**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA

****

**COURSE LA1 REPORT**

on

**Fake News Detection Using LSTM Method**

*Submitted in partial fulfilment of the requirement for the award of Degree of*

*Bachelor of Engineering*

*in*

*Computer Science and Engineering*

Submitted by:

|  |  |
| --- | --- |
| Rahul R Satwik  Rakshith Maradi  Rakshith R  Tarun Kumar Arcot | 1NT18CS124  1NT18C125  1NT19CS413  1NT18CS174 |



**Department of Computer Science and Engineering**

2020-21

|  |  |  |
| --- | --- | --- |
|  | **Nitte Meenakshi Institute of Technology**  (AN AUTONOMOUS INSTITUTION AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM)  PB No. 6429, Yelahanka, Bangalore 560-064, Karnataka  Telephone: 080- 22167800, 22167860  Fax: 080 – 22167805 |  |

**Department of Computer Science and Engineering**

**CERTIFICATE**

This is to certify that the Course Project titled “**Fake News Detection using LSTM**” is an authentic work carried out by Rahul R Satwik (1NT18CS124), Rakshith Maradi **(1T18CS125),** Rakshith R **(1NT19CS413)** and Tarun Kumar Arcot **(1NT18CS174)**  Bonafede students of **Nitte Meenakshi Institute of Technology**, Bangalore in partial fulfilment for the award of the degree of ***Bachelor of Engineering*** in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belagavi during the academic year ***2019-2020.***

|  |  |  |
| --- | --- | --- |
| **Name Signature of the Faculty Incharge** |  | **Name and Signature of the HOD** |

**DECLARATION**

We hereby declare that

(i) This Presentation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the report and in the References sections.

(ii) All corrections and suggestions indicated during the internal presentation have been incorporated in the report.

(iii) Content of the report has been checked for the plagiarism requirement

Name USN Signature

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NAME** | | | **USN** | | **Signature** | |
| Rahul R Satwik | | | 1NT18CS124 | |  | |
| Rakshith Maradi | | | 1NT18CS125 | |  | |
| Rakshith R | | | 1NT19CS413 | |  | |
| Tarun Kumar Arcot | | | 1NT18CS174 | |  | |
|  |  | |  | |
|  |  | |  | |
|  |  | |  | |
|  |  | |  | |

**Date: 15-01-2021**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. we express our sincere gratitude to our Principal **Dr. H. C. Nagaraj**, Nitte Meenakshi Institute of Technology for providing facilities.

We wish to thank our HoD**, Dr. Thippeswamy M.N** for the excellent environment created to further educational growth in our college. We also thank him for the invaluable guidance provided which has helped in the creation of a better technical report.

Thanks to our Subject Faculty. We also thank all our friends, teaching and non-teaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the presentation.

|  |  |  |
| --- | --- | --- |
| **NAME** | **USN** | **Signature** |
| Rahul R Satwik | 1NT18CS124 |  |
| Rakshith Maradi | 1NT18CS125 |  |
| Rakshith R | 1NT19CS413 |  |
| Tarun Kumar Arcot | 1NT18CS174 |  |

Date: 15-01-2021

ABSTRACT

Society and individuals are negatively influenced both politically and socially by the widespread increase of fake news either way generated by humans or machines. In the era of social networks, the quick rotation of news makes it challenging to evaluate its reliability promptly. Therefore, automated fake news detection tools have become a crucial requirement. To address the aforementioned issue, a hybrid Neural Network architecture, that combines the capabilities of CNN and LSTM, is used with two different dimensionality reduction approaches, Principal Component Analysis (PCA) and Chi-Square. This work proposed to employ the dimensionality reduction techniques to reduce the dimensionality of the feature vectors before passing them to the classifier. To develop the reasoning, this work acquired a dataset from the Fake News Challenges (FNC) website which has four types of stances: agree, disagree, discuss, and unrelated. The nonlinear features are fed to PCA and chi-square which provides more contextual features for fake news detection. The motivation of this research is to determine the relative stance of a news article towards its headline. The proposed model improves results by 4% and 20% in terms of *Accuracy* and *F*1 score. The experimental results show that PCA outperforms than Chi-square and state-of-the-art methods with 97:8% accuracy.

**TABLE OF CONTENTS**

1. **DECLARATION**
2. **ACKNOWLEDGEMENTS**
3. **ABSTRACT**
4. **TABLE OF CONTENTS**

**(***Include:-*

*LIST OF FIGURES*

*LIST OF TABLES*

*LIST OF ACRONYMS***)**

**CHAPTER 1 : INTRODUCTION -----------1**

* 1. Brief Technology/concept/application -----------1
  2. Research objectives
  3. Organization of the technical report

The following provides the roadmap for remainder of this report:

**Chapter 2:** Presents a brief overview/working of the proposed topic that you have undertaken. Highlight the application of the topic chosen/concept used, advantage and disadvantage of the technology/concept chosen,

**Note: This chapter can be split into multiple sections as needed**

**Chapter 3:** The program you executed and result

**Chapter 4: Conclusion**

**Chapter 5: References**

***Follow this format compulsorily to write the refernce in the reference chapter as applicable to your report.***

# Chapter 1

# Introduction

## Brief description of your topic:-

Deep learning models such as recurrent neural networks (RNN) and its variants and convolution neural networks (CNN) have been used effectively in many NLP tasks that share similarities to fake news and consist of calculating semantic similarity between sentences and community based question answering. Siamese MaLSTM is used to compute the semantic similarity of question pairs. A deep neural network converts the text sequence into fixed length vector representation which is then used to measure the relevance of two textual sequence, which is the relevance of each headline-body pair.

Stance Detection is a well-established and well-researched task in NLP. It is defined as determining from the text whether the audience is in favor, against or neutral about the target. Stance detection has become foundation for many tasks such as fake news detection, claim validation, and argument search. Previous studies in fake news detection

focused on target-specific stance prediction in which the stance of a text entity relating to a topic or a named entity is determined. In many researches, target-specific stance prediction is performed for tweets (where tweets are the text and target is single stance) and online debates. Such target-specific approaches are based on structural features, and linguistic and lexical features. Stance prediction in tweets and online debates is different from stance detection in a news article in which the stance detection of a news article is relative to the headline in NLP. The authenticity of claims is predicted with the use of the stance of articles and the reliability.

In detecting the reliability of fake news, stance features are used, which are also defined as unsupported claims. A researcher used tweets publishing time and stances as

the only features for determining the authenticity of tweets by using Hidden Markov Models. Another study provides an approach to the claim-relevance discovery problem

by leveraging various information retrieval and machine learning techniques and yielding 91.6% accuracy. The first fake news stance detection challenge was initiated

back in 2017. The inspiration behind the FNC-1 stance detection task was taken from the work proposed, in which they classify the stance of a single sentence of a news headline towards a specific claim. The dataset used in the FNC-1 challenge was partially labeled and based on the Emergent dataset. In FNC-1, the stance is detected on document level in which the entire news article is classified relative to a headline. The top-performing system in FNC-1 is developed by Talos Research Intelligence team called SOLAT in the SWEN. It is based on a 50-50 weighted average ensemble method that combines deep CNN with Google News pre-trained vectors, and gradient-boosted

decision trees. The model achieved 82:02% accuracy. The 1st runner up `Athene' team, consisting of members from the Ubiquitous Knowledge Processing Lab and

the Adaptive Preparation of Information from Heterogeneous Sources Research Training Group at Technische Universität Darmstadt (TU Darmstadt), uses a multi-layer

perceptron (MLP) as an ensemble of six hidden layers with hand-crafted features and obtains 81:97% accuracy. The 2nd runner up team, UCL Machine Reading (UCLMR),

uses two bags of words (BOW), term frequency (TF), and term frequency-inverse document frequency (TF-IDF) as features and proposes a model consisting of MLP with 81:72% accuracy. The fourth-best team extract both semantic embedding and lexical matching features and pass them to another gradient boosting trees. Additionally, a two-step logistic regression classifier and an ensemble models of five classifiers achieve 9th and 11th places respectively. All three challenge winners in SemEval and FNC make use of both hand-crafted and neural network-based features with classification-based algorithms. In another research, the focus was on predicting rumor news using an agreement aware article search. They developed an agreement-aware search framework designed to provide users with a holistic view of a question, for which the ground truth was not confident. They designed a two-step model consisting of a tree-based model based on handcrafted features and an RNN plus attention model focusing on only a

few key sentences. The proposed model in is a single, end-to-end ranking-based algorithm with MLP. TF-IDF is used to extract features to represent both headlines and bodies of the news articles. The model obtains 86:66% accuracy on FNC-1. A deep learning method is used for addressing the stance detection problem from the FNC-1 task. It incorporates bi-directional RNNs together with max-pooling and neural attention mechanisms to build representations from headlines and from the body of news articles and combine these representations with external similarity features. The use of pre-training and the combination of neural representations together with external similarity features produces 83.8% accuracy. Another work uses deep recurrent model

to compute the neural embedding, weighted n-gram bag-of words model to compute the statistical features and feature engineering heuristics to extract hand crafted external features. Finally, all the features are combined, by using deep neural layer for the classification of the headline-body news pair as agree, disagree, discuss, or unrelated. The obtained accuracy is 89.29%. It is proved in [30] that neural network outperforms hand-crafted features. By implementing bilateral multi-perspective matching models (BiMPM) and improving the existing Attentive Reader with a full attention mechanism

between words in body text and headlines makes the model able to achieve 86.5% accuracy. A Conditional Encoding LSTM model with attention yields a 80.8% score in.

