FAKE NEWS DETECTOR

A PROJECT REPORT

Submitted by

RASHMI R(19BCS067)

KIRUTHIKA P(19BCS077)

BHAVIKA ARAGE(19BCS079)

JOSEPHINE CYNTHIA(19BCS092)

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641 049

(An Autonomous Institution Affiliated to Anna University, Chennai)

December 2021

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE 641 049

(An Autonomous Institution Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"FAKE NEWS DETECTION"** is the Bonafide work of "***RASHMI R(19BCS067),KIRUTHIKA P(19BCS077), BHAVIKA ARAGE(19BCS079),JOSEPHINE CYNTHIA(19BCS092)*** **"** who carried out the project work under my supervision.

SIGNATURE

Dr. Devaki. P, Ph.D.,

HEAD OF THE DEPARTMENT

Department of Computer Science and Engineering,

Kumaraguru College of Technology

Coimbatore – 641 049

SIGNATURE

Senthil Kumar V, Assistant Professor

SUPERVISOR

Department of Computer Science and Engineering,

Kumaraguru College of Technology

Coimbatore – 641 049

The candidates with University register number **19BCS067, 19BCS077, 19BCS079, 19BCS092** were examined in the Project Viva-Voce examination held on …………………

Internal Examiner

External Examiner

DECLARATION

We affirm that the project work titled ”**FAKE NEWS DETECTION”** being submitted in partial fulfillment for the award of B.E Computer Science and Engineering is the original work carried out by us. It has not formed the part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

RASHMI R(19BCS067) **KIRUTHIKA P(19BCS077)**

BHAVIKA ARAGE(19BCS079) **JOSEPHINE CYNTHIA(19BCS092)**

I certify that the declaration made above by the candidates is true.

Senthil Kumar V,
Assistant Professor,
Department of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore – 641 049.

TABLE OF CONTENTS

ABSTRACT

- The advent of the World Wide Web and the rapid adoption of social media platforms (such as Facebook and Twitter) paved the way for information dissemination that has never been witnessed in the human history before. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain has to explore multiple aspects before giving a verdict on the truthfulness of an article. In this work, we propose to use machine learning ensemble approach for automated classification of news articles. Our study explores different textual properties that can be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various ensemble methods and evaluate their performance on 4 real world datasets. Experimental evaluation confirms the superior performance of our proposed ensemble learner approach in comparison to individual learners.
- With recent booming of social media, users can get infected by fake news easily, which has brought about tremendous effects on the offline society already. Consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality.
- We will be exploring BERT models, that can classify fake news as true or false, by using tools like tensorflow and google colab.

1. INTRODUCTION

1.1 CONCEPTUAL STUDY OF THE PROJECT

There has been a rapid increase in the spread of fake news in the last decade, most prominently observed in the 2016 US elections . Such proliferation of sharing articles online that do not conform to facts has led to many problems not just limited to politics but covering various other domains such as sports, health, and also science . One such area affected by fake news is the financial markets , where a rumor can have disastrous consequences and may bring the market to a halt. The World Wide Web contains data in diverse formats such as documents, videos, and audios. News published online in an unstructured format (such as news, articles, videos, and audios) is relatively difficult to detect and classify as this strictly requires human expertise. However, computational techniques such as natural language processing (NLP) can be used to detect anomalies that separate a text article that is deceptive in nature from articles that are based on facts . Other techniques involve the analysis of propagation of fake news in contrast with real news . More specifically, the approach analyzes how a fake news article propagates differently on a network relative to a true article. The response that an article gets can be differentiated at a theoretical level to classify the article as real or fake. A more hybrid approach can also be used to analyze the social response of an article along with exploring the textual features to examine whether an article is deceptive in nature or not.

1.2 OBJECTIVES OF THE PROJECT

The primary aim of the research is to identify patterns in text that differentiate fake articles from true news.

1.3 SCOPE OF THE PROJECT

Our study explores different textual properties that can be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various ensemble methods and evaluate their performance on 4 real world datasets. Experimental evaluation confirms the superior performance of our proposed ensemble learner approach in comparison to individual learners. Multiple analysis techniques are discussed in the paper to explore the spread of fake news online, such as the depth, the size, the maximum breadth, the structural virality, the mean breadth of true and false rumor cascades at various depths, the number of unique Twitter users reached at any depth, and the number of minutes it takes for true and false rumor cascades to reach depth and number of Twitter users.

