Detection Of Fake News using Deep Learning

CS4098 Project
Final Report

Submitted by

Justine P Devasia (B170212CS)
Midhun Martin (B170149CS)
Shuhaib K.S (B170150CS)

Under the Guidance of

Dr. Jay Prakash
Assistant Profesor

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

May, 2020

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that this is a bonafide report of the project work titled

DETECTION OF FAKE NEWS USING DEEP LEARNING

done by

Justine P Devasia(B170212CS)
Midhun Martin (B170149CS)
Shuhaib K S (B170150CS)

of Eighth Semester B. Tech, during the Winter Semester 2020-'21, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology Calicut.

Dr Jay Prakash
Assistant Professor
Project Guide

Dr. Murali Krishnan K
Head of the Department

Date

DECLARATION

We hereby declare that the project titled, **Detection Of Fake News using Deep Learning**, is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place :
Date :

Name : Midhun Martin
Roll. No. : B170149CS

Name : Shuhaib K.S
Roll. No. : B170150CS

Name : Justine P Devasia
Roll. No. : B170212CS

Abstract

Fake news is defined as a fabricated story to mislead or deceive. This paper tries to find a solution to tackle the situation by identifying fake news using deep learning architecture.

The rise in news reading on social media is getting popular day by day, so is the creation and spreading of fake news. It can take advantage of multimedia content to mislead readers and create false stereotypes among people in various fields like politics, religion, etc. Nowadays, the amount of fake news generating from social media is increasing exponentially. This exponential increase of inaccurate news contents poses a need for automatic tagging and detecting such malicious content. However, detecting fake news using automated methods is a challenging task to accomplish as it requires the model to understand nuances in natural language.

In this project, we propose a model based on deep learning that can automatically detect fake news. We use various Natural language processing(NLP) techniques to vectorize the news text. We also use two separate CNNs to extract latent features from image and text data, combined with some explicit features to make predictions.

Contents

Chapter 1

Introduction

Any untrue information presented as news is termed Fake News. It is often spread through non-traditional media channels like social media. Such news is often created with the intent to damage the reputation of an entity or a person. Such news is often written to manipulate the beliefs and decisions of people and can affect mass events like political elections. Since people are often tempted to read fake news, such news can also be used to generate revenue by clicking. Many social media users, who blindly believe news online, provide a breeding ground for fake news. Readers are often attracted to fake news like conspiracy and pitfalls, making them want to share such information. Also, nowadays, most people read the news by browsing through the headlines without reading the content at all. Therefore whenever fake news contents are published with fancy headlines, people usually trust them.

Today, most people consume news through social media, and unsurprisingly, fake news is mostly spread through social media. It is due to the fact that social media platforms like Facebook and Twitter provide a space for the general public to share their views and opinions. Fake news is also spread faster than real news. One Research that studied the spread of fake news concluded that false tweets reach people six times faster than truthful tweets.

Also, some articles shared on social media even get more views than the official media websites.

Hence it is of utmost importance to stop spreading such fake news on social media as it has adverse effects. In this project, we formulate a solution to this spread of fake news by building a model that can automatically detect fake news and prevent such news from appearing on user's news feeds. We propose a fake news detection model which can detect fake news by accurately predicting the stance between the headline and the news article. Images in fake news are observed to be either irrelevant to the topic or some misleading ones. Hence, we also gather information of the image attached to the news in detecting the given news as fake. Our model (FN-CNN) uses both the image and text content present in the news article to predict whether the article is fake or not. FN-CNN uses multi-model feature extraction to extract visual and textual features separately and then use the cosine similarity measure to classify the news article accurately.

Chapter 2

Problem Statement

Given a news article, consisting of headline, news content and image attached with it, the task is to classify the news as fake or not depending on the given visual and textual features. The textual features comprises of the news headline and content of the news and the visual features comprises of the features of the image attached with the news. Together with these features and relationship between them we have to predict the given news article as fake or not by constructing a deep neural network model.

