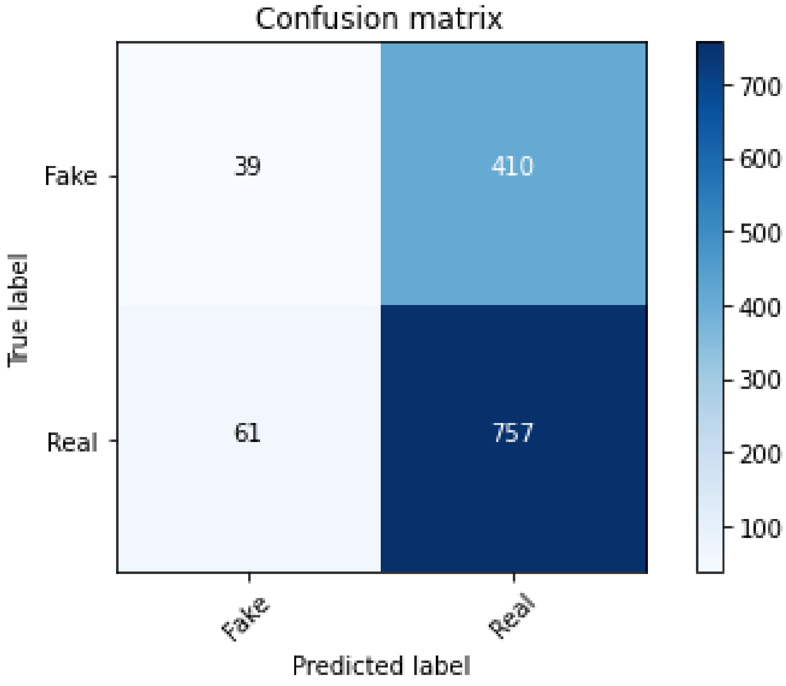
```
[ ] #Model Evaluation
    model.evaluate(test_dataset.shuffle(len(X_test)).batch(BATCH_SIZE), return_dict=True, batch_size=BATCH_SIZE)

4044/4044 [==============================] - 790s 195ms/step - loss: 0.0989 - accuracy: 0.9674
{'accuracy': 0.9674023985862732, 'loss': 0.09888310730457306}
```

Experiment on LIAR dataset

Naïve Bayes:

accuracy: 62.83%

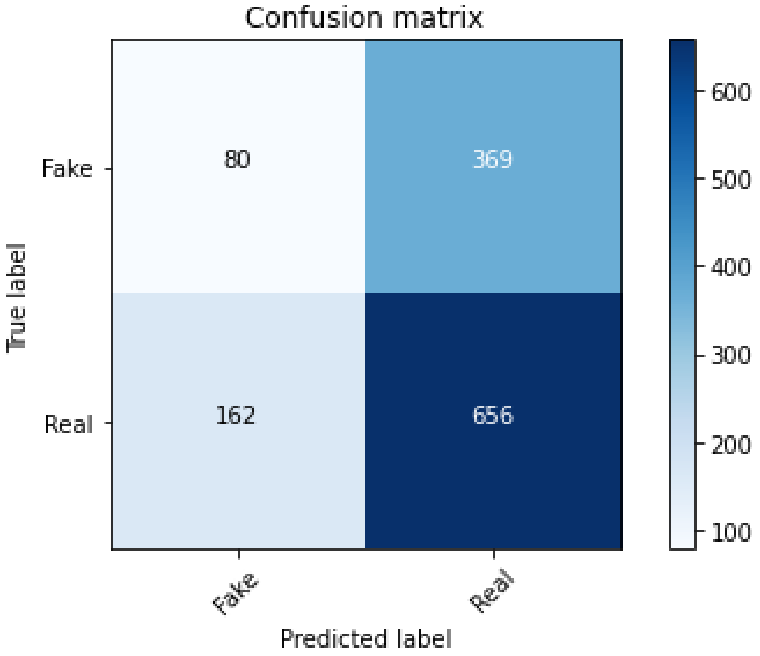
Confusion matrix, without normalization



Logistic Regression:

accuracy: 58.09%

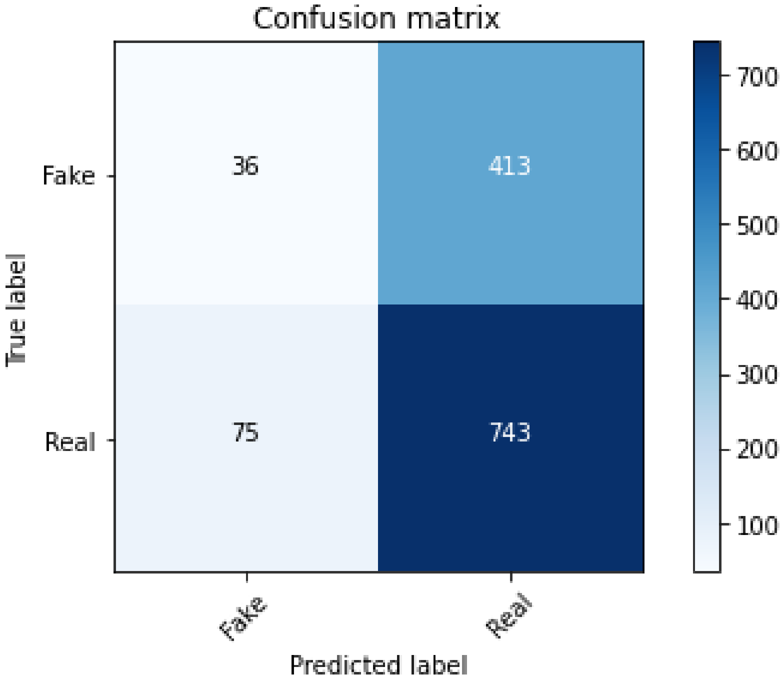
Confusion matrix, without normalization



Decision Tree:

