NATURAL LANGUAGE PROCESSING LAB

MINI PROJECT
ON

FAKE NEWS DETECTION

Submitted in partial fulfilment of the requirements of the degree
BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

Pooja Bhagat (BEA-110)

Priyanka Korde (BEA-113)

Raghuwardayal Maurya (BEA-154)

Asmit Patil (BEA-131)

Under the guidance of

Prof. Rakhi Akhare

LOKMANYA TILAK COLLEGE OF ENGINEERING

Department of Computer

Engineering **YEAR 2022 – 2023**

CERTIFICATE

This is to certify that the project entitled **"Fake News Detection"** is a bonafide work of **Pooja Bhagat (BEA-110), Priyanka Korde (BEA-113), Raghuwardayal Maurya (BEA-154), Asmit Patil (BEA-131)** Submitted to **Prof. Rakhi Akhare** in partial fulfilment of the requirement for the course of Natural Language Processing Lab.

(Prof. Rakhi Akhare)

ACKNOWLEDGEMENT

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to my highly respected and esteemed guide **Prof. Rakhi Akhare,** Computer Department for her valuable guidance, encouragement and help for completing this work. Her useful suggestions for this whole work and co-operative behaviour are sincerely acknowledged.

We also wish to express my gratitude to Prof. **R.D.Gawal**i (Head – Computer Engineering) for his kind hearted support.

Students Signature

TABLE OF CONTENTS

Acknowledgements		**3**
1.	**Introduction**	**5**
2.	**Problem Statement**	**6**
3.	**Motivation**	**6**
4.	**Design of System**	**7**
5.	**Project Code**	**9**
6.	**Results**	**11**
7.	**Conclusion**	**12**
References		**13**

INTRODUCTION

Fake News is news, stories, or hoaxes created to deliberately misinform or deceive readers. Usually, these stories are created to either influence people' s views, push a political agenda, or cause confusion and can often be a profitable business for online publishers. The purpose of choosing this topic is because it is becoming a serious social challenge . It is leading to a poisonous atmosphere on the web and causing riots and lynching on the road. Examples: political fake news, news regarding sensitive topics such as religion, covid news like salt and garlic can cure corona and all such messages we get through social media. We all can see the damage that can be caused because of fake news which is why there is a dire need for a tool that can validate particular news weather it is fake or real and give people a sense of authenticity based on which they can decide whether or not to take action, amongst so much noise of fake news and fake data if people lose faith in information, they will no longer be able to access even the most vital information that can even sometimes be life- changing or lifesaving. Our approach is to develop a model wherein it will detect whether the given news is false or true using LSTM (long short-term memory) and other machine learning concepts such as NLP, word embedding one hot representation, etc. The model will give us the results for the dataset provided. It gives accuracy up to 99 %.

PROBLEM STATEMENT

To develop a FAKE NEWS DETECTION system using natural language processing and the model must be able to detect news is true or fake in a given scenario.

MOTIVATION

The rise of fake news during the 2016 U.S. Presidential Election highlighted not only the dangers of the effects of fake news but also the challenges presented when attempting to separate fake news from real news. Fake news may be a relatively new term but it is not necessarily a new phenomenon. Fake news has technically been around at least since the appearance and popularity of one-sided, partisan newspapers in the 19th century. However, advances in technology and the spread of news through different types of media have increased the spread of fake news today. As such, the effects of fake news have increased exponentially in the recent past and something must be done to prevent this from continuing in the future. I have identified the three most prevalent motivations for writing fake news and chosen only one as the target for this project as a means to narrow the search in a meaningful way. The first motivation for writing fake news, which dates back to the 19th century one-sided party newspapers, is to influence public opinion. The second, which requires more recent advances in technology, is the use of fake headlines as clickbait to raise money. As such, this paper will focus primarily on fake news as defined by politifact.com, “fabricated content that intentionally masquerades as news coverage of actual events.” This definition excludes satire, which is intended to be humorous 8 and not deceptive to readers. Most satirical articles come from sources. Satire can already be classified, by machine learning techniques Therefore, our goal is to move beyond these achievements and use machine learning to classify, at least as well as humans, more difficult discrepancies between real and fake news.