Chapter 3

Literature Survey

Much research has currently been done in the field of fake news detection. This includes various tasks such as rumor detection and spam detection.The main task is how to classify a news as a fake or not according to features. News features can be extracted in several ways, such as title, content, attached image, the user who posted, etc. Stance Detection is a popular and well-researched task in Natural Language Processing.It is about finding the stand of the audience that they are against,neutral or against the news. Detection of fake news is also a stance detection problem where stance detection of the news text body is relative to its attached image. We were inspired and more inclined by the following remarkable conclusions made by the related work available.

The first challenge in this problem of fake news identification is the choice of features. The various features that can be considered for this problem is discussed in [1]. The authors has classified the various features into textual features, visual features, post related features, user related features. And various tools that can be employed to extract these features are also discussed. Thus, we must choose those features that can help our model to predict the result with higher accuracy.

In [2] the authors used only user characteristics to detect the fake news. The characteristics of users who posted as well as retweeted or shared a news article was taken into consideration. A multivariate time series was modeled based on the propagation path of each news content with characteristics of a user engaged in propagating news with a numerical vector. Then a CNN and RNN which is aimed to analyse and find the global and local variation of user characteristics along the path is incorporated into a time classifier that predicts the news as fake or not. However the body of the news was not taken into considerations in their work.

In [3] the authors present neural network architecture to accurately predict the news as fake or not by including the news content as well as a similarity measure between headline and body. They were able to achieve an accuracy of 0.8421 on test data. They employed the technique of "Term Frequency-Inverse Document Frequency" to vectorise the news text after its processing has been done. This increased their accuracy. A measure indicating the relatedness between the news headline and news body was also considered.

In [4] the authors proposed a model named as TI-CNN (Text and Image information based Convolutional Neural Network). This model is trained with both the text and image information by projecting the explicit and latent features into a unified feature space. The method of considering the features of image attached with the news was found successful. Although they did not consider the relationship between the image and news body, their model predicted the news with higher accuracy.

A tweet-level feature "has multimedia" is defined in [5], to record the status of multimedia attachment of a tweet: whether the tweet has any picture, video or audio attached. Gupta et al. [6] make an effort to understand the temporal, social reputation and influence patterns for the spreading of fake images on mircoblogs. They propose a classification model to identify the fake images on Twitter during Hurricane Sandy. Some interesting conclusions are drawn: The original fake images are limited, and 86% of fake images were re-tweets. These conclusions meet our assumption that images in the fake news are less diverse and limited in amount. However, their work is still based on traditional text, user, and propagation features.

Fake news, nowadays, is more accompanied by an image that is aimed to gain attention of the reader. Hence image features are also to be considered. According to the study conducted by the authors of [7], they figured that images are very popular and have a great influence on the propogation of fake new in microblogs. They also found out that fake and real news images have several different features attached to them. Hence serveral features were taken into consideration by them to determine an image as fake or not. Experiments were done on SinaWeibo dataset and its results emphasises the effectiveness of their proposed image features. The features chosen for classification included Visual Clarity Score, the distribution difference between event image set and collection image set, Visual Coherence Score, the average of visual similarities between image pairs in event image set, Visual Similarity Distribution Histogram, the distribution histogram of the image visual similarity matrix, Visual Diversity Score, the weighted average of visual dissimilarities between image pairs, Visual Clustering Score, the number of image clusters after cluster images with visual patterns. Considering the technology of time they got higher accuracy. They used machine learning algorithms like logistic regression, random forest, kstar for the classification.

Works are also done in classification of fake news just by considering the features related with image. The authors of [8], propose a novel framework Multi-domain Visual Neural Network (MVNN) to combine the information of frequency and pixel domains for detecting fake news. They designed a CNN to capture the features of the fake images. Their emphasis was only on images. They also used a CNN-RNN model to extract various semantic level features in the pixel domain. An attention mechanism is also then used to concatenate the vector representations of pixel and frequency domains dynamically.

Hence it can be concluded that apart from textual features, image features can also aid in the fake news detection problem. Also deep learning technology gives higher accuracy than traditional machine learning algorithms. Extraction of textual and image features can be done using deep learning techniques. We are employing deep learning technology like Neural Networks for the prediction.