accuracy: 61.48%

Confusion matrix, without normalization

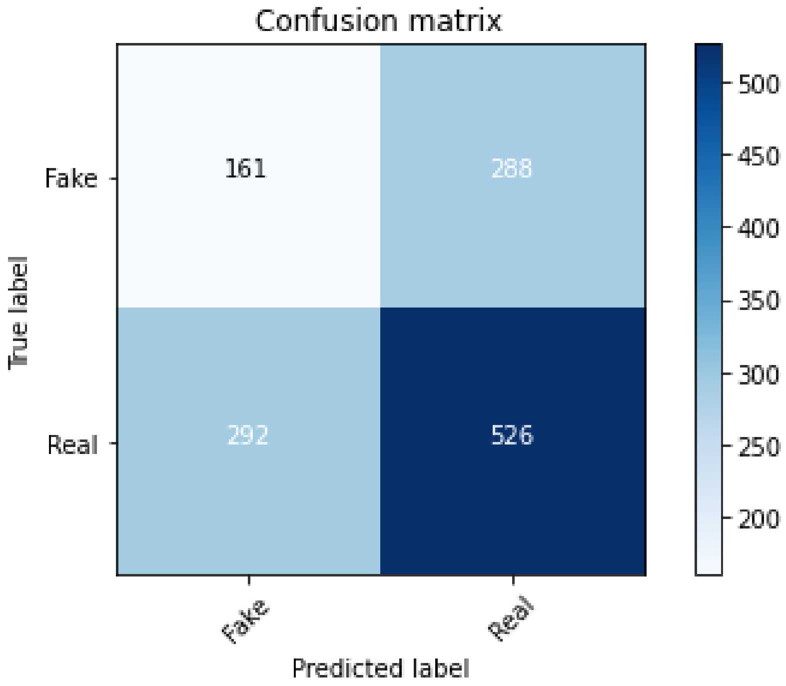


Experiment on LIAR dataset

Random Forest:

accuracy: 54.22%

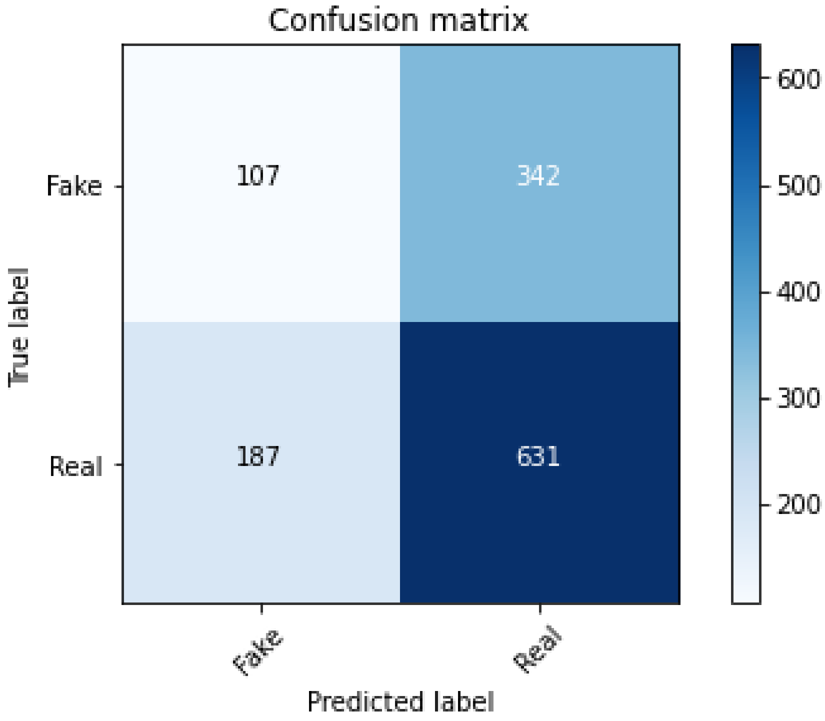
Confusion matrix, without normalization



SVM:

accuracy: 58.25%

Confusion matrix, without normalization

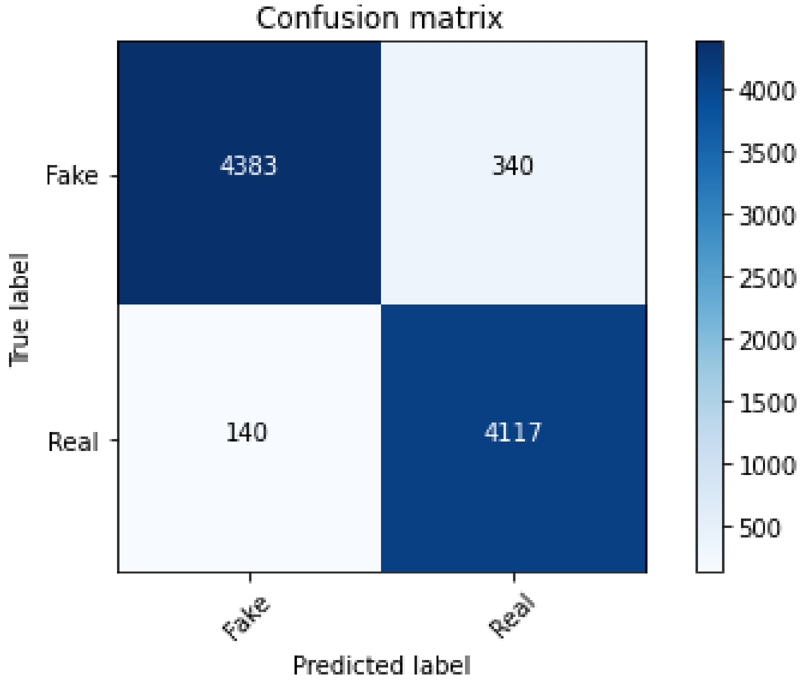


Experiment on ISOT dataset

Naïve Bayes:

```
accuracy: 94.65%
```

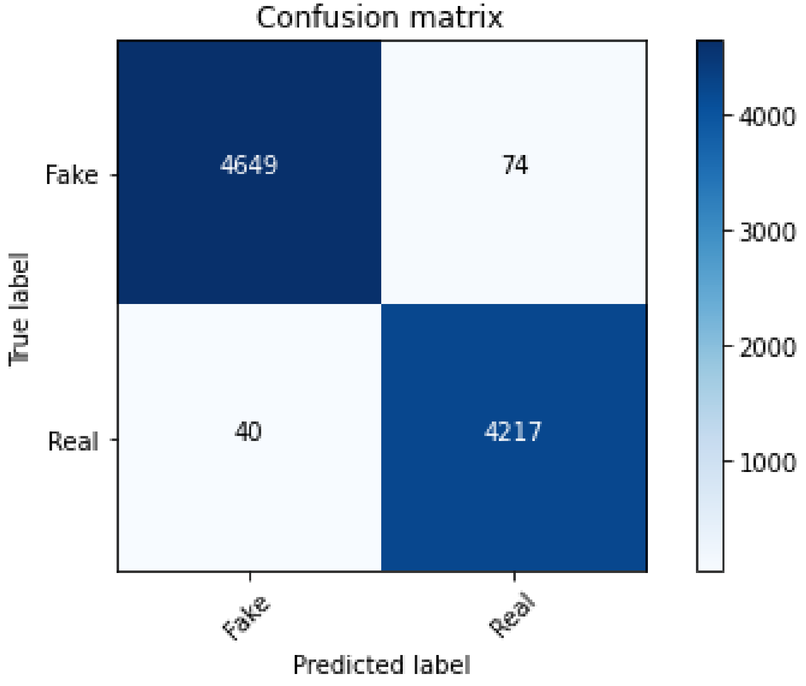
Confusion matrix, without normalization



Logistic Regression:

```
accuracy: 98.73%
```

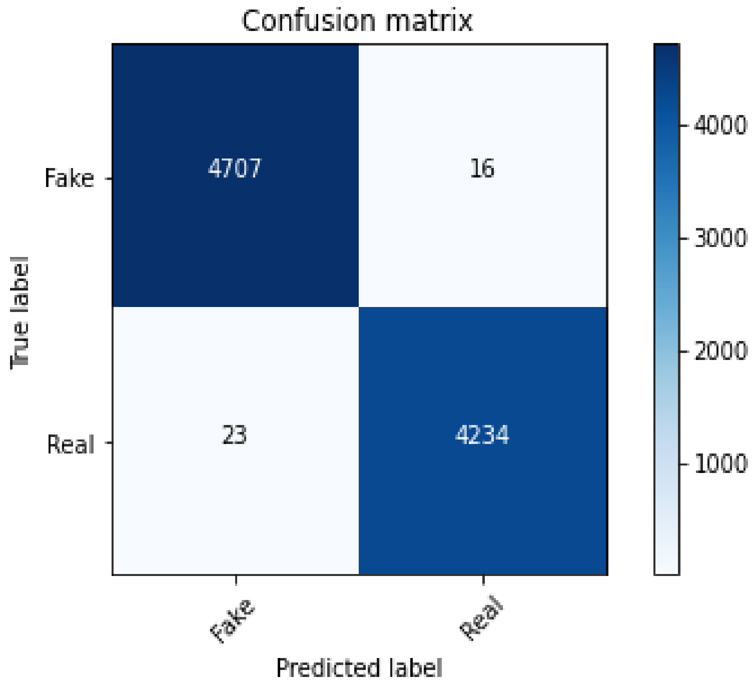
Confusion matrix, without normalization



Decision Tree:

```
accuracy: 99.57%
```

Confusion matrix, without normalization

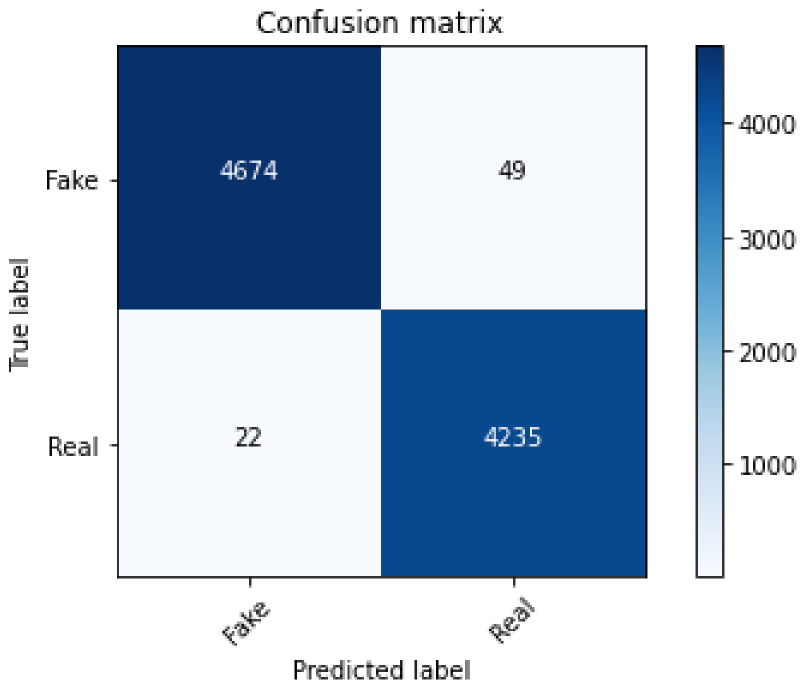


Experiment on ISOT dataset

Random Forest:

accuracy: 99.21%

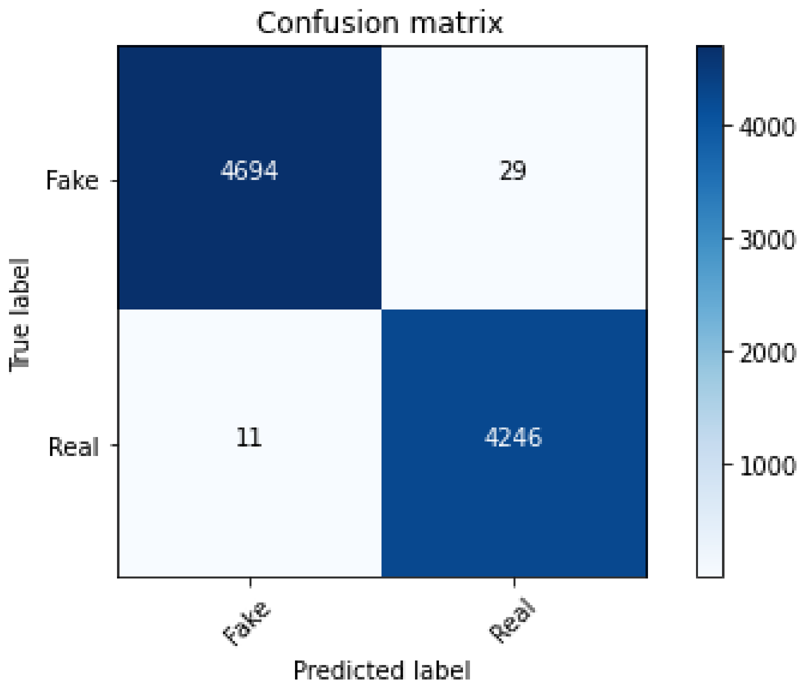
Confusion matrix, without normalization



SVM:

accuracy: 99.55%

Confusion matrix, without normalization

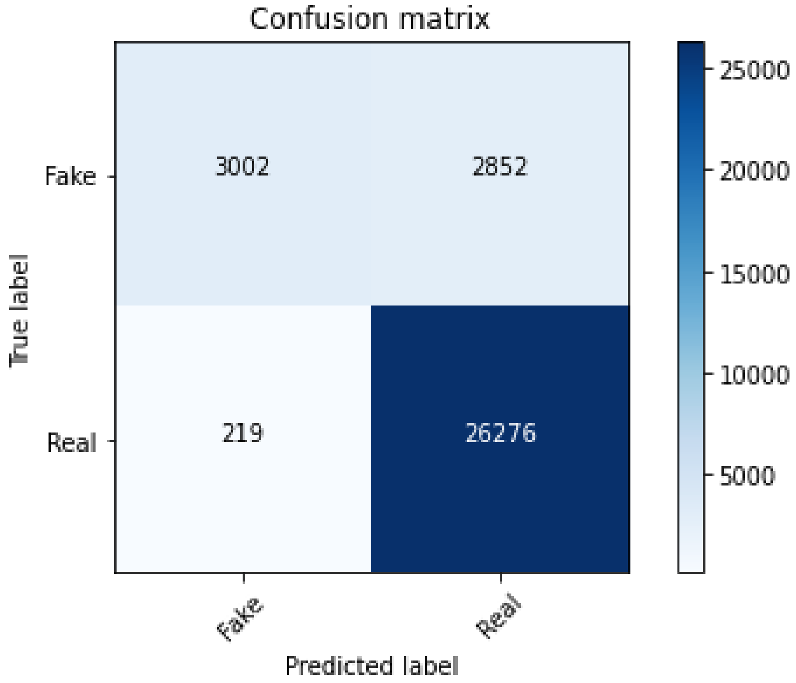


Experiment on Twitter dataset

Naïve Bayes:

accuracy: 90.51%

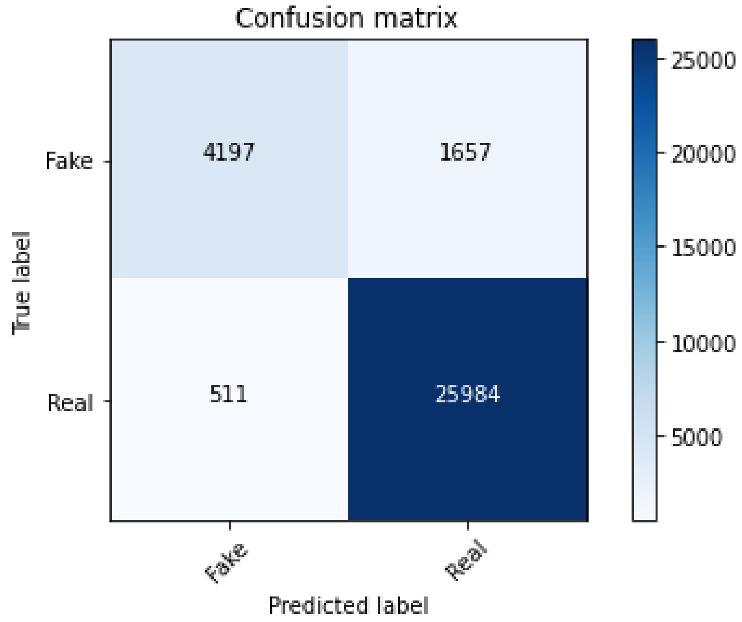
Confusion matrix, without normalization



Logistic Regression:

accuracy: 93.3%

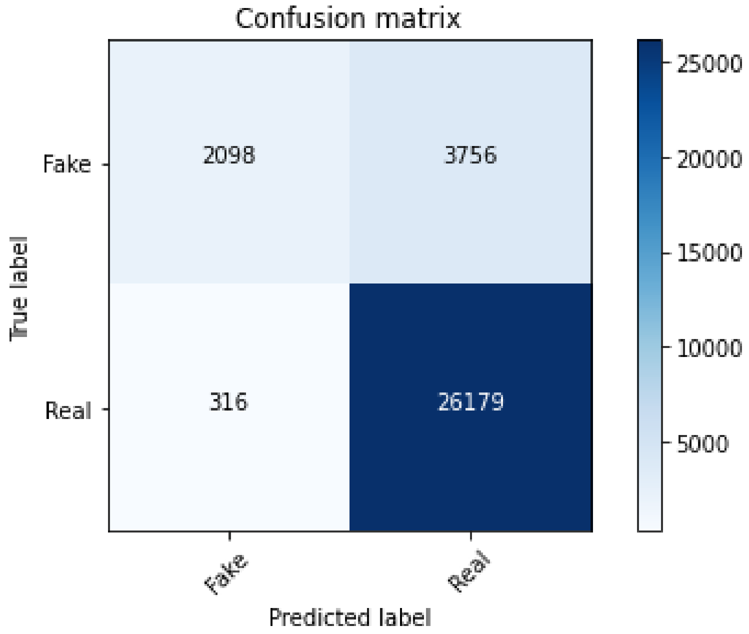
Confusion matrix, without normalization



Decision Tree:

accuracy: 87.41%

Confusion matrix, without normalization

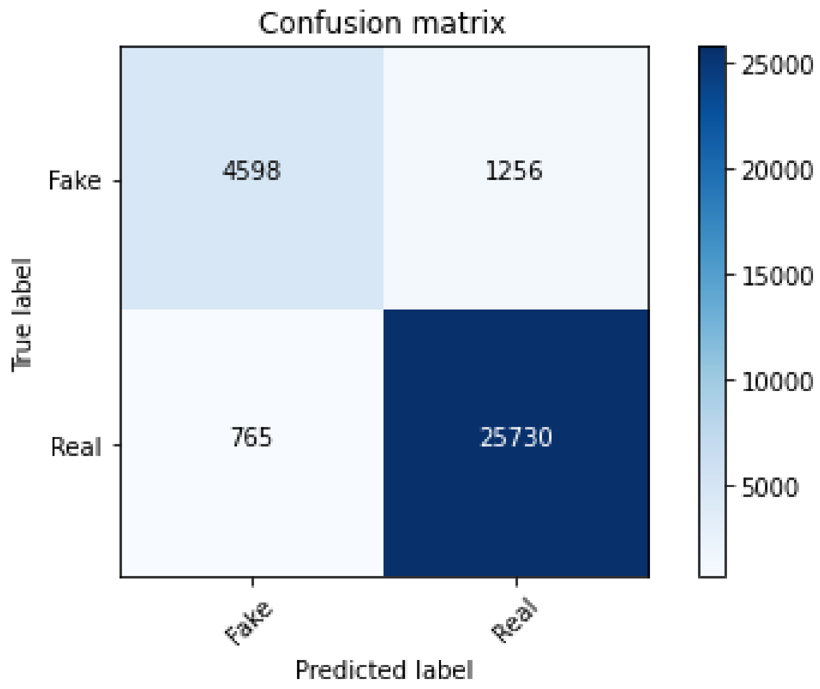


Experiment on Twitter dataset

Random Forest:

accuracy: 93.75%

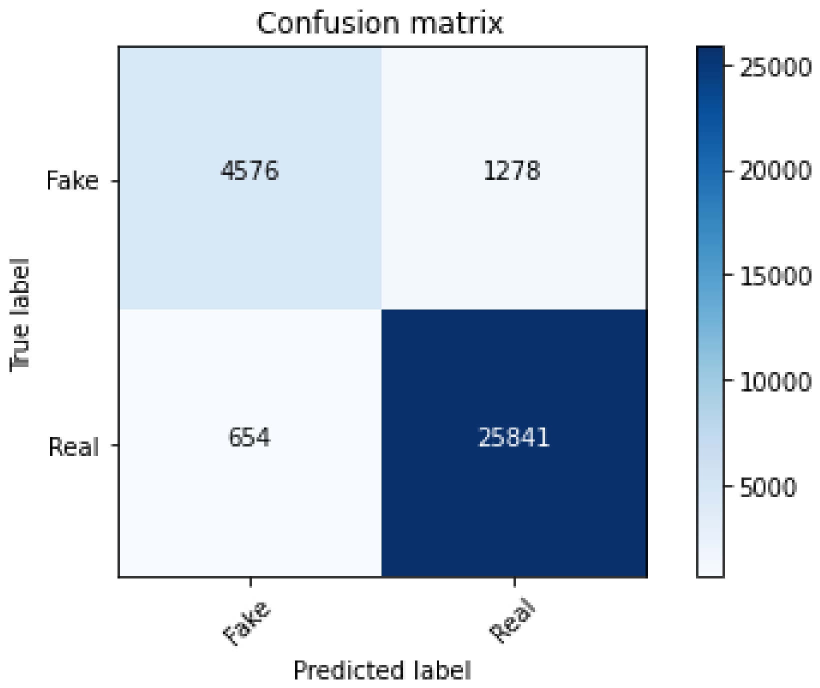
Confusion matrix, without normalization



SVM:

accuracy: 94.03%

Confusion matrix, without normalization

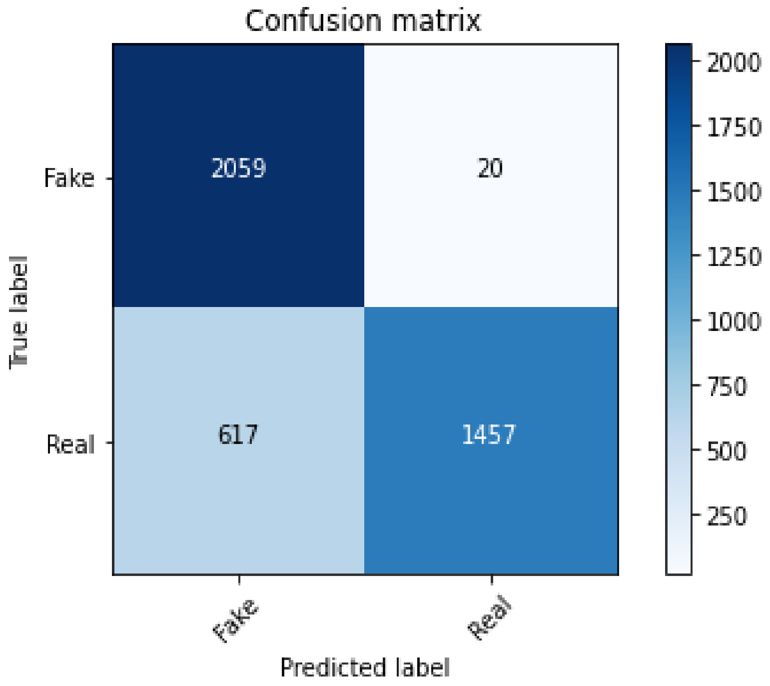


Experiment on Kaggle dataset

Naïve Bayes:

accuracy: 84.66%

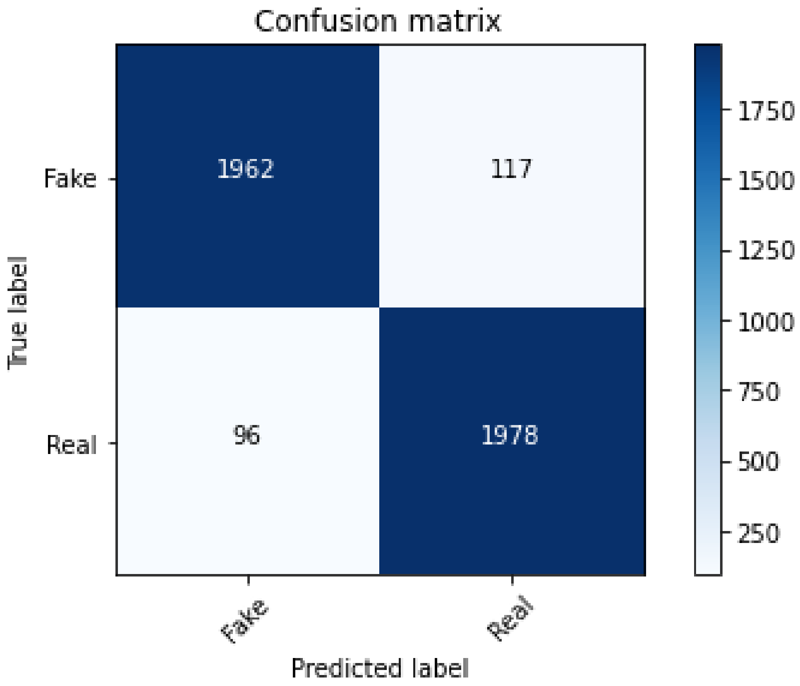
Confusion matrix, without normalization



Logistic Regression:

accuracy: 94.87%

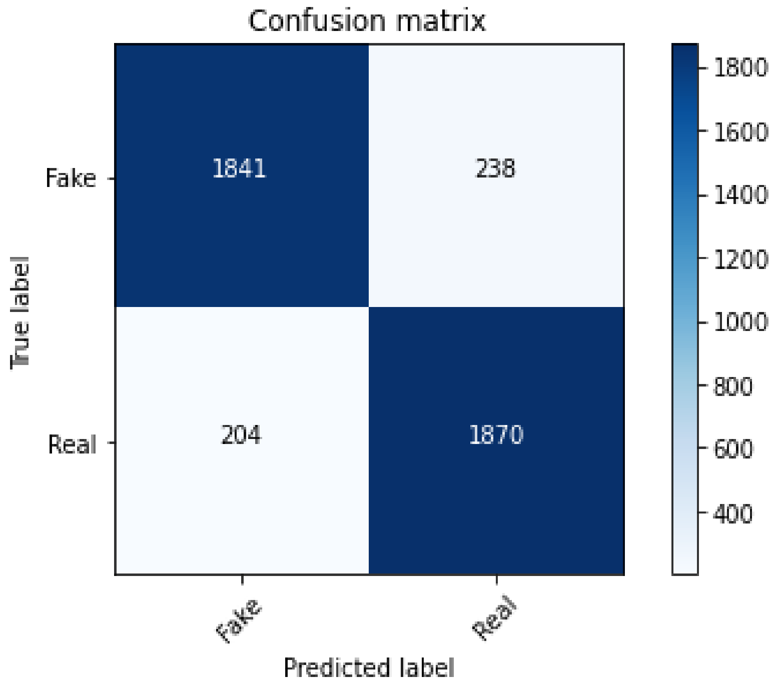
Confusion matrix, without normalization



Decision Tree:

accuracy: 89.36%

Confusion matrix, without normalization

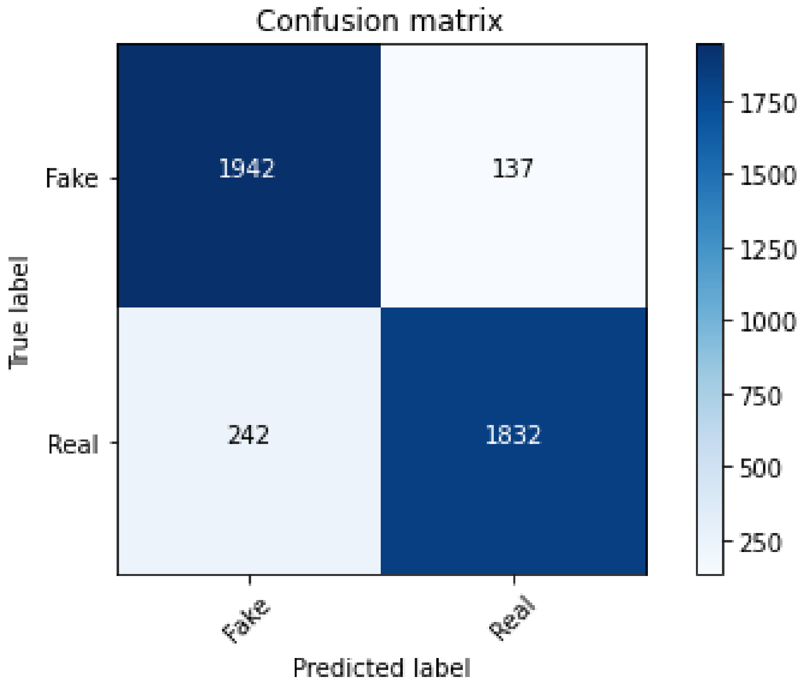


Experiment on Kaggle dataset

Random Forest:

```
accuracy: 90.87%
```

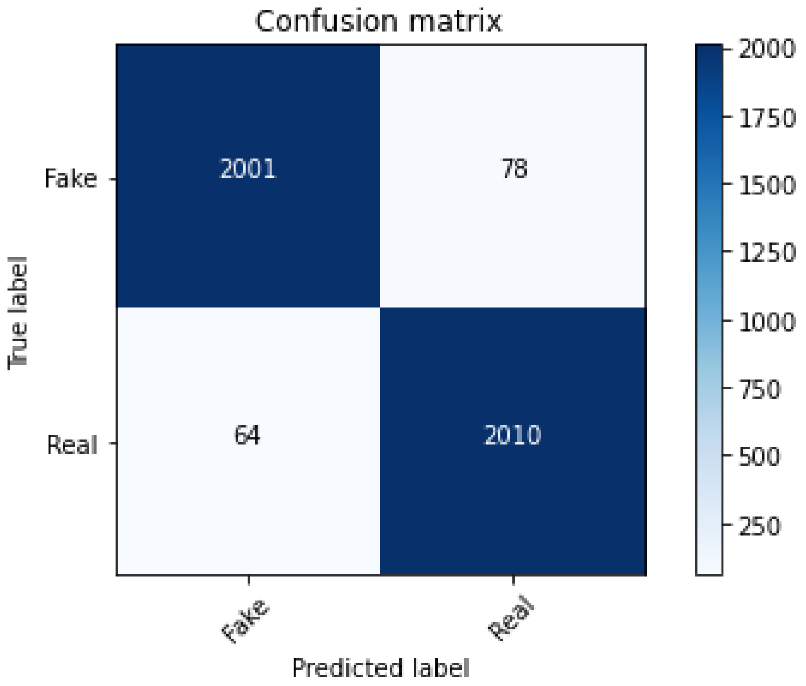
```
Confusion matrix, without normalization
```