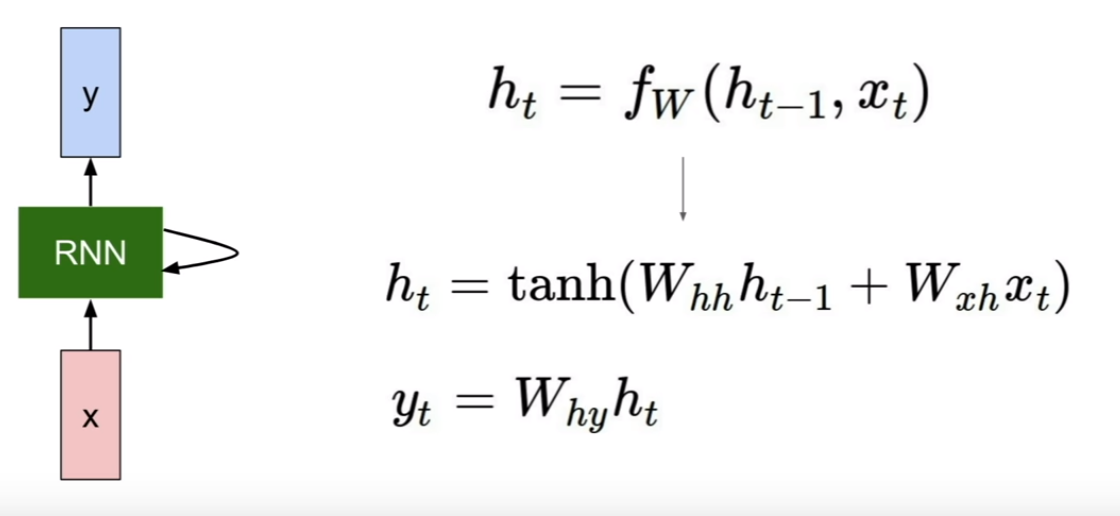
A conditioned bidirectional LSTM with global features is used. It demonstrates that the combination of global features and local word embedding features is better at predicting the stance of headline-article pairs than each of them individually by obtaining 87.4% accuracy. Rather than using a classification-based method, this research tackles the

news stance detection task by using a ranking-based method. The ranking-based method compares and maximizes the difference between the true and false stances of a given pair of headlines and article bodies. This approach results in 86.66% accuracy.

The architecture of LSTM is given below:-

Ordinary Neural Networks don’t perform well in cases where sequence of data is important. For example: language translation, sentiment-analysis, time-series and more. To overcome this failure, RNNs were invented. RNN stands for “Recurrent Neural Network”. An RNN cell not only considers its present input but also the output of RNN cells preceding it, for it’s present output.

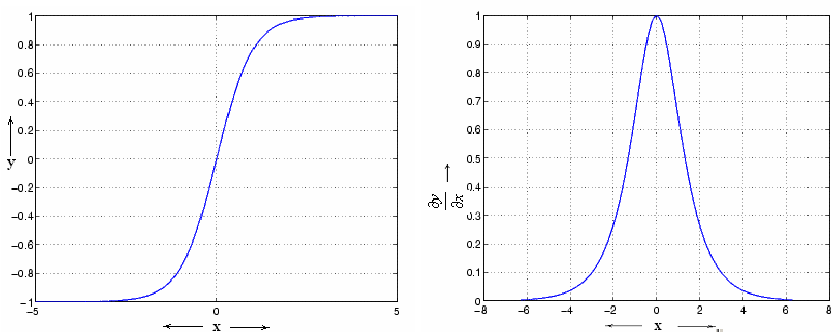
**Simple form of Vanilla RNN’s present state could be represented as**:



RNNs performed very well on sequential data and performed well on tasks where sequence was important.

**But there exists many problems with ordinary RNNs**

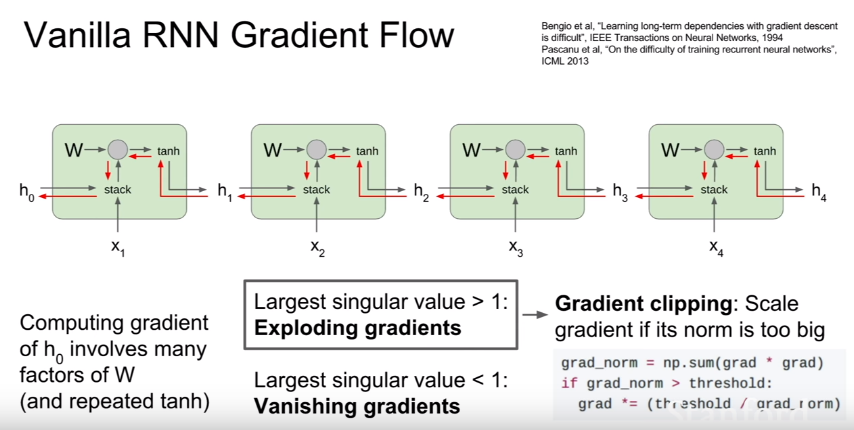
**Vanishing gradients problem:**



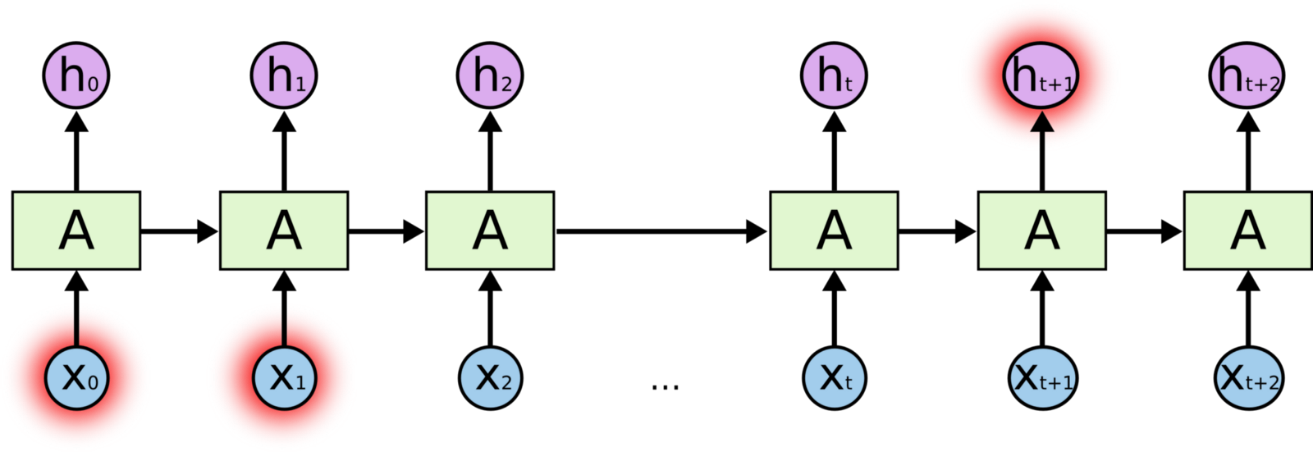
Hyperbolic tangent(tanh) is mostly used as activation function in RNNs which lies in [-1,1] and derivative of tanh lies in [0,1]. During backpropagation, as gradient is calculated by chain rule, it has an effect of multiplying these small numbers **n** (number of times tanh used in rnn architecture) **times** which squeezes the final gradient to almost zero and hence subtracting gradient from weights doesn’t make any change to them which stops the training of model.

**Exploding gradients problem:**

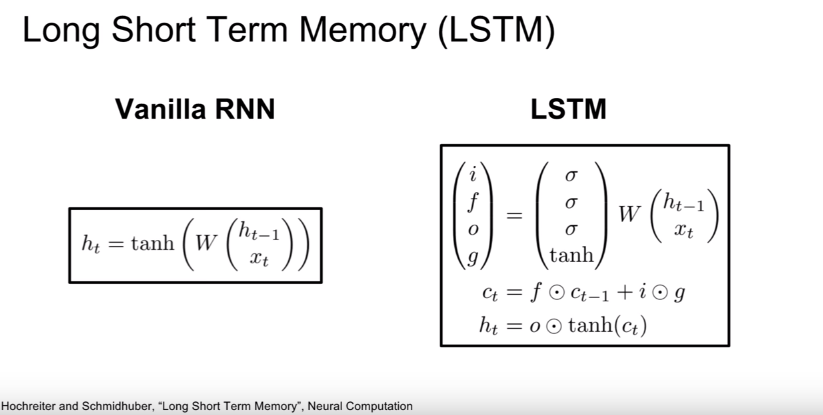
Opposite to vanishing gradient problem, while following chain rule we multiply with the weight matrix(transposed W )too at each step, and if the values are larger than 1, multiplying a large number to itself many times leads to a very large number leading to explosion of gradient.



**Long-Term Dependencies problem**



RNNs are good in handling sequential data but they run into problem when the context is far away. **Example: I live France and I know \_\_\_\_.** The answer must be ‘**French**’ here but if the there are some more words in between ‘**I live in France**’ & ‘**I know \_\_\_\_**’. It’ll be difficult for RNNs to predict ‘French’. **This is the problem of Long-Term Dependencies.** Hence we come to LSTMs.