Chapter 4

Methodology

4.1 Dataset

The dataset we used contains 13830 news, i.e., 5470 fake news and 8360 real news. It is available online. The dataset contains text and metadata scraped from 240 different websites which includes popular authoritative websites like New York Times, Washington Post, etc (used to collect real news) and some fake news websites (which were scraped to collect fake news samples). The dataset contains all the information related to the news article such as its title, content, image, date, author, etc. But we only choose those columns that are necessary for us and they are the title, which is the title of the news, the content that displays the content of the news, and the url of the image which is the url of the image attached to the news title.

4.2 Data cleaning

The data is looked up for null or missing values. If the title of a news item is missing it is replaced with 'No heading'. On analyzing the dataset there were only a few headlines missing from the data.

4.3 Feature extraction

The Next task is to extract the features required from the dataset. The dataset contains news headline, news content and image url. We should be able to extract some meaningful features from these information available in the dataset. In this paper we are dividing the features into textual features- features extracted from the text content of the news mainly headline and content, and, image features- features extracted from the image attached with the news. In this section we are discussing about the features that can be explicitly derived from the dataset. These features are easy to observe. Such features can be termed as explicit features.

4.3.1 Text features

Textual features can be obtained from the following areas of observations:

1. Computational Lingustic:

   (a) Number of words and sentence: On analysing the dataset and plotting the information, it was observed that, on an average fake news contained fewer words than real news. We found that on an average fake news contained around 3,943 words whereas real news contained 4,360 words. We also found that the number of words in fake news distributed over a wide range. It was also observed that fake news has larger sentences when compared to real news. Therefore number of words can be used as an explicit feature to distinguish between fake and real news.

   (b) Question mark, exclamation and capital letters: Because of the fact that fake news often contain many rhetorical questions (usually used to intensify the sentiment), counting the number of question marks might give us some insight. Statistics shows that fake news usually contains more questions than real news. Similarly fake news was also found to contain more number of exclamation marks than real news. Similarly, on an average fake news was found to contain more capital letters compared to real news. This is because, the authors of fake news usually use more capital letters to emphasize their idea.

(c) Cognitive perspective: The authors of real news often use more negations when compared to the authors of fake news. This is because the authors of fake news often try to avoid the use of negations in order reduce their chance of being caught [9].

2. Psychology Perspective: We found that the authors of fake news often prevent the usage of first-person pronouns like I,we,...etc. This is because people who are lying often minimize the reference to themselves [10]. Similar is the case of second-person pronouns and third-person pronouns. Similarly, the authors of fake news use fewer motion verbs. This is because unlike the authors of real news, they do not know the details of the event.

4.3.2 Image features

The explicit features that can be derived from image are not many. The features that we can observe directly are size of image expressed in pixels and number of faces in the image. We found that on an average, real news contained more number of faces than that of fake news. Similarly the images found in fake news articles were of lower resolution when compared to that of real news images.

4.4 Model - The architecture

So far we have obtained only explicit features. These features are not sufficient to build a good model. Hence more features must be extracted from the news content and image. In order to extract more features we employ two CNN - one for extracting text features called as Text CNN and the other to extract image features termed as Image CNN. Later, we will discuss how to concatenate all these features. We will be discussing both these CNN in detail.

4.4.1 Text CNN

Apart from the explicit features that we obtained above we require more features that can be obtained from the text content of the news. The challenge here is that news content is expressed in English language, rather a language that is not understandable to the machine. Hence it is necessary to convert this to a machine readable format.

1. Data Preprocessing

We often need to preprocess the text data before using any machine learning algorithms on them. There are various techniques which can be used to convert text data into a form that is ready for modeling. We also provide insights into different word vector representations we used as part of our analysis.

Stemming and Lemmatization

Stemming is one of the most common preprocessing step. In Stemming the word is reduced to its root form. This is a basic requirement for all NLP tasks and must be done before any word vectorization techniques are applied. Another such technique which can be used is Lemmatization. Lemmatization works similar to stemming, the key difference being that lemmatization considers the morphological analysis of the words. For example, a word “studies” when stemmed would return “studi” whereas when lemmatized would return “study”. Although it might seem that lemmatization does a better job at reducing the word to its root form, they both usually give similar results.

For our implementation we used stemming.

Stop Word Removal

Stop Words are words in the sentence which occur frequently and does not provide much contextual meaning to the sentence.The advantage of removing these words are the size of the dataset decreases and model performance increases.Some examples of stop words are “a”, “and”, “but”, “how”, “or”, and “what”.