2. LITERATURE REVIEW

[1] **FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network Jiawei Zhang1 , Bowen Dong2 , Philip S. Yu2 1 IFM Lab, Department of Computer Science, Florida State University, FL, USA 2BDSC Lab, Department of Computer Science, University of Illinois at Chicago, IL, USA jzhang@ifmlab.org, {bdong, psyu}@uic.edu**

Year Published : 2019

In FAKEDETECTOR, the fake news detection problem is formulated as a credibility score inference problem, and FAKEDETECTOR aims at learning a prediction model to infer the credibility labels of news articles, creators and subjects simultaneously. FAKEDETECTOR deploys a new hybrid feature learning unit (HFLU) for learning the explicit and latent feature representations of news articles, creators and subjects respectively, and introduce a novel deep diffusive network model with the gated diffusive unit for the heterogeneous information fusion within the social networks. We formulate the fake news detection problem as a credibility inference problem, where the real ones will have a higher credibility while unauthentic ones will have a lower one instead.

[2] **Fake News Detection Using Machine Learning Ensemble Methods**

Iftikhar Ahmad,1 Muhammad Yousaf,1 Suhail Yousaf,1 and Muhammad Ovais Ahmad2

Academic Editor: M. Irfan Uddin

Year Published : 2020

Various ensemble techniques such as bagging, boosting, and voting classifier are explored to evaluate the performance over the multiple datasets. We used two different voting classifiers composed of three learning models: the first voting classifier is an ensemble of logistic regression, random forest, and KNN, whereas the second voting classifier consists of logistic regression, linear SVM, and classification and regression trees (CART). The criteria used for training the voting classifiers is to train individual models with the best parameters and then test the model based on the selection of the output label on the basis of major votes by all three models.

[3] **Z Khanam1 , B N Alwasel1 , H Sirafi1 and M Rashid2**

1College of Computing and Informatics, Saudi Electronic University, Dammam, KSA
2School of Computer Science and Engineering, Lovely Professional University, Jalandhar, India
Year Published: 2020

The research in this paper focuses on detecting the fake news by reviewing it in two stages:
characterization and disclosure. In the first stage, the basic concepts and principles of fake news are highlighted in social media. During the discovery stage, the current methods are reviewed for detection of fake news using different supervised learning algorithms.

3. PROBLEM DEFINITION

With recent booming of social media, users can get infected by fake news easily, which has brought about tremendous effects on the offline society already. Consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality.

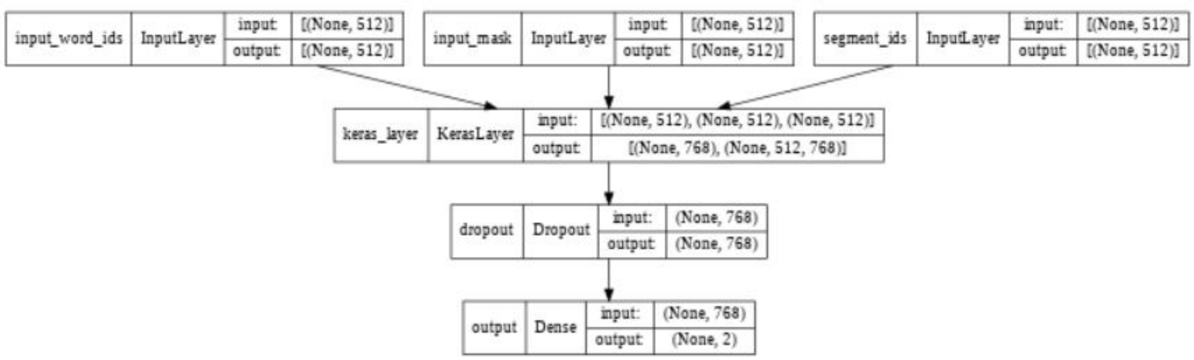
We have tried to implement a solution to classify any given news as fake or real using the latest machine learning technique for natural language processing pre-training - BERT (Bidirectional Encoder Representations from Transformers)