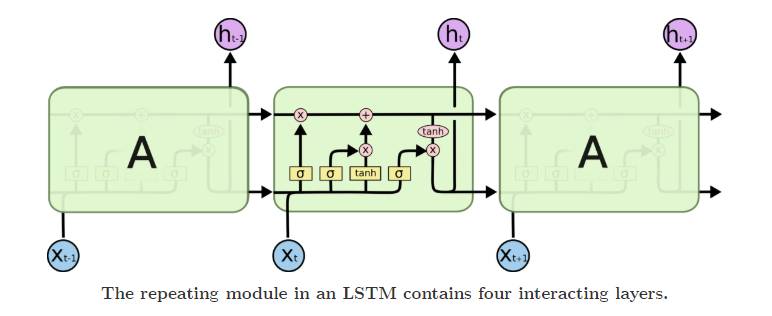


**RNN vs LSTM cell representation,** source: stanford

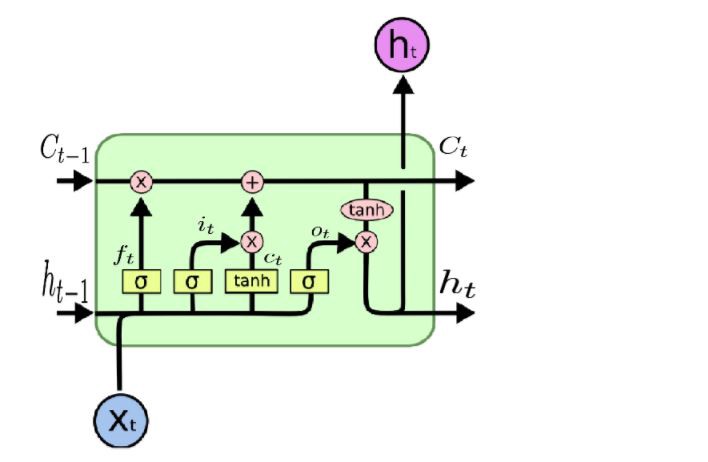
At start, we need to initialize the weight matrices and bias terms as shown below.

**Some information about an LSTM cell**

A simple LSTM cell consists of 4 gates:



**3 LSTM cells connected to each other.** source:Google



## Organization of the report

The following provides the roadmap for remainder of the report:

**Chapter 2:** Presents a brief overview/working of the proposed area of interest that you have undertaken. Highlight the application of the technology/concept used, advantage and disadvantage of the technology/concept chosen.

**Chapter 3:** The program you executed and result,

**Chapter 4:** Conclusion

# Chapter 2

# 2.1 Related Work

The research in the field of fake news detection has been intense in recent years. However, most of the work in the area is focused on the idea of studying and detecting the hoaxes on its main spreading channel: social media. The probability of a given post to be false is studied using its own characteristics such as likes, followers, shares, etc., through classical machine learning methods (classification trees, SVM, ...). Applying these kinds of approximations, the best results are obtained in click-bait news are detected achieving results of 93% of accuracy. Other works like [8] use graph-based approximations for studying the relations between users who share news and the path which the shared content follows in order to stop it in order to mitigate its potential deception effects. Although the general trend is to analyse the way hoaxes are spread, other alternatives focused on the analysis of the content of the news the have begun to appear. Besides of the user features who shares the news, the text is used to discriminate fake news. On the other hand, the statements in an article are studied in order to detect false facts in the content. Regarding the use of modern deep learning algorithms, the company Fabula.ai, recently acquired by Twitter, uses a method which takes into account both the content of news and the features extracted from the social networks, achieving results of 0:93 of AUC. The performance of several algorithms (both classic and deep learning) is compared categorizing news among the categories of true and fake, achieving results of 95% of accuracy. Finally, using only the content of the news, a convolutional neural network-based technique is proposed to detect fake news using the titles and the heading image, obtaining results of 92% of accuracy.

# 2.2 Brief history of technologies/concept

Deep learning models such as recurrent neural networks (RNN) and its variants and convolution neural networks (CNN) have been used effectively in many NLP tasks that share similarities to fake news and consist of calculating semantic similarity between sentences and community based question answering. Siamese MaLSTM is used to compute the semantic similarity of question pairs. A deep neural network converts the text sequence into fixed length vector representation which is then used to measure the relevance of two textual sequence, which is the relevance of each headline-body pair.

Stance Detection is a well-established and well-researched task in NLP. It is defined as determining from the text whether the audience is in favor, against or neutral about the target. Stance detection has become foundation for many tasks such as fake news detection, claim validation, and argument search. Previous studies in fake news detection

focused on target-specific stance prediction in which the stance of a text entity relating to a topic or a named entity is determined. In many researches, target-specific stance prediction is performed for tweets (where tweets are the text and target is single stance) and online debates. Such target-specific approaches are based on structural features, and linguistic and lexical features. Stance prediction in tweets and online debates is different from stance detection in a news article in which the stance detection of a news article is relative to the headline in NLP. The authenticity of claims is predicted with the use of the stance of articles and the reliability.

In detecting the reliability of fake news, stance features are used, which are also defined as unsupported claims. A researcher used tweets publishing time and stances as

the only features for determining the authenticity of tweets by using Hidden Markov Models. Another study provides an approach to the claim-relevance discovery problem

by leveraging various information retrieval and machine learning techniques and yielding 91.6% accuracy. The first fake news stance detection challenge was initiated

back in 2017. The inspiration behind the FNC-1 stance detection task was taken from the work proposed, in which they classify the stance of a single sentence of a news headline towards a specific claim. The dataset used in the FNC-1 challenge was partially labeled and based on the Emergent dataset. In FNC-1, the stance is detected on document level in which the entire news article is classified relative to a headline. The top-performing system in FNC-1 is developed by Talos Research Intelligence team called SOLAT in the SWEN. It is based on a 50-50 weighted average ensemble method that combines deep CNN with Google News pre-trained vectors, and gradient-boosted

decision trees. The model achieved 82:02% accuracy. The 1st runner up `Athene' team, consisting of members from the Ubiquitous Knowledge Processing Lab and

the Adaptive Preparation of Information from Heterogeneous Sources Research Training Group at Technische Universität Darmstadt (TU Darmstadt), uses a multi-layer

perceptron (MLP) as an ensemble of six hidden layers with hand-crafted features and obtains 81:97% accuracy. The 2nd runner up team, UCL Machine Reading (UCLMR),

uses two bags of words (BOW), term frequency (TF), and term frequency-inverse document frequency (TF-IDF) as features and proposes a model consisting of MLP with 81:72% accuracy. The fourth-best team extract both semantic embedding and lexical matching features and pass them to another gradient boosting trees. Additionally, a two-step logistic regression classifier and an ensemble models of five classifiers achieve 9th and 11th places respectively. All three challenge winners in SemEval and FNC make use of both hand-crafted and neural network-based features with classification-based algorithms. In another research, the focus was on predicting rumor news using an agreement aware article search. They developed an agreement-aware search framework designed to provide users with a holistic view of a question, for which the ground truth was not confident. They designed a two-step model consisting of a tree-based model based on handcrafted features and an RNN plus attention model focusing on only a

few key sentences. The proposed model in is a single, end-to-end ranking-based algorithm with MLP. TF-IDF is used to extract features to represent both headlines and bodies of the news articles. The model obtains 86:66% accuracy on FNC-1. A deep learning method is used for addressing the stance detection problem from the FNC-1 task. It incorporates bi-directional RNNs together with max-pooling and neural attention mechanisms to build representations from headlines and from the body of news articles and combine these representations with external similarity features. The use of pre-training and the combination of neural representations together with external similarity features produces 83.8% accuracy. Another work uses deep recurrent model

to compute the neural embedding, weighted n-gram bag-of words model to compute the statistical features and feature engineering heuristics to extract hand crafted external features. Finally, all the features are combined, by using deep neural layer for the classification of the headline-body news pair as agree, disagree, discuss, or unrelated. The obtained accuracy is 89.29%. It is proved in [30] that neural network outperforms hand-crafted features. By implementing bilateral multi-perspective matching models (BiMPM) and improving the existing Attentive Reader with a full attention mechanism

between words in body text and headlines makes the model able to achieve 86.5% accuracy. A Conditional Encoding LSTM model with attention yields a 80.8% score in.