2. Feature Engineering

Feature Engineering is the intelligent application of domain knowledge to extract valuable information from raw data.It helps in increasing the accuracy of model for better prediction.In this project we will apply various word vectorization techniques like tf-idf,Bow,Glove,Word2Vec to effectively convert word to vectors. It also include other process like generation of explicit features. Reshaping of the word vectors to the required shape also includes feature engineering.

TF-IDF

TF-IDF is a word vectorization technique which statistically measure the relevance of a word with respect to a document in a set of documents.This is done by multiplying two metrics.Term-Frequency and Inverse Document Frequency.Term Frequency is the count of words appearing in a document.Inverse Document Frequency is defined as the log to the base e of number of the total documents divided by the documents in which the word appears.

Bow

A bag of words is another word vectorization of text that describes the occurrence of words within a document.The model assigns a vector to a document as d = (x1, x2, . . . , xn),where xi denotes the normalized number of occurrence of the i-th most frequent term and n is the size of the collection of that terms. We only keep track of word counts and disregard other details like order and grammar.It is not concerned with the location of word in the document and checks only the occurence of a word.

Word2Vec

Word2Vec is a widely used algorithm based on neural networds which has many applications in machine translation and recommender systems. Word2Vec model is made up of three parts. They are the Vocabulary Builder, Context builder and a neural network. The Word2Vec model comes in two forms, namely Skip-grams(SG) and Continuous-bag-of-words (CBOW). For our implementation we used the gensim library to train the Word2Vec model with CBOW on our dataset. By doing so the Word2Vec generates a vocabulary based on the words in our dataset. Using such a custom trained model actually gave better results than using the pretrained model. The main advantage of using Word2Vec over other methods is that a trained Word2Vec model successfully learns the similarity between words, i.e, when using Word2Vec similar words get represented by similar vectors. For our Word2Vec model we used a vector size of 300.

Text CNN Explained

After the content and headline of news are represented in terms of vectors of fixed size, it can be used as input to the text CNN. Apart from the explicit features explained above, it is necessary to extract more features from the content and headline of the news as well. This is done using the Text CNN. These textual features in the model are based on a variant of CNN. Such features can also be termed as latent features. CNNs are actually used in various fields of Computer Science like Computer Vision tasks. Also CNNs also show notable performance in areas such as image classification and object recognition and many Natural Language Processing tasks. This is because the convolution that are being employed in CNN can derive local features around each word with the help of nearby words and then combine these features using a max operation giving a fixed size of ouput which we call as word-level embedding.Thus a CNN which we call text-CNN is used to extract the textual features.

We extend Text-CNN [11] by introducing an additional fully connected layer to automatically extract textual features for each news article. The architecture of Text-CNN is explained below, which contains a convolutional layer and max pooling. Given a piece of content with n words, each word is first embedded as $x_t^l \epsilon R^k$, $l = 1,2,....,n$ i.e each word is represented as a vector of k dimensions. The feature map is then produced by the convolution layer, denoted as $C_t = \{c_t^i\}_{i=1}^{n-h+1}$, from a sequence of local inputs $\{x_t^{i:(i+h-1)}\}_{i=1}^{n-h+1}$, via a filter w_t. where each local input is referring to a group of h continuous words. Mathematically,