4.PROPOSED SYSTEM

FakeBERT In this paper, the most fundamental advantage of selecting a deep convolutional neural network is the automatic feature extraction. In our proposed model, we pass the input in the form of a tensor in which local elements correlates with one another. More concrete results can be achieved with a deep architecture which develops hierarchical representations of learning.we can perceive the computational graph of our proposed approach (FakeBERT). In many existing and useful studies , the problem of fake news has examined utilising a unidirectional pre-trained word embedding model followed by a 1D-convolutional-pooling layer network . Our suggested model obtains the advantages of automated feature engineering approach . In our model, inputs are the vectors generated after word-embedding from BERT. We give the equal dimensional input vectors to all three convolutional layers present in parallel blocks followed by a pooling layer in each block. In our proposed model, the decision of chosen number of convolutional layers, kernels_sizes, no. of filters, and optimal hyperparameters etc.

4.1. METHODOLOGY:

```
tf.keras.utils.plot_model(model, show_shapes=True, dpi=48)
```



4.3. IMPLEMENTATION

Cleaning the Dataset

Reading Dataset

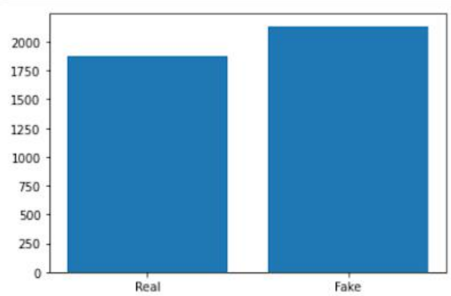
```
In [3]:
df=pd.read_csv('data.csv')
df.columns = ['URLs','Headline','Body','Label']
del df['URLs']
del df['Headline']
df = df.fillna('No data')
df.head(10)
```

Out[3]:

	Body	Label
0	Image copyright Getty Images\nOn Sunday mornin...	1
1	LONDON (Reuters) - "Last Flag Flying", a comed...	1
2	The feud broke into public view last week when...	1
3	MEXICO CITY (Reuters) - Egypt's Cheiron Holdin...	1
4	Country singer Jason Aldean, who was performin...	1
5	JetNation FanDuel League; Week 4\n% of readers...	0
6	In 2012, Kansas lawmakers, led by Gov. Sam Bro...	1
7	The Reserve Bank of India (RBI) Governor Urjit...	1
8	Scott Pruitt, Administrator of the U.S. Enviro...	1
9	FILE PHOTO - An Air Berlin sign is seen at an ...	1

```
In [4]:
df.Label.unique()

Out[4]:
array([1, 0])
```

```
In [5]:
classes = df.Label.unique()
counts=[]
for i in classes:
  count=len(df[df.Label==i])
  counts.append(count)

plt.bar(["Real","Fake"],counts)
plt.show()
```

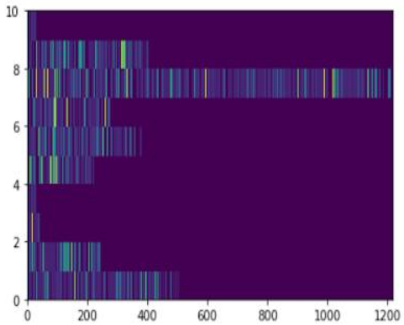


We declare the bert tokenizer and tokenize the text. The 3 input required for a bert model are tokens, masks and input type.

Tokens (into ids)

In [18]:

```
cls = [tokenizer.convert_tokens_to_ids(['[CLS]'])]*articles.shape[0]
input_word_ids = tf.concat([cls, articles], axis=-1)
_ = plt.pcolormesh(input_word_ids[0:10].to_tensor())
```



Function to get the input into the right format for bert model

Function to return the 3 arguments easily

In [23]:

```
max_seq_length=512
```

In [24]:

```
def encode_names(n, tokenizer):
   tokens = list(tokenizer.tokenize(n))
   tokens.append('[SEP]')
   return tokenizer.convert_tokens_to_ids(tokens)

def bert_encode(string_list, tokenizer, max_seq_length):
  num_examples = len(string_list)

  string_tokens = tf.ragged.constant([
      encode_names(n, tokenizer) for n in np.array(string_list)])

  cls = [tokenizer.convert_tokens_to_ids(['[CLS]'])]*string_tokens.shape[0]
  input_word_ids = tf.concat([cls, string_tokens], axis=-1)

  input_mask = tf.ones_like(input_word_ids).to_tensor(shape=(None, max_seq_length))

  type_cls = tf.zeros_like(cls)
  type_tokens = tf.ones_like(string_tokens)
  input_type_ids = tf.concat(
      [type_cls, type_tokens], axis=-1).to_tensor(shape=(None, max_seq_length))

  inputs = {
      'input_word_ids': input_word_ids.to_tensor(shape=(None, max_seq_length)),
      'input_mask': input_mask,
      'input_type_ids': input_type_ids}

  return inputs
```