SVM:

```
accuracy: 96.58%
```

```
Confusion matrix, without normalization
```



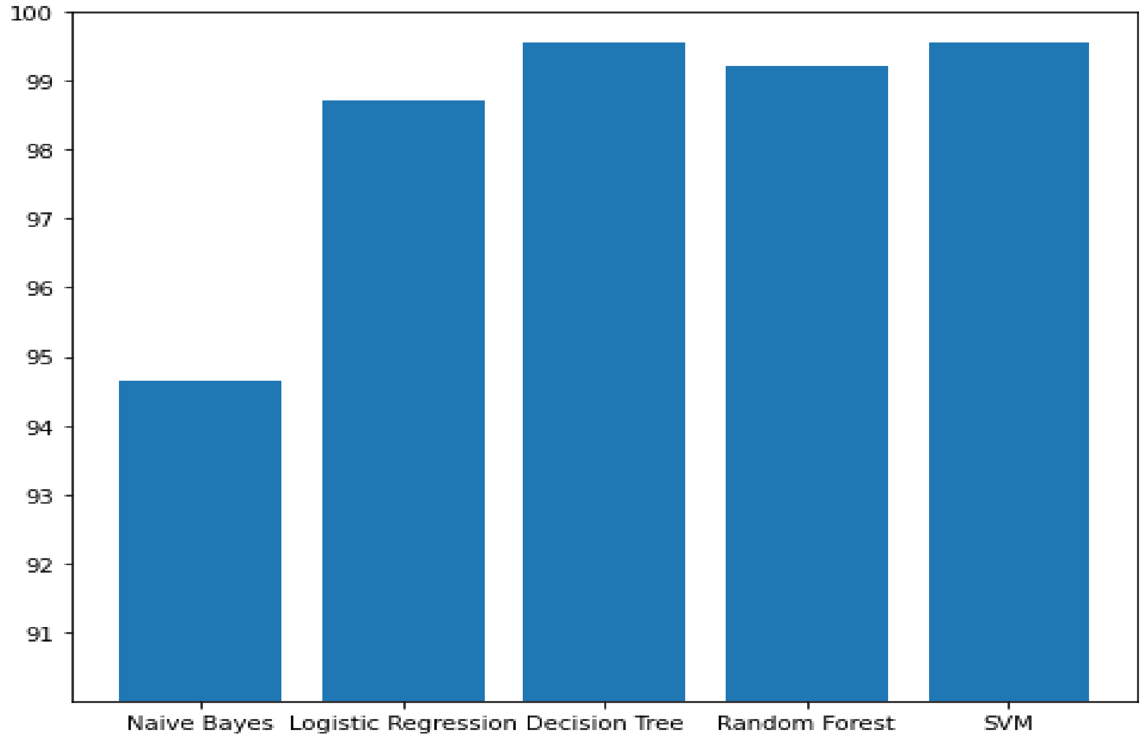
Conclusion

Experimentation using ML Algorithms

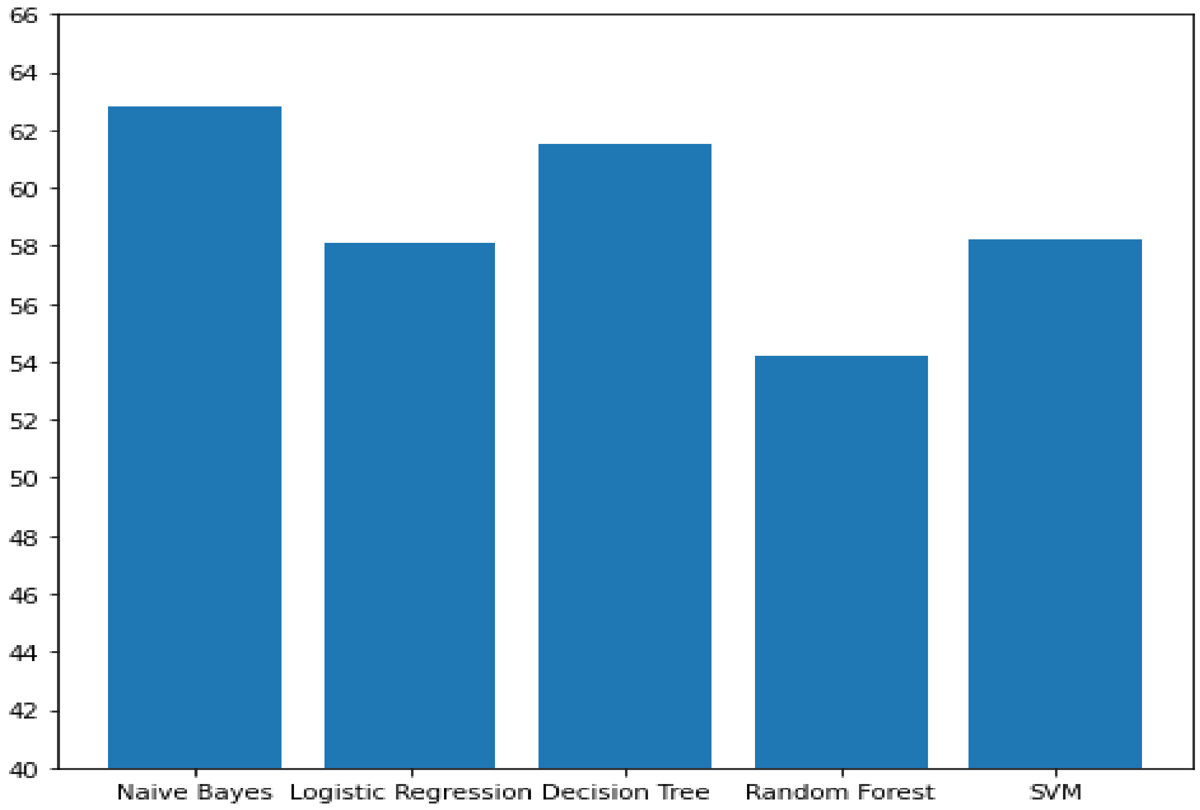
	LIAR Dataset	ISOT Dataset	Twitter Dataset	Kaggle Dataset
Model	Accuracy (in %)			
Logistic regression	58.09	98.73	93.3	94.87
Decision Tree	61.48	99.57	87.41	89.36
Random Forest	54.22	99.21	93.75	90.87
Naïve Bayes	62.83	94.65	90.51	84.66
SVM	58.25	99.55	94.03	96.58