$$c_t^i = \sigma(w_t \cdot x_t^{i:(i+h-1)} + b_t)$$

$$x_{i:(i+h-1)} = x_i \oplus x_{i+1} \oplus .... \oplus x_{i+h-1}$$

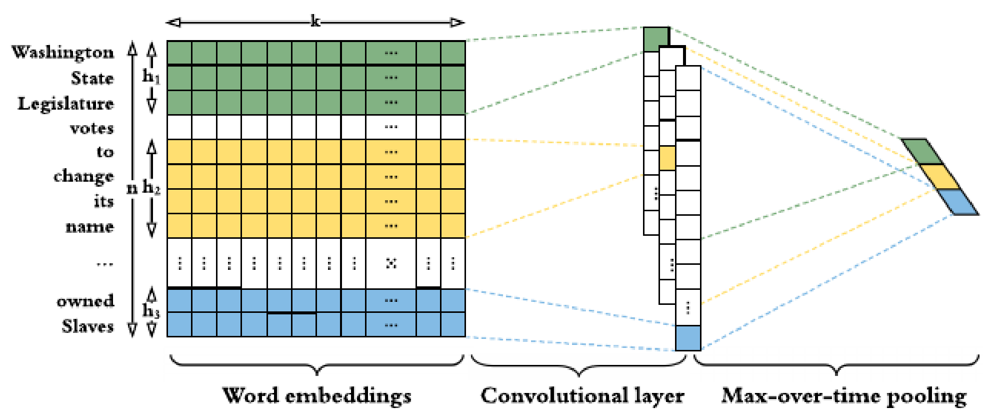


Figure 4.1: Text CNN[11]

, where $w_t, x_{i:(i+h-1)} \epsilon R_{hk}, b_t \epsilon R$ is a bias, $\oplus$ is the conctenation operator, and σ is ReLU function. Here w_t and b_t are the parameters within Text-CNN which requires to be learned. To the obtained feature map, we apply max-pooling operation so that we get a feature map with reduced dimension, i.e., $\hat{c_t} = max\{c_t^i\}_{i=1}^{n-h+1}$. Finally, the news text can be represented by $t = \mathbf{W}_t \hat{c_t} + \mathbf{b}_t$, where $\hat{c_t} \epsilon R^g$, g is the diferent number of window sizes chosen; $\mathbf{W}_t \epsilon R^{d \times g}$ and $\mathbf{b}_t \epsilon R^d$ are parameters to be learned.

Regularization

In order to prevent the dataset from overfitting we use regularization techinque on the second last layer of the Text-CNN. Regularization is done by randomly dropping out certain hidden values of the layer. It is defined in terms of probability p such that each hidden unit has a probability of p for being getting dropped. Mathematically we can write it as: Suppose $\mathbf{y} = [\hat{d_1}, ..., \hat{d_g}]$ is the second last layer, number of filter used is given by g, for output unit in y in forward propagation, dropout output is given by,

$$y = \mathbf{w} \cdot (\mathbf{y} \circ \mathbf{p}) + b$$

where $\mathbf{p}$ is the vector having one or zeros defined by probability p. $\mathbf{p}$ will decide what neurons to drop.

4.4.2 Image CNN

Similar to text features, the visual explicit features alone are not sufficient to train our model. To directly learn from raw images we also need to derive more powerful features. We use CNN to extract such features. An explanation to Image CNN is given below.

For extracting features of the images we used the same Text-CNN architecture with the same fully connected layer. Since Image cannot be directly fed to the Text-CNN, we must first find a way to convert image to text in the most appropriate and meaningful way possible. For this we used a pretrained *image2sentence* model [12]. This pretrained model generates the caption for our images and hence can be considered as the most meaningful textual representation of the image. Also using this pretrained model has an advantage over the existing multimodal fake news detection designs that uses pretrained CNN like VGG to exract image features. Also by using this caption generator has an increased advantage of being able to calculate similarity measure with the other modalities. As it will be shown later this includence of similarity measure will lead to improvement in performance measure. Let $\hat{c_v}$ denote the output of the above mentioned neural network with parameters w_v (filter) and b_v (bias). Similarly, the final representation of news visual information is then computed $v = W_v \hat{c_v} + b_v$ where W_v and b_v are parameters to be learned.

4.4.3 Similarity measures

Apart from considering the multimodal featues alone in deciding the truth value of the news articles, realtionship between various modalites can also be included. Our results shows an increase in performance measure on including such relationship measures. In our design two different similarity measures can be included. (i) the relationship between the textual and visual information (ii) the relationship between headline and the news content. It has been observed that fake news creator uses faulty images or catchy images that has no relationship with the news content so as to gain reader's attention. Hence such relationship measures must be considered. Similarly fakenews creators tends to use headline that are more catchy so as to capture the attention of the reader more easily. Hence these similarity measures are also considered as a feature.