DESIGN OF SYSTEM

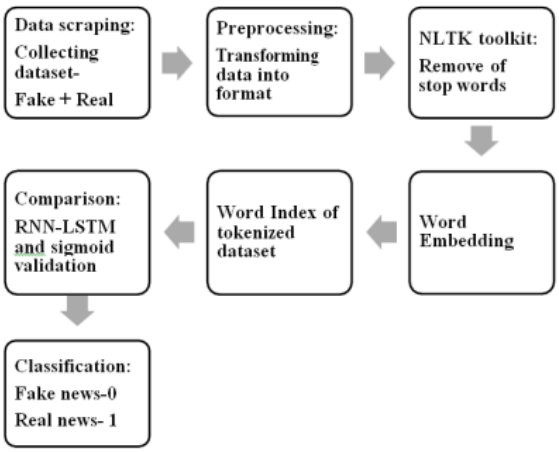

Fig1. Architecture flow of Proposed System

Overview of Dataset:

Dataset is taken from Kaggle platform. It has the following attributes: Title, Text, Subject, Date.. Dataset consists of 40,000 news articles for training and testing of models. Dataset is formed with a combination of real and fake news.

Pre processing:

To transform data into the relevant format the data set needs preprocess. Firstly, we removed all the NAN values from the dataset. Vocabulary size of 1000 words is decided. Then NLTK (Natural Language Processing) Tool Kit is used to remove all the stop words from the dataset. Stop words is list of punctuations + stop words from nltk toolkit i.e. Words such as 'and' ' the' and 'I' that don't convey much information converting them to lowercase and removing punctuation. For each word in documents if it is not a stop word then that words tag is taken from the postag. Then, this collection of words is appended to the document. Word tokenizing, appends text to a list and the list be named as documents. The output for this stage is the list of all the words in the narration

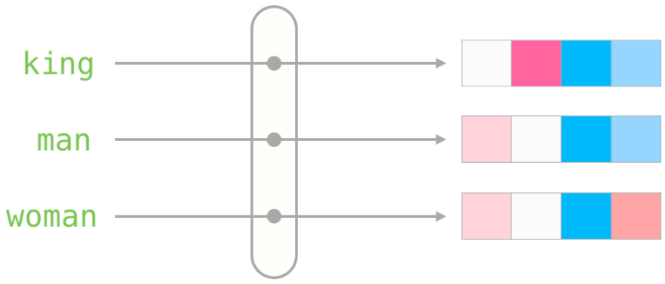

Fig2. Word2vec

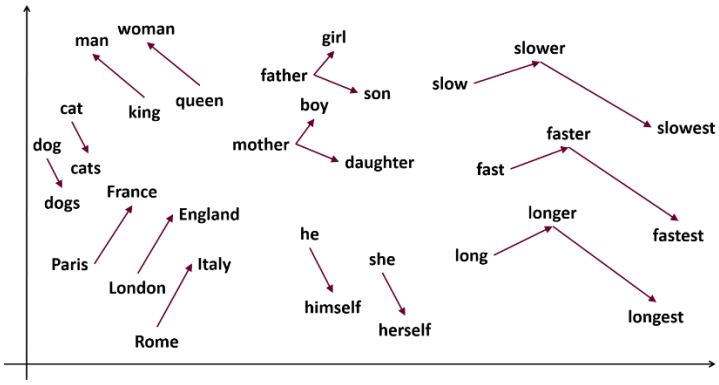

Fig3. Vectorization of Vocabulary words: Word to Vec

Vectorization - Word2Vec:

Word2Vec is one of the most popular techniques to learn word embeddings using shallow neural networks Word embedding is the most popular representation of document vocabulary. It is capable of capturing the context of word in a document semantic and syntactic, relation with other words, etc. Using Gensim Library for vectorization of the words in the vocabulary.

LSTM model:

Long short term memory (LSTM) units are a building block for the layers of recurrent neural network (RNN). A LSTM unit is composed of a cell, an input gate , an output gate and a forget gate. The cell is responsible for"remembering" values over a vast time interval so that the relation of the word in the starting of the text can influence the output of the word later in the sentence. Traditional neural networks cannot remember or keep the record of what all is passed before they are executed. This stops the desired influence of words that comes in the sentence before having any influence on the ending words, and it seems like a major shortcoming.

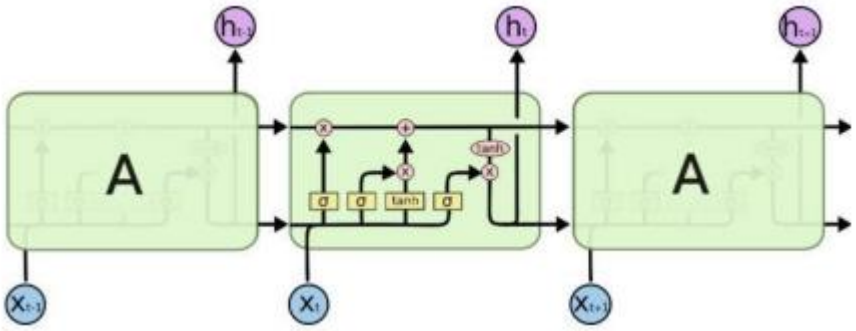

Fig4. Architecture of LSTM

PROJECT CODE

Creating model using word embedding and lSTM in Deep learning:

```
[69] model = Sequential()
     model.add(Embedding(vocab_size, output_dim=DIM, weights=[embedding_vectors], input_length=maxlen, trainable=False))
     model.add(LSTM(units=128))
     model.add(Dense(1,activation='sigmoid'))
     model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
```

```
[70] model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1050, 100)         37537400

 lstm (LSTM)                 (None, 128)               117248

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 37,654,777
Trainable params: 117,377
Non-trainable params: 37,537,400
_________________________________________________________________
```