Experimentation using ML Algorithms

ISOT Dataset

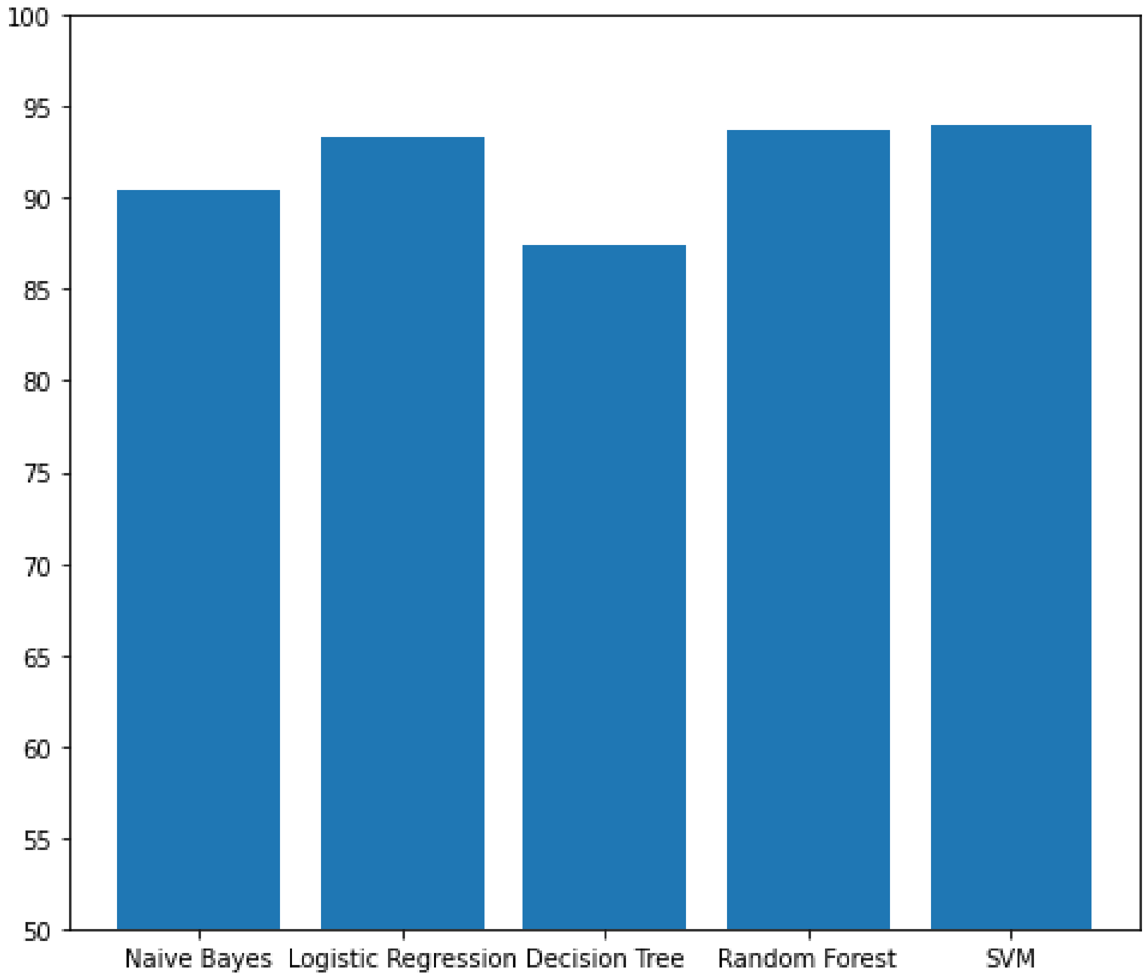


LIAR Dataset

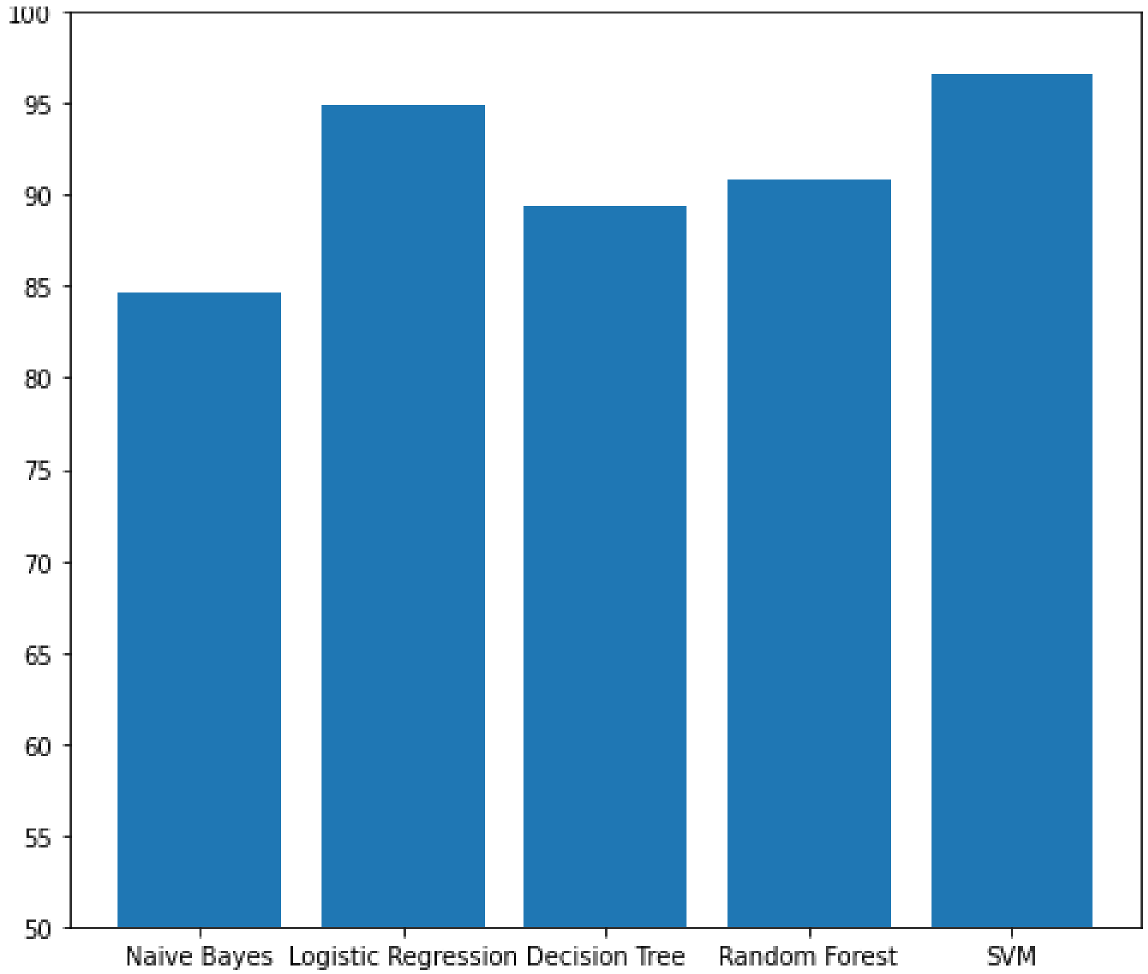


Experimentation using ML Algorithms

Twitter Dataset



Kaggle Dataset



Conclusion (Contd.)

Experimentation using BERT

Dataset	Training				Validation			
	Accuracy	F1 Score	Precision	Recall	Accuracy	F1 Score	Precision	Recall
LIAR	0.6416	0.7816	0.6416	1.000	0.6402	0.7397	0.6908	0.7960
ISOT	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Kaggle	0.9989	0.9985	0.9975	1.000	0.998906	0.9987	0.9975	1.000
Twitter	0.94812	0.9634	0.9648	0.9620	0.948593	0.9637	0.959584	0.96839

Conclusion (Contd.)

Experimentation using DistilBERT

Dataset	Training accuracy	Validation accuracy
LIAR	0.9287	0.7868
ISOT	1.00	0.9998
Twitter	0.9845	0.9074

Conclusion (Contd.)

- From the experimental study it is clear that BERT and DistilBERT can be used as a generalised model for fake news detection.
- From the four set of datasets performance of LIAR dataset is much lower because some of the articles in the LIAR dataset are from the wrong set of data (PolitiFact's Flip-o-Meter rather than its Truth-o-Meter), and yet are tagged with a truth value. As a result, those datapoint are not useful for training the model.
- LIAR dataset is considered hard to classify due to lack of sources or knowledge bases to verify with.
- In future we can expand this work to Multi modal analysis (text + images + voice).
- I focused solely on text analysis in this study, but a fraudulent source making fake news is very high; so that adding source information in addition to text analysis would improve the proposed model's real-time prediction.
- Challenges:
  - Fake news detection depends on the quality of data which varies across social media platforms.
  - Multi lingual and mixed languages.

References

[1]. Briskilal, J., and C. N. Subalalitha. "An ensemble model for classifying idioms and literal texts using BERT and RoBERTa." *Information Processing & Management* 59, no. 1 (2022): 102756.

[2]. Tuan, Nguyen Manh Duc, and Pham Quang Nhat Minh. "Multimodal Fusion with BERT and Attention Mechanism for Fake News Detection." *arXiv preprint arXiv:2104.11476* (2021).

[3]. Mehta, Divyam, Aniket Dwivedi, Arunabha Patra, and M. Anand Kumar. "A transformer-based architecture for fake news classification." *Social Network Analysis and Mining* 11, no. 1 (2021): 1-12.

[4]. Kaliyar, Rohit Kumar, Anurag Goswami, and Pratik Narang. "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach." *Multimedia Tools and Applications* 80, no. 8 (2021): 11765-11788.

[5]. Kula, Sebastian, Rafał Kozik, and Michał Choraś. "Implementation of the BERT-derived architectures to tackle disinformation challenges." *Neural Computing and Applications* (2021): 1-13.

[6]. Shishah, Wesam. "Fake News Detection Using BERT Model with Joint Learning." *Arabian Journal for Science and Engineering* (2021): 1-13.

THANK YOU…!