Similarity between headline and news content

Suppose $\mathbf{h}$ represent the tf-idf vector (or vector of any other vectorisation technique) and $\mathbf{b}$ represent the corresponding body content vector, the similarity between them is expressed in terms of cosine similarity. Mathematically,

$$K(\mathbf{h}, \mathbf{b}) = \frac{\mathbf{h} \cdot \mathbf{b}}{\|\mathbf{h}\|\|\mathbf{b}\|}$$

Similarity between the content and image attached

Suppose $\mathbf{t}$ represent the text vector, rather the vector that is the output of Text CNN and $\mathbf{v}$ represent the vector that is the output of Image CNN, the similarity between them is expressed in terms of a modified cosine similarity. Mathematically expressed as,

$$K(\mathbf{t}, \mathbf{v}) = \frac{\mathbf{t} \cdot \mathbf{v} + \|\mathbf{t}\|\|\mathbf{v}\|}{2\|\mathbf{t}\|\|\mathbf{v}\|}$$

4.4.4 Model Integration

Integrating all features

Now that we have extracted all the features, next task is to combine or concatenate all these features together. Text features are comprised of both the explicit features as well as the output of Text CNN. Cosine similarity of headline and news body is also included. Image features are composed of the explicit features as well as the output of Image CNN. These text features and image features are concatanated together to form the final set of vectors. The measure of similarity between the output of Text CNN and output of Image CNN is also included in the final set of vectors. This concatanate feature vector is then used for training the dataset with the help of a Deep

Neural Network.

Loss function

When detecting as news as fake or not, we do it on the basis of the following aspects (1) textual and visual information, or (2) the relationship between them. Thus we represent our loss function as a comination of two loss functions as explained below

To denote the importance of both visual and textual features extracted in determining the news as fake or not the first loss function is expressed in terms of it.That is here we are mapping the textual and visual features into

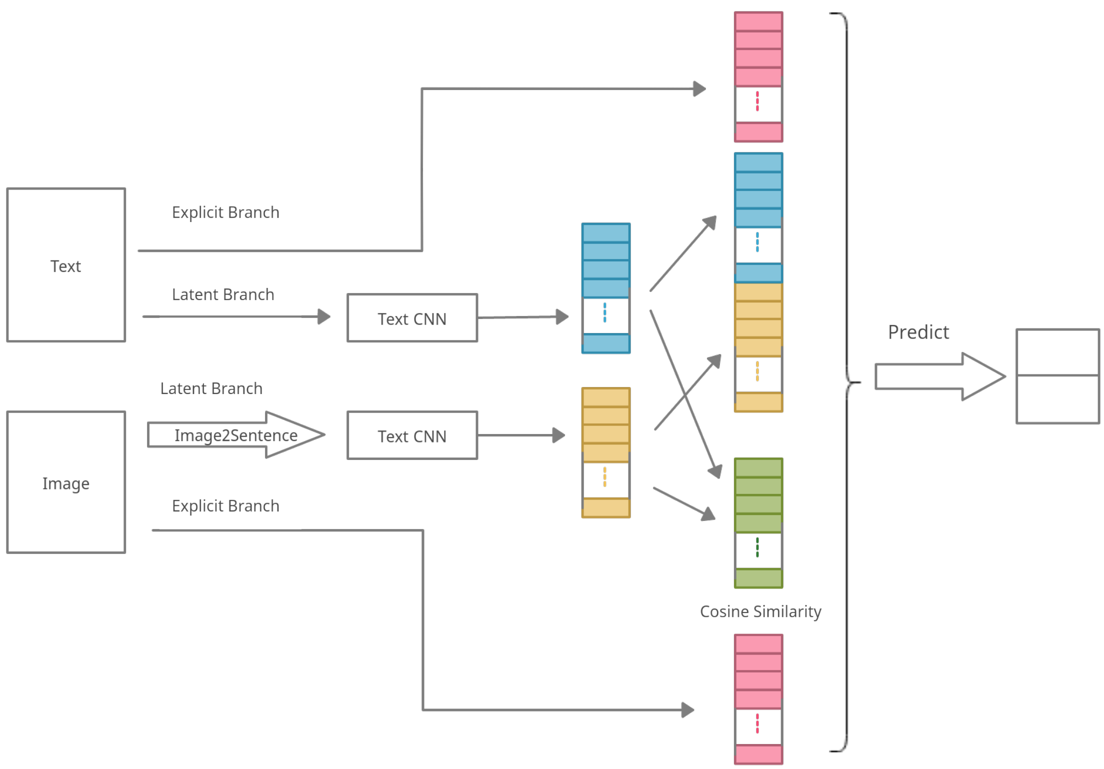


Figure 4.2: Architecture

its label. Mathematically, such possibilities can be computed by,

$$M_p(\mathbf{f}) \ = \ \mathbf{1} \cdot softmax(\mathbf{W}_p(\mathbf{f}) \ + \ \mathbf{b}_p)$$