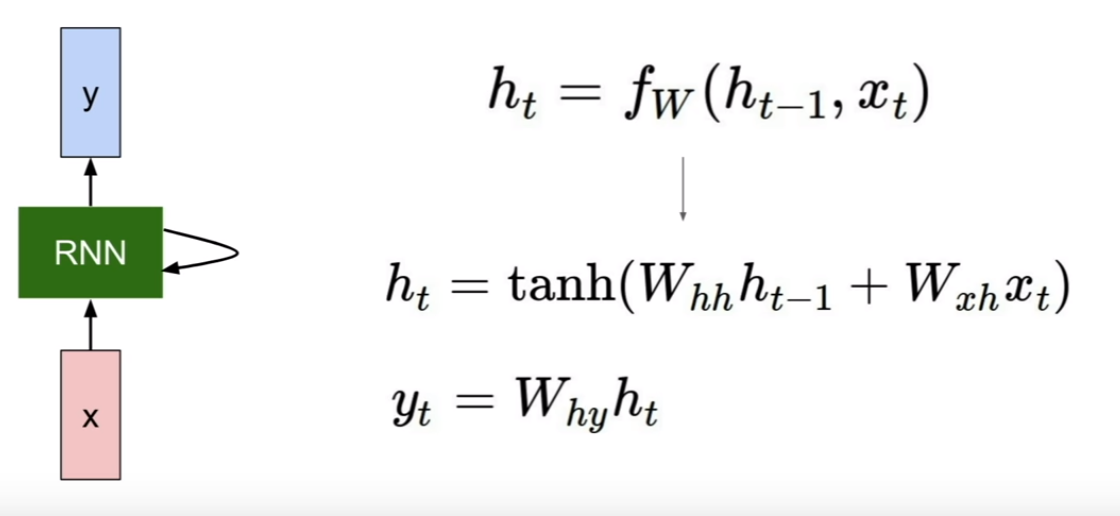
A conditioned bidirectional LSTM with global features is used. It demonstrates that the combination of global features and local word embedding features is better at predicting the stance of headline-article pairs than each of them individually by obtaining 87.4% accuracy. Rather than using a classification-based method, this research tackles the

news stance detection task by using a ranking-based method. The ranking-based method compares and maximizes the difference between the true and false stances of a given pair of headlines and article bodies. This approach results in 86.66% accuracy.

The architecture of LSTM is given below:-

Ordinary Neural Networks don’t perform well in cases where sequence of data is important. For example: language translation, sentiment-analysis, time-series and more. To overcome this failure, RNNs were invented. RNN stands for “Recurrent Neural Network”. An RNN cell not only considers its present input but also the output of RNN cells preceding it, for it’s present output.

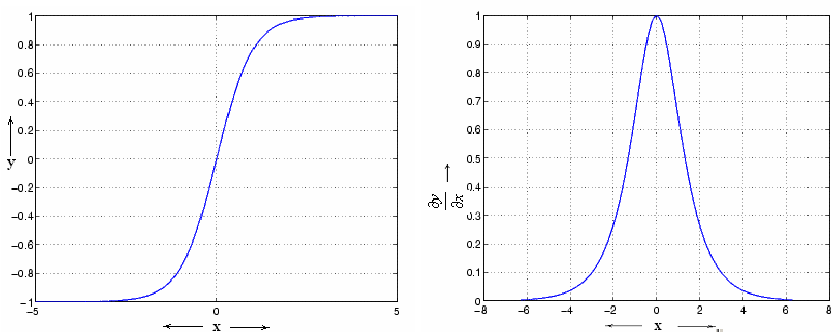
**Simple form of Vanilla RNN’s present state could be represented as**:



RNNs performed very well on sequential data and performed well on tasks where sequence was important.

**But there exists many problems with ordinary RNNs**

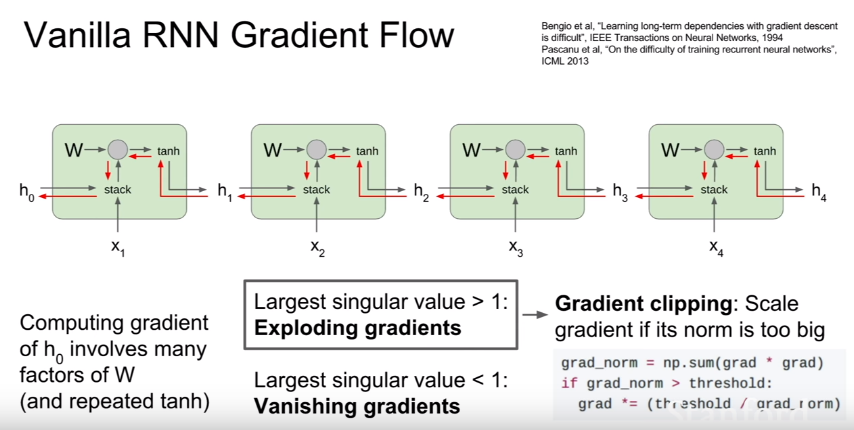
**Vanishing gradients problem:**



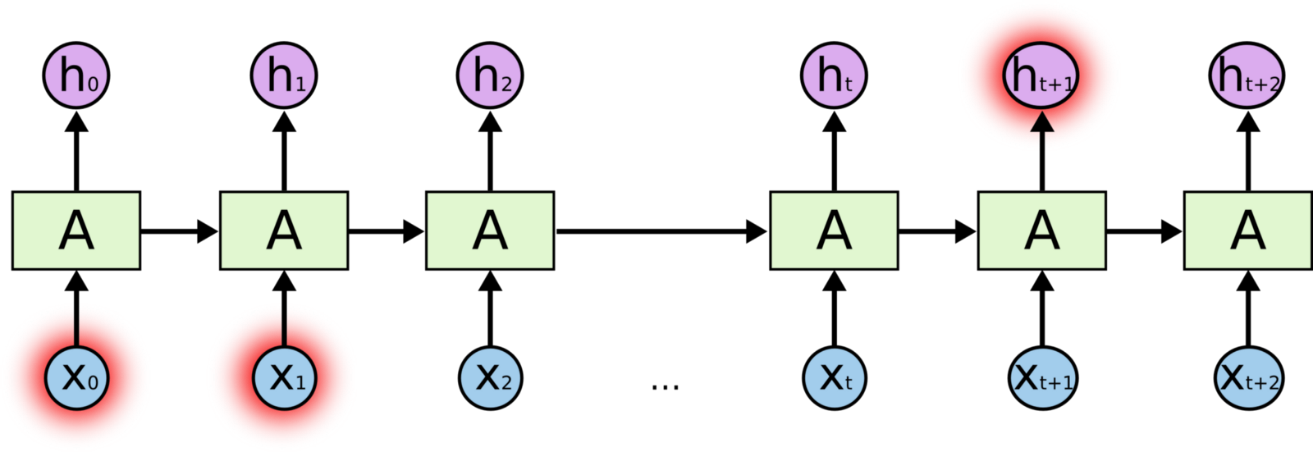
Hyperbolic tangent(tanh) is mostly used as activation function in RNNs which lies in [-1,1] and derivative of tanh lies in [0,1]. During backpropagation, as gradient is calculated by chain rule, it has an effect of multiplying these small numbers **n** (number of times tanh used in rnn architecture) **times** which squeezes the final gradient to almost zero and hence subtracting gradient from weights doesn’t make any change to them which stops the training of model.

**Exploding gradients problem:**

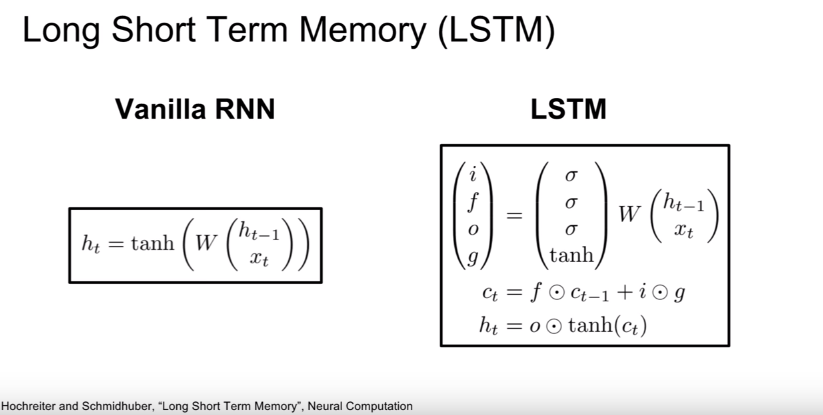
Opposite to vanishing gradient problem, while following chain rule we multiply with the weight matrix(transposed W )too at each step, and if the values are larger than 1, multiplying a large number to itself many times leads to a very large number leading to explosion of gradient.



**Long-Term Dependencies problem**



RNNs are good in handling sequential data but they run into problem when the context is far away. **Example: I live France and I know \_\_\_\_.** The answer must be ‘**French**’ here but if the there are some more words in between ‘**I live in France**’ & ‘**I know \_\_\_\_**’. It’ll be difficult for RNNs to predict ‘French’. **This is the problem of Long-Term Dependencies.** Hence we come to LSTMs.