Output from the word embedding is provided to the model. The machine learning model implemented here is a sequential model consisting of embedding as the first layer which consists of values, vocabulary size, number of features and length of sentence. The next is LSTM with 100 neurons for each layer, followed by a Dense layer with sigmoid activation function as we need one final output. We have used binary cross entropy to calculate loss, Adam optimizer fori adaptive estimation, finally adding drop out layer in between so that overfitting is avoided. Then training and testing of the model is done.

Model Training:

```
[71] X_train, X_test, y_train, y_test = train_test_split(X,y)
```

```
[72] model.fit(X_train, y_train, validation_split=0.3, epochs=5)
```

```
Epoch 1/5
737/737 [==============================] - 46s 54ms/step - loss: 0.1179 - acc: 0.9571 - val_loss: 0.0605 - val_acc: 0.9783
Epoch 2/5
737/737 [==============================] - 39s 53ms/step - loss: 0.0342 - acc: 0.9889 - val_loss: 0.0382 - val_acc: 0.9879
Epoch 3/5
737/737 [==============================] - 40s 55ms/step - loss: 0.0262 - acc: 0.9925 - val_loss: 0.0237 - val_acc: 0.9926
Epoch 4/5
737/737 [==============================] - 39s 53ms/step - loss: 0.0112 - acc: 0.9961 - val_loss: 0.0298 - val_acc: 0.9913
Epoch 5/5
737/737 [==============================] - 40s 54ms/step - loss: 0.0106 - acc: 0.9963 - val_loss: 0.0219 - val_acc: 0.9935
<keras.callbacks.History at 0x7f8b61d28510>
```

Model Accuracy:

```
[73] # Accuracy on Test Dataset
     y_pred = (model.predict(X_test) >= 0.5).astype(int)
```

```
[74] accuracy_score(y_test, y_pred)
```

```
0.9952783964365256
```

Conclusion Matrix:

```
[75] print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      5842
           1       0.99      1.00      1.00      5383

    accuracy                           1.00     11225
   macro avg       1.00      1.00      1.00     11225
weighted avg       1.00      1.00      1.00     11225
```

RESULT

True New:

```
[100] x = ['''CBI grills Russian national for manipulating software of JEE Mains examination
      The foreign national was picked up by the CBI from the Bureau of Immigration
      at Indira Gandhi International Airport in Delhi where he was detained on arrival from Kazakhstan.''']
      x = tokenizer.texts_to_sequences(x)
      x = pad_sequences(x,maxlen=1050)
      if((model.predict(x) >=0.5).astype(int) == 0):
        print("Fake News !!!")
      else:
        print("True New!!!")
```

```
True New!!!
```

Fake New:

```
[82] x = ['''Govt making efforts to obtain files relating to Netaji: MoS Muraleedharan in Rajya Sabha''']
     x = tokenizer.texts_to_sequences(x)
     x = pad_sequences(x,maxlen=1050)
     if((model.predict(x) >=0.5).astype(int) == 0):
       print("Fake News !!!")
     else:
       print("True New!!!")
```

```
Fake News !!!
```

CONCLUSION

In this project , we are predicting whether an article is a real or fake article based on the relationship between the words . We have used the political datasets for creation of this system . We used Word2Vec model for building model and 4 ML algorithms such as logistic regression , decision tree classification, random forest classifier and gradient boosting classification for the prediction and obtained an accuracy of 99 %.

REFERENCES

1. https://www.irjet.net/archives/V8/i4/IRJET-V8I4465.pdf
2. Building a fake news classifier using natural language processing BY NATHAN (https://towardsdatascience.com/building-a-fake-news-classifier-using-naturallanguage-processing-83d911b237e1).
3. Shloka Gilda,"Evaluating Machine Learning Algorithms for Fake News Detection" ,2017 IEEE 15th Student Conference on Research and Development (SCOReD).
4. International journal of recent technology and engineering (IJRTE) ISSN: 2277-3878, volume-7, issue-6, march 2019.