In [25]:

```
X_train = bert_encode(x_train, tokenizer, max_seq_length)
X_test = bert_encode(x_test, tokenizer, max_seq_length)
```

Training the model and getting the accuracy

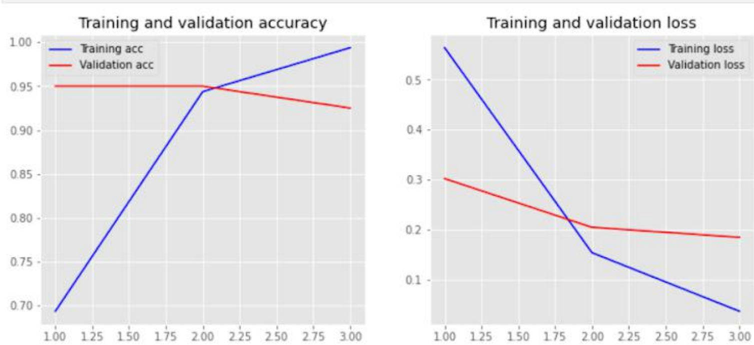
```
In [32]:  history = model.fit(X_train,
                     dummy_y_train,
                     epochs=epochs,
                     batch_size=batch_size,
                     validation_data=(X_test, dummy_y_test))

Epoch 1/3
27/27 [==============================] - 66s 2s/step - loss: 0.5639 - accuracy: 0.6938 - val_loss: 0.3021 - val_accuracy: 0.9500
Epoch 2/3
27/27 [==============================] - 45s 2s/step - loss: 0.1539 - accuracy: 0.9438 - val_loss: 0.2046 - val_accuracy: 0.9500
Epoch 3/3
27/27 [==============================] - 44s 2s/step - loss: 0.0368 - accuracy: 0.9937 - val_loss: 0.1846 - val_accuracy: 0.9250
```

```
In [33]:  loss, accuracy = model.evaluate(X_train, dummy_y_train, verbose=False)
          print("Training Accuracy: {:.4f}".format(accuracy))
          loss, accuracy = model.evaluate(X_test, dummy_y_test, verbose=False)
          print("Testing Accuracy:  {:.4f}".format(accuracy))

Training Accuracy: 1.0000
Testing Accuracy:  0.9250
```

Graph for accuracy

```
def plot_history(history):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']
    x = range(1, len(acc) + 1)

    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    plt.plot(x, acc, 'b', label='Training acc')
    plt.plot(x, val_acc, 'r', label='Validation acc')
    plt.title('Training and validation accuracy')
    plt.legend()
    plt.subplot(1, 2, 2)
    plt.plot(x, loss, 'b', label='Training loss')
    plt.plot(x, val_loss, 'r', label='Validation loss')
    plt.title('Training and validation loss')
    plt.legend()
```

```
In [35]:  plot_history(history)
```



5. <u>SYSTEM REQUIREMENTS</u>

Hardware and Software

Language used : Python

- Google colab
- Bert
- Tensorflow

6. CONCLUSION

Fake news detection has many open issues that require attention of researchers. For instance, in order to reduce the spread of fake news, identifying key elements involved in the spread of news is an important step. Graph theory and machine learning techniques can be employed to identify the key sources involved in spread of fake news. Likewise, real time fake news identification in videos can be another possible future direction.

REFERENCES

1. https://arxiv.org/pdf/1805.08751.pdf
2. https://www.hindawi.com/journals/complexity/2020/8885861/
3. https://iopscience.iop.org/article/10.1088/1757-899X/1099/1/012040/pdf
4. https://en.wikipedia.org/wiki/Detecting_fake_news_online#Limitations_of_detecting_fake_news
5. https://www.kdnuggets.com/2017/10/guide-fake-news-detection-social-media.html