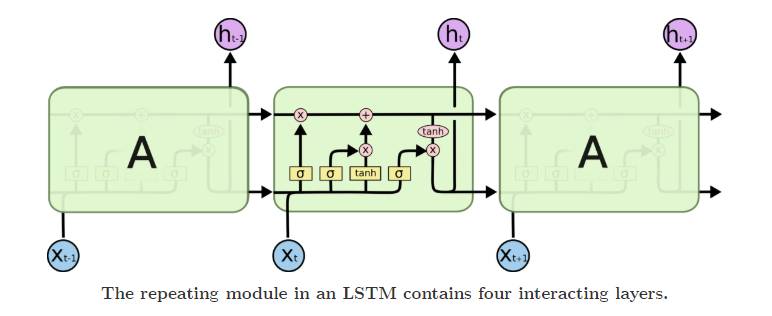


**RNN vs LSTM cell representation,** source: stanford

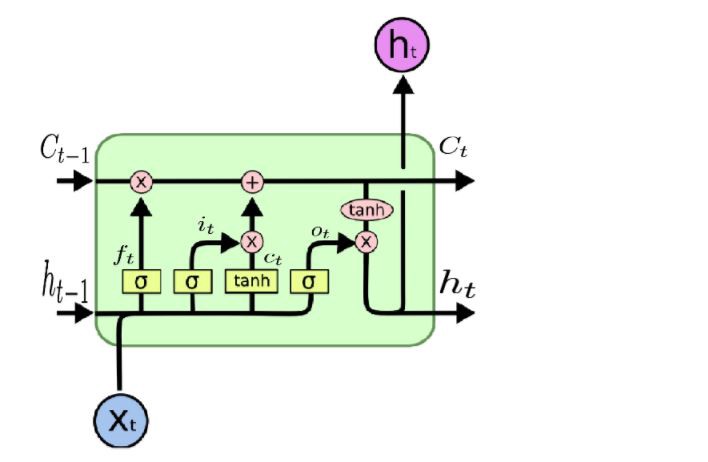
At start, we need to initialize the weight matrices and bias terms as shown below.

**Some information about an LSTM cell**

A simple LSTM cell consists of 4 gates:



1. **LSTM cells connected to each other.** source:Google



# 2.3 Application:-

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

We are focusing on Fake news detection as application.

# Chapter 3

# Algorithm for LSTM :-

# TASK #1: IMPORT LIBRARIES AND DATASETS

1). !pip install --upgrade tensorflow-gpu==2.0

2) !pip install plotly

!pip install --upgrade nbformat

!pip install nltk

!pip install spacy # spaCy is an open-source software library for advanced natural language processing

!pip install WordCloud

!pip install gensim # Gensim is an open-source library for unsupervised topic modeling and natural language processing

import nltk

nltk.download('punkt')

import tensorflow as tf

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from wordcloud import WordCloud, STOPWORDS

import nltk

import re

from nltk.stem import PorterStemmer, WordNetLemmatizer

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize, sent\_tokenize

import gensim

from gensim.utils import simple\_preprocess

from gensim.parsing.preprocessing import STOPWORDS

# import keras

from tensorflow.keras.preprocessing.text import one\_hot, Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Flatten, Embedding, Input, LSTM, Conv1D, MaxPool1D, Bidirectional

from tensorflow.keras.models import Model

from jupyterthemes import jtplot

jtplot.style(theme='monokai', context='notebook', ticks=True, grid=False)

# setting the style of the notebook to be monokai theme

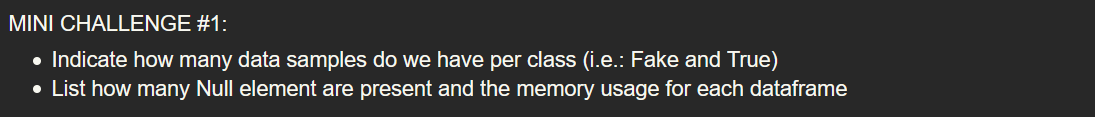
# this line of code is important to ensure that we are able to see the x and y axes clearly

# If you don't run this code line, you will notice that the xlabel and ylabel on any plot is black on black and it will be hard to see them.

3) # load the data

df\_true = pd.read\_csv("True.csv")

df\_fake = pd.read\_csv("Fake.csv")

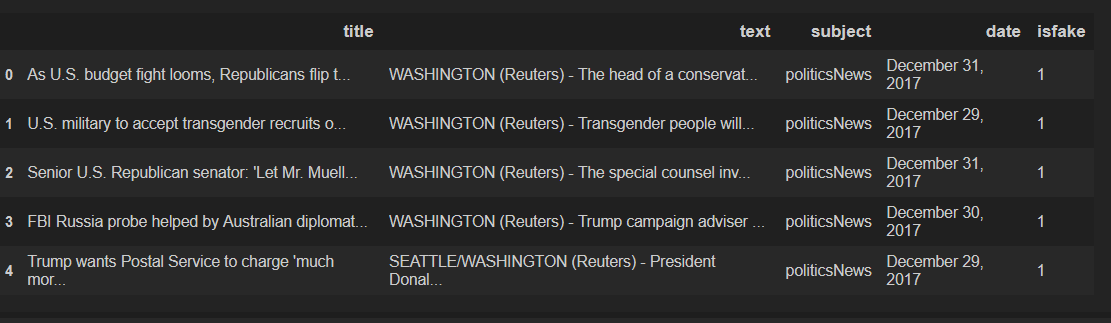


# TASK #3: PERFORM EXPLORATORY DATA ANALYSIS

1) # add a target class column to indicate whether the news is real or fake

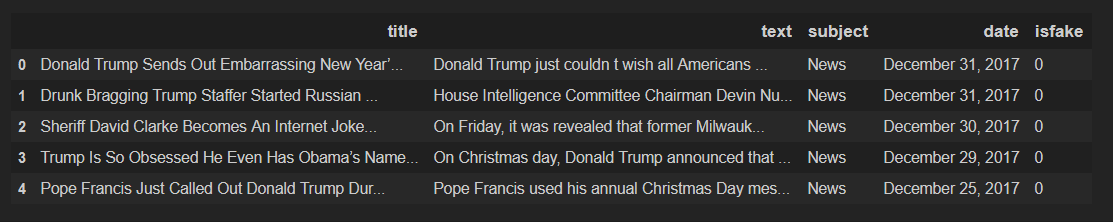
df\_true['isfake'] = 1

df\_true.head()



2) df\_fake['isfake'] = 0

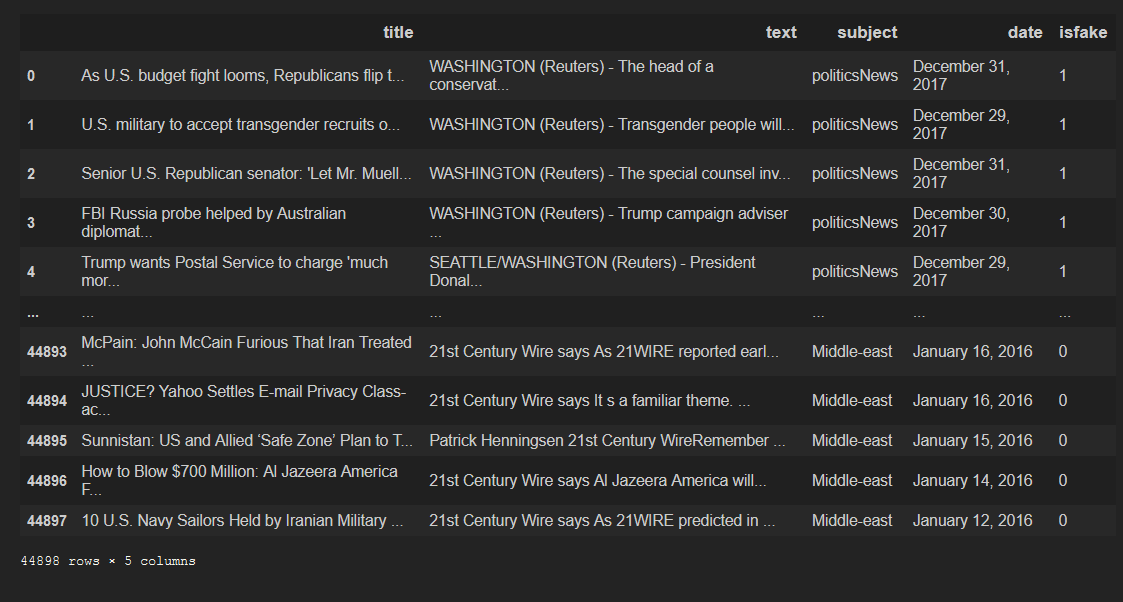
df\_fake.head()



3) # Concatenate Real and Fake News

df = pd.concat([df\_true, df\_fake]).reset\_index(drop = True)

df

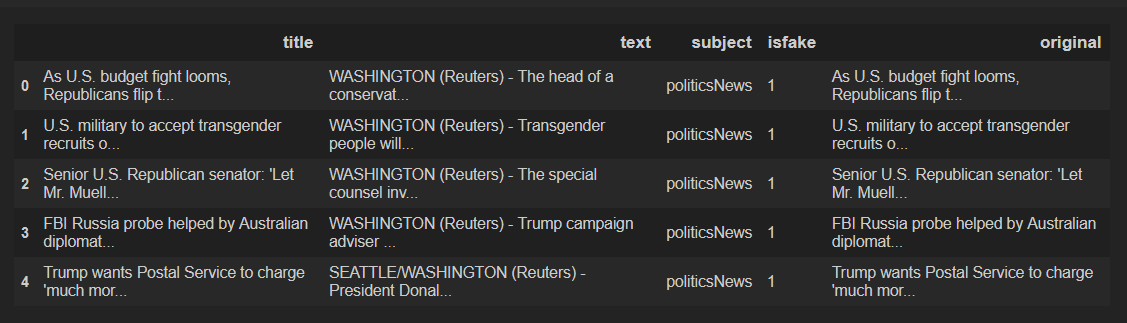


4) df.drop(columns = ['date'], inplace = True)

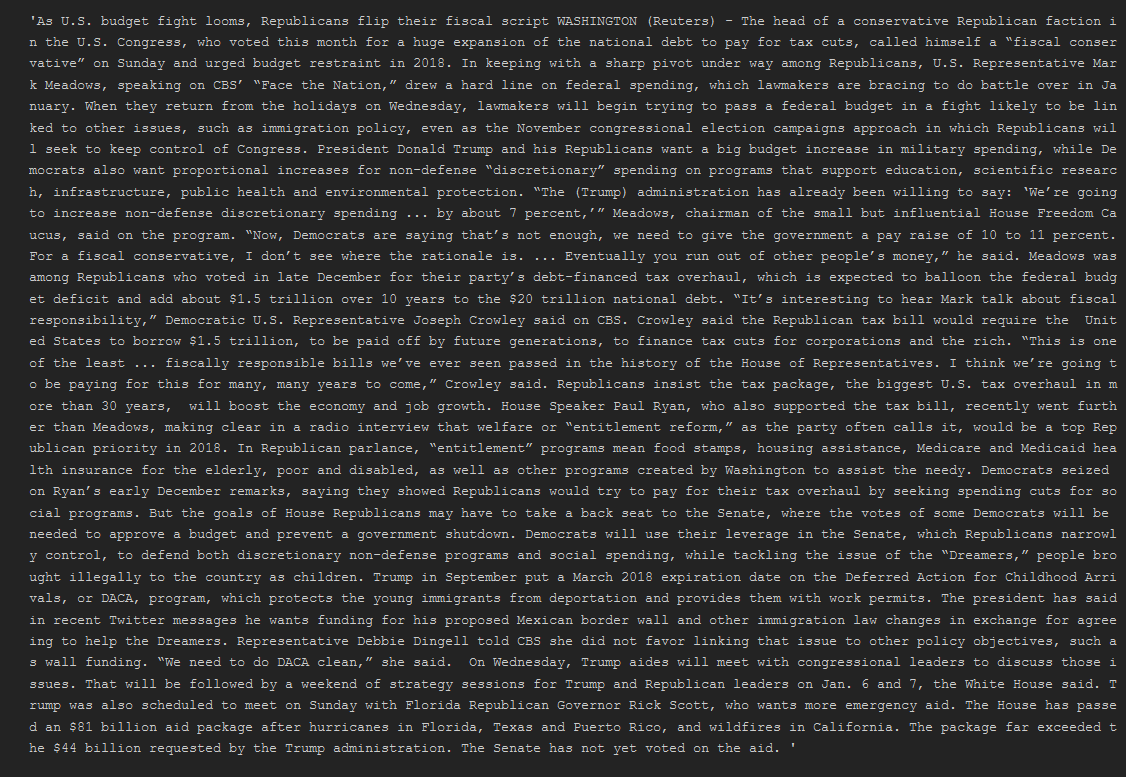
5) # combine title and text together

df['original'] = df['title'] + ' ' + df['text']

df.head()



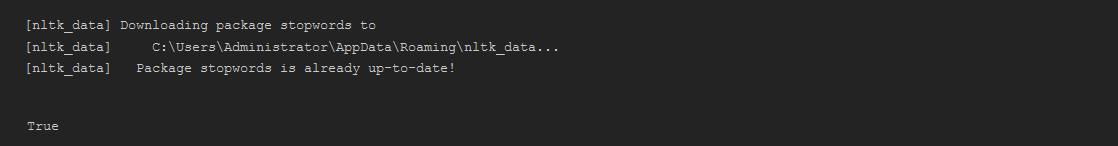
5) df['original'][0]



# TASK #4: PERFORM DATA CLEANING

1. # download stopwords

nltk.download("stopwords")



1. # Obtain additional stopwords from nltk

from nltk.corpus import stopwords

stop\_words = stopwords.words('english')

stop\_words.extend(['from', 'subject', 're', 'edu', 'use'])

1. # Remove 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 and token not in stop\_words:

result.append(token)

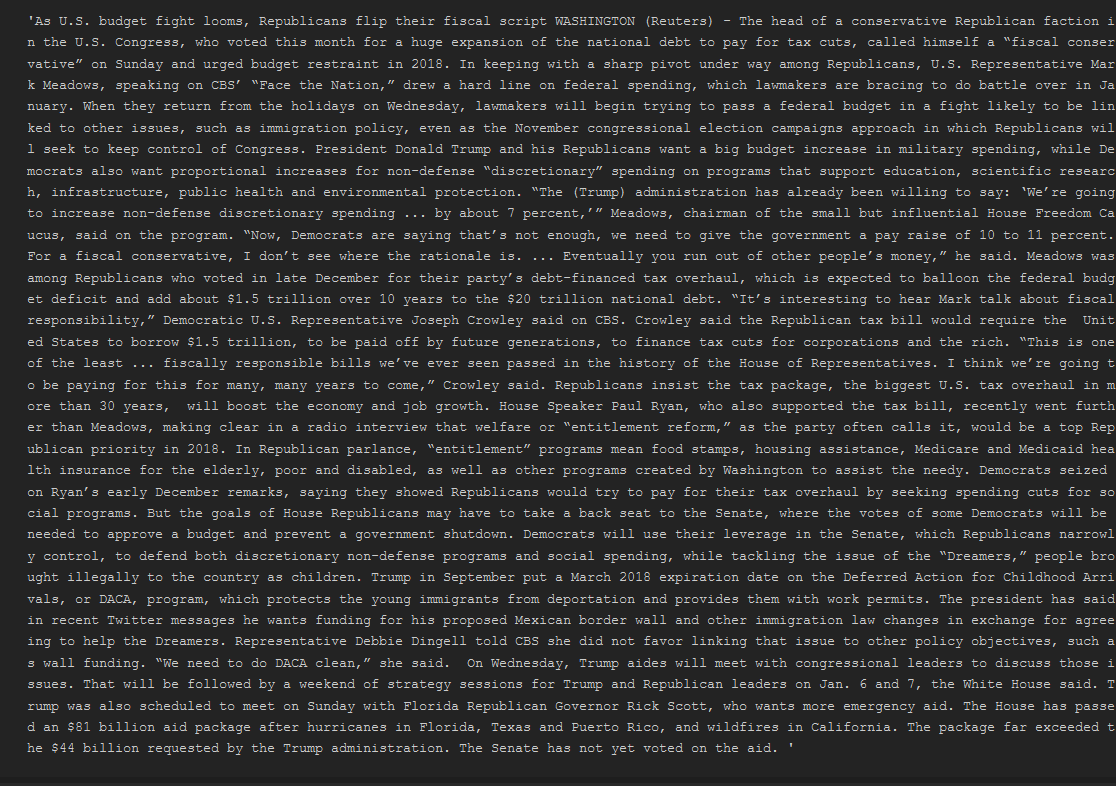
return result

1. # Apply the function to the dataframe

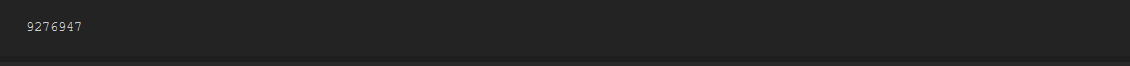
df['clean'] = df['original'].apply(preprocess)

1. # Show original news

df['original'][0]



1. len(list\_of\_words)



1. # join the words into a string

df['clean\_joined'] = df['clean'].apply(lambda x: " ".join(x))

1. df

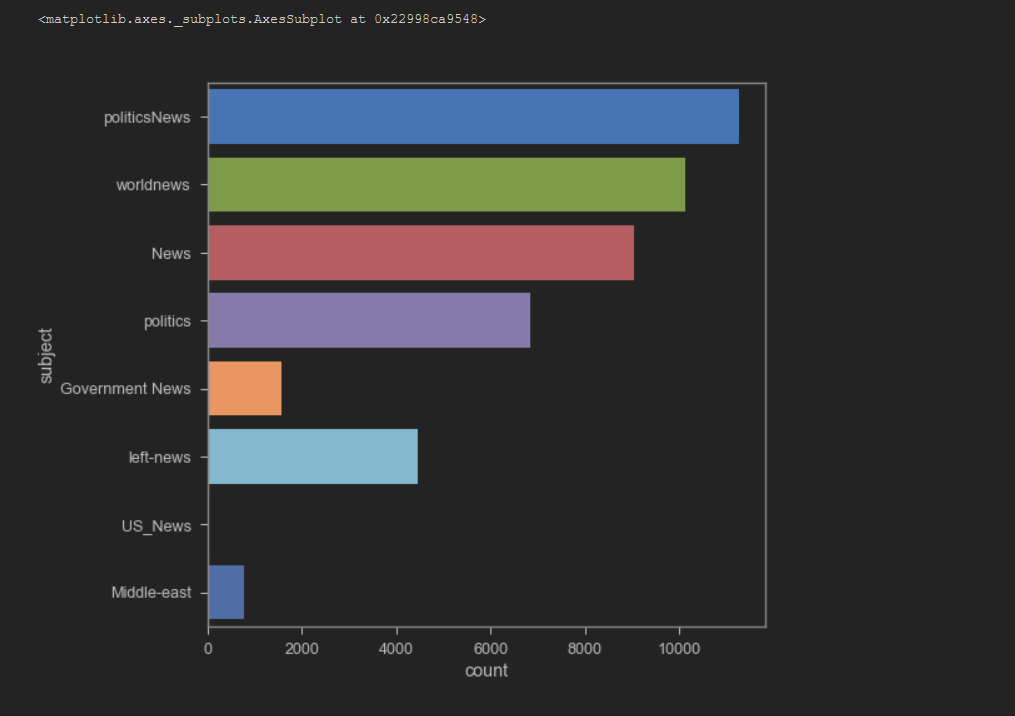


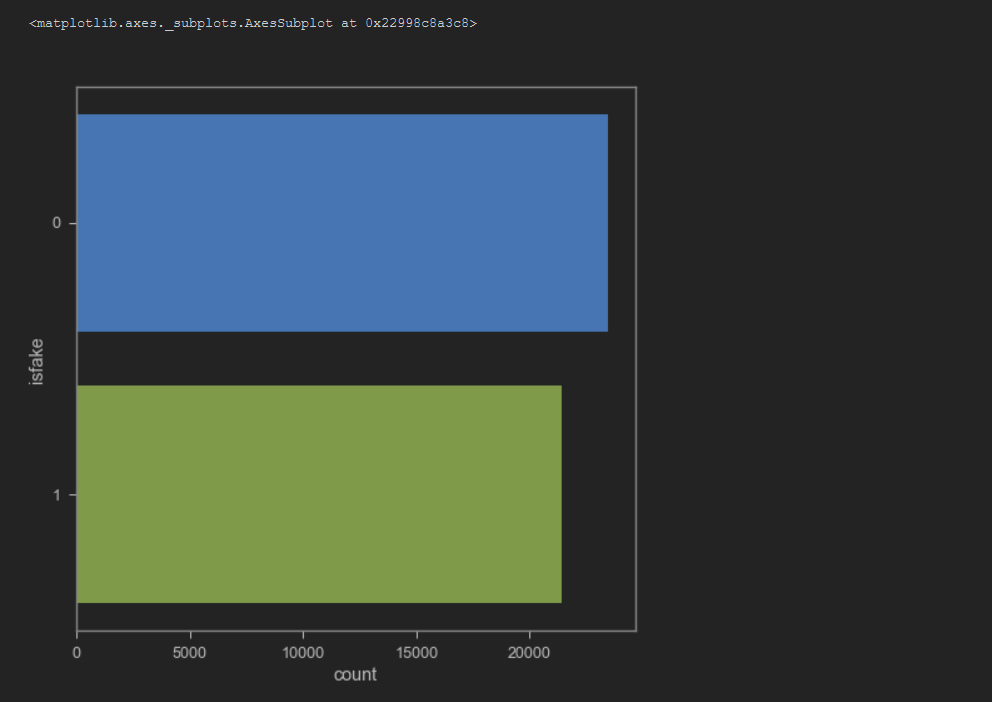
# TASK #5: VISUALIZE CLEANED UP DATASET

1. Df
2. # plot the number of samples in 'subject'

plt.figure(figsize = (8, 8))

sns.countplot(y = "subject", data = df)



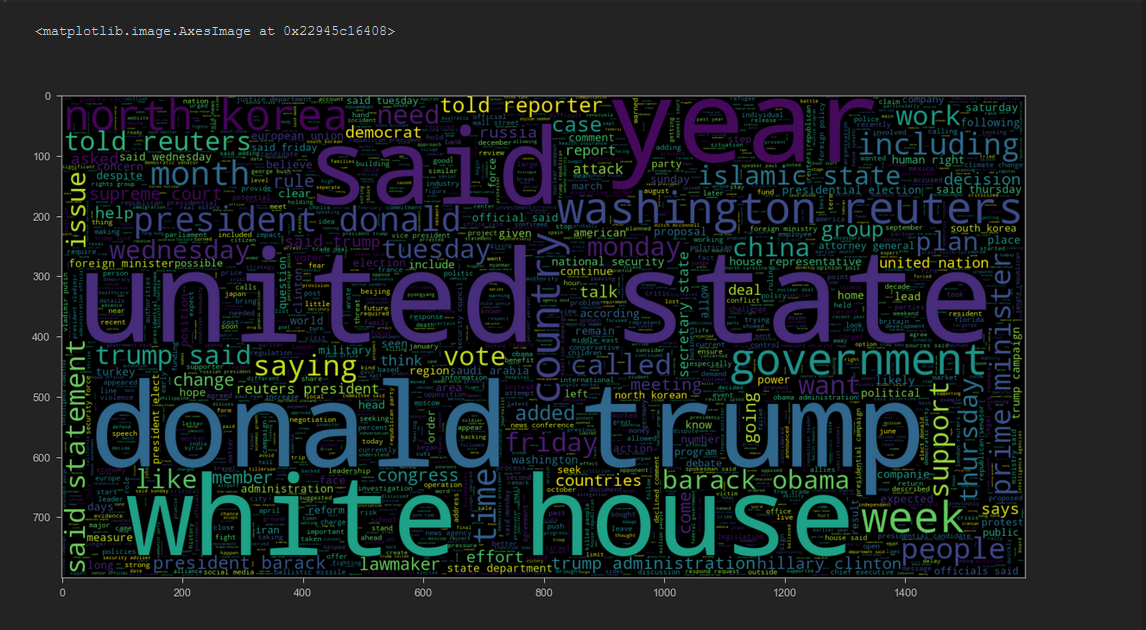


1. # plot the word cloud for text that is Real

plt.figure(figsize = (20,20))

wc = WordCloud(max\_words = 2000 , width = 1600 , height = 800 , stopwords = stop\_words).generate(" ".join(df[df.isfake == 1].clean\_joined))

plt.imshow(wc, interpolation = 'bilinear')

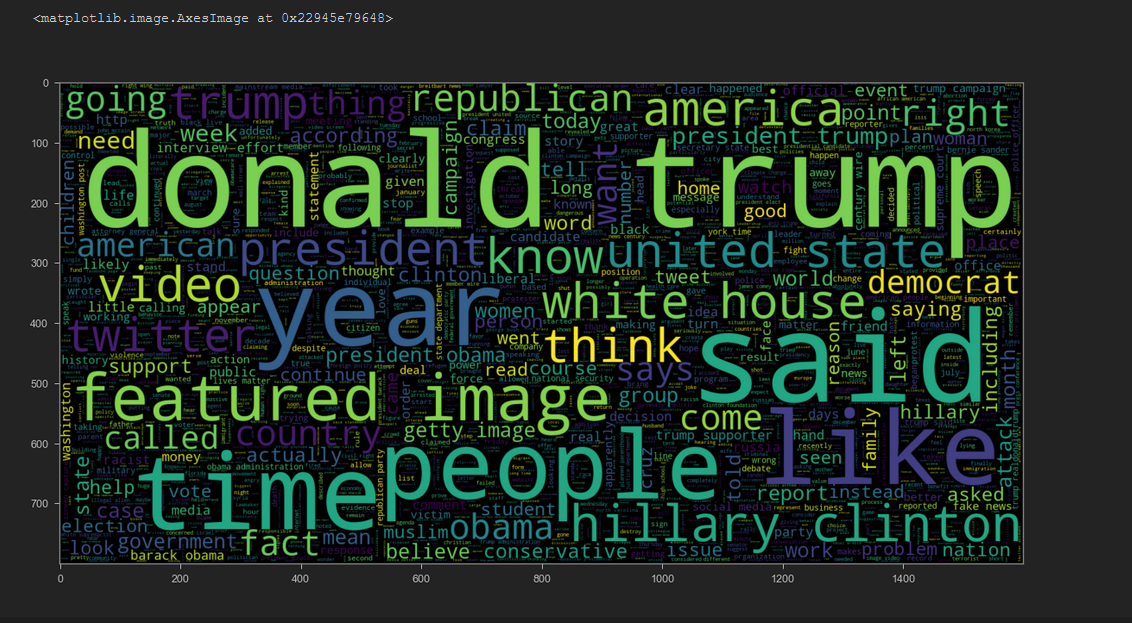


1. # plot the word cloud for text that is Fake

plt.figure(figsize = (20,20))

wc = WordCloud(max\_words = 2000 , width = 1600 , height = 800 , stopwords = stop\_words).generate(" ".join(df[df.isfake == 0].clean\_joined))

plt.imshow(wc, interpolation = 'bilinear')



1. # length of maximum document will be needed to create word embeddings

maxlen = -1

for doc in df.clean\_joined:

tokens = nltk.word\_tokenize(doc)

if(maxlen<len(tokens)):

maxlen = len(tokens)

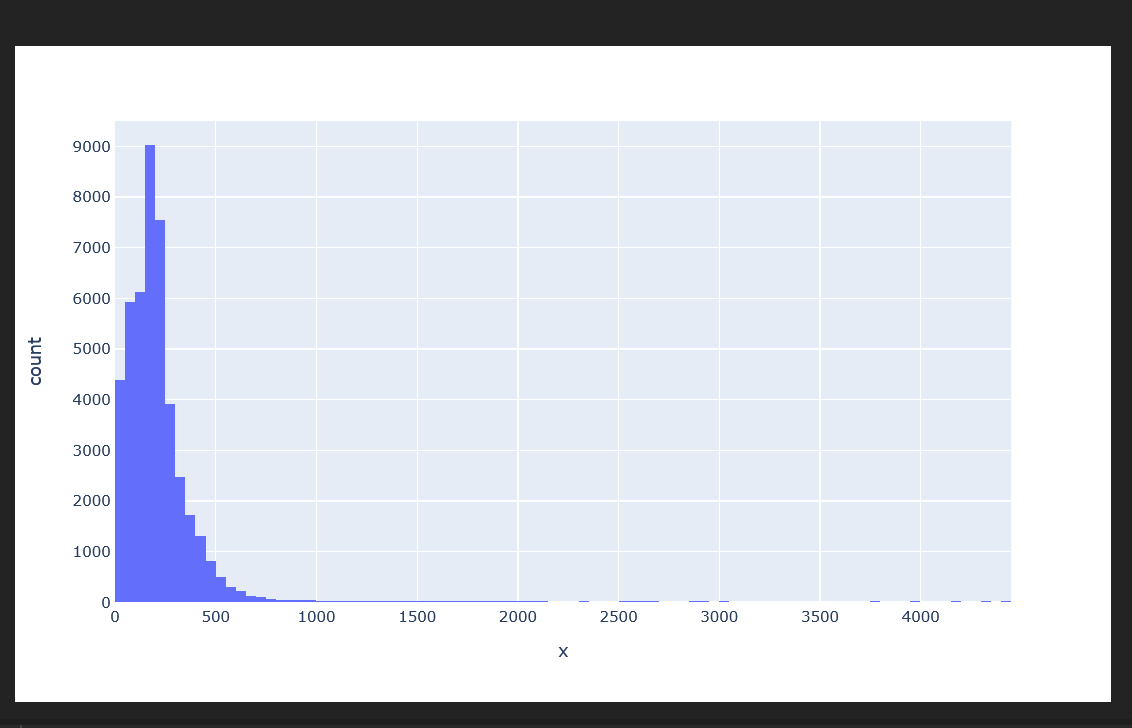
print("The maximum number of words in any document is =", maxlen)

1. # visualize the distribution of number of words in a text

import plotly.express as px

fig = px.histogram(x = [len(nltk.word\_tokenize(x)) for x in df.clean\_joined], nbins = 100)

fig.show()



# TASK #9: ASSESS TRAINED MODEL PERFORMANCE

1. # make prediction

pred = model.predict(padded\_test)

1. # if the predicted value is >0.5 it is real else it is fake

prediction = []

for i in range(len(pred)):

if pred[i].item() > 0.5:

prediction.append(1)

else:

prediction.append(0)

1. # getting the accuracy

from sklearn.metrics import accuracy\_score

accuracy = accuracy\_score(list(y\_test), prediction)

print("Model Accuracy : ", accuracy)

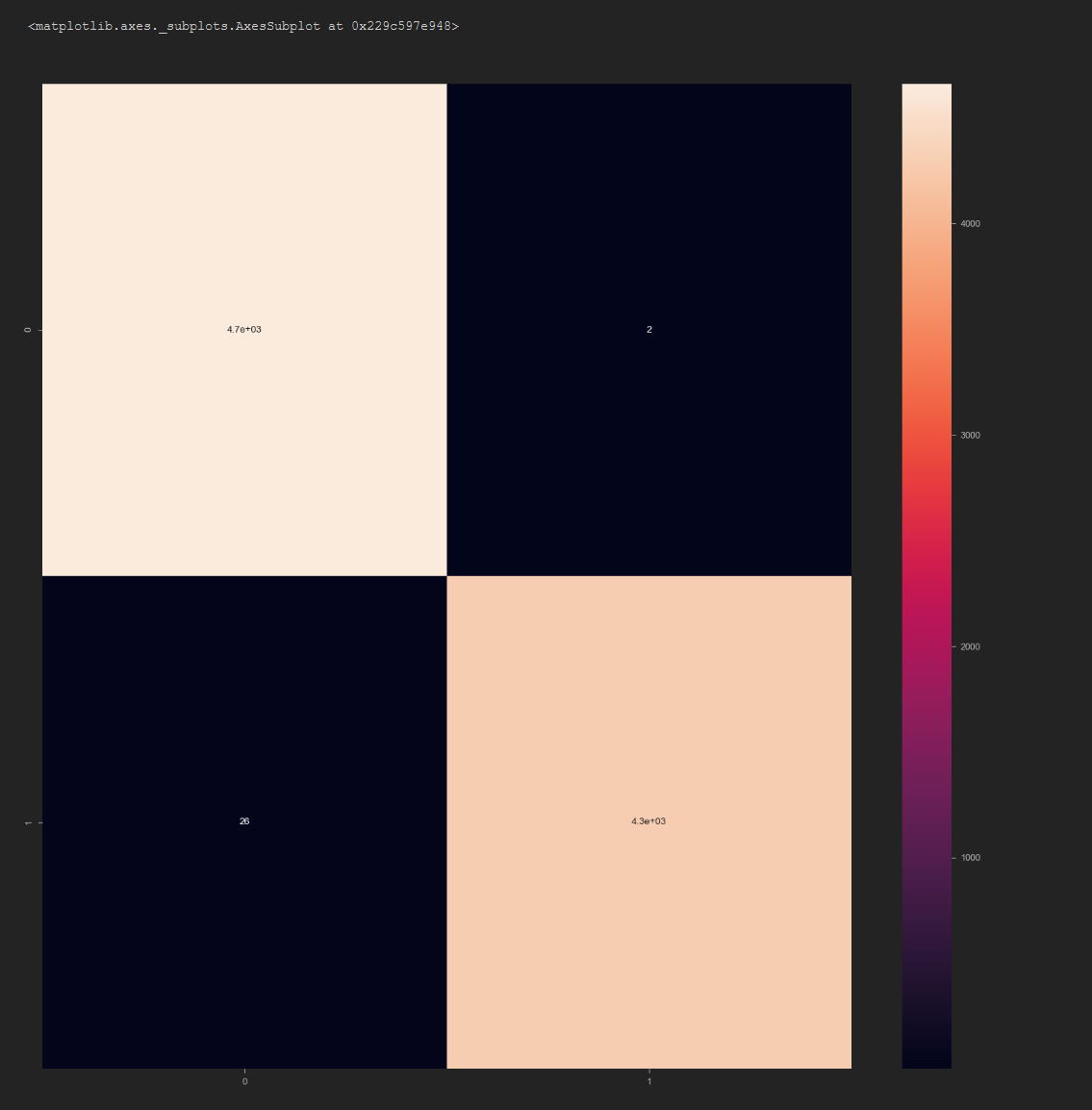
1. # get the confusion matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(list(y\_test), prediction)

plt.figure(figsize = (25, 25))

sns.heatmap(cm, annot = True)



1. # category dict

category = { 0: 'Fake News', 1 : "Real News"}

# Chapter 4

# Conclusion:-

The increasing number of fake news online are a danger for societies and governments as has been already probed. Moreover, the yearly increasing number of hoaxes and the emergence of the artificial intelligence trained to generate texts denote the need of strong systems oriented to discriminate the deceptions. In this work, we have proposed three novel architectures applied to textual analysis for fake news detection. Two of those created, optimized and trained from scratch and the last one fine-tuned from BERT, a pre-trained language model which achieved state of the art results in a great number of NLP tasks. Although there isn’t any benchmark available aimed at evaluating the task of fake news detection, the models here presented outperform the results in the original work which compiled the dataset used [13] (93% of accuracy) and get superior metrics compared to all the other related ones (see Section 2). This allow us to presume that it’s possible to train neural networks focused on detecting fake news merely using textual features but also that results at the state-of-the-art level can be reached through the strategies proposed. The experience gained during the development of this models allows us to state that the use of deep learning models for this task can be potentially beneficial for a wide range of actors, from social network companies to the final user in order to mitigate the increasing deceptions on the Internet. Regarding the future improvement of these models, firstly, it is mandatory to collect more data, specially from a recent period of time. This is also proposed by the researches who compiled the TI-CNN dataset, as the news there are mostly

obtained during the US electoral campaign. In order to accomplish the above, a system to automatically collect quality news should be developed. Finally, with the aim that these models can be used by the people, some method of serving them to users is also necessary (integration with social networks, browser extensions, mobile apps...). With these lines of work in mind we expect that our system can be ready to be successfully used in the real world.

# Chapter 5

# References:-

[1] Statista. Which, if any, is your main source of news? https://www.statista.com/statistics/198765/

main-source-of-international-news-in-selected-countries/. Online; Accessed on 03-29-2019.

[2] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. Technical report, National

Bureau of Economic Research, 2016.

[3] Alexandre Bovet & Hernán A. Makse. Influence of fake news in twitter during the 2016 us presidential election.

Nature Communications, 10(7), 2019.

[4] Max Boot. Without the russians, trump wouldn’t have won. https://www.

washingtonpost.com/opinions/without-the-russians-trump-wouldnt-have-won/2018/07/

24/f4c87894-8f6b-11e8-bcd5-9d911c784c38\_story.html?utm\_term=.ac812004f1e2, 2018. Online;

Accessed on 03-29-2019.

5] Eugenio Tacchini, Gabriele Ballarin, Marco L. Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it

hoax: Automated fake news detection in social networks. 2017.

[6] Nguyen Vo and Kyumin Lee. The rise of guardians: Fact-checking url recommendation to combat fake news,

2018.