where $\mathbf{f}$ is the final concatanated vector, $\mathbf{W}_p$ and $\mathbf{b}_p$ are parameters of the neural network to be learned. Here the loss function is a cross entropy based one that compute the possibility of how far a news article is from to be declared as fake or not.

$$L_p(\theta_p, \theta_t, \theta_v) \ = \ -\Sigma(y \ logM_p(\mathbf{f}) \ + \ (1-y) \ log(1 - M_p(\mathbf{f})))$$

where $\theta_p \ = \ \{\mathbf{W}_p, \mathbf{b}_p\}$, $\theta_t \ = \ \{\mathbf{W}_t, \mathbf{b}_t, W_t, b_t\}$ and $\theta_v \ = \ \{\mathbf{W}_v, \mathbf{b}_v, W_v, b_v\}$ However besides considering only the textual and visual features, we can consider the relation ship between the two modalities. Here we are using a modified cosine similarity measure to measure the similarity between the text and image features. The modified consine similarity is denoted as :

$$M_s(\mathbf{t}, \mathbf{v}) \ = \ \frac{\mathbf{t} \ \cdot \ \mathbf{v} \ + \ \|\mathbf{t}\|\|\mathbf{v}\|}{2\|\mathbf{t}\|\|\mathbf{v}\|}$$

By this modified similarity measure, it is guaranteed that $M_s(\mathbf{t}, \mathbf{v})$ is positive and ϵ [0,1]. 0 indicates that $\mathbf{t}$ and $\mathbf{v}$ are quite different, while 1 indicates that $\mathbf{t}$ and $\mathbf{v}$ are exactly the same.

Then, a similar cross entropy based loss function is defined such that the news articles with unrelated visual and textual features is more likely to be considered fake as compared to the ones with related textual and visual features. It is defined as:

$$L_s(\theta_t, \theta_v) \ = \ -\Sigma(y \ logM_s(\mathbf{t}, \mathbf{v}) \ + \ (1-y) \ log(1 - M_s(\mathbf{t}, \mathbf{v})))$$

To involve both cases, we specify our final loss function as

$$L(\theta_p, \theta_t, \theta_v) \ = \ \alpha L_p(\theta_p, \theta_t, \theta_v) \ + \ \beta L_s(\theta_t, \theta_v)$$

where α and β are values between 0 and 1. Thus while training our model we will be minimising this loss function.

Chapter 5

Results

5.1 Datasets

The dataset we used contains 13830 news, i.e., 5470 fake news and 8360 realnews. It is available online. The dataset which we used is actually a combination of two seperate datasets Politifact and Gossipcop. Both of these are popular fact-checking websites and are therefore very reliable. These datasets are provided by experts which guarentees the quality of news labels. The combined dataset contains all the information related to a news article such as its title, content, image, date, author, etc. But we only choose those columns that are necessary for us and they are the title, which is the title of the news, the content that displays the content of the news, and the url of the image which is the url of the image attached to the news title. The table below shows the number of news articles in each dataset.

	Fake	**True**	**Overall**
Politifact	270	210	480
Gossipcop	5200	8150	13350

5.2 Implementation details

In our design, each dataset was seperated in the ratio 80:20 such that 80% of the data was used for training, to learn the training parameters and the rest 20% for testing. We used five-fold cross validation for the model training. The learning rate is set as 10^{-4}. The other model paramters are provided below.

With the limited recourse available these parameters were found to give

Hyperparameter	**Value**
Batch size	200
Number of epochs	60
Number of filters used in CNN	128
Dropout keep probability	0.5

maximum accuracy.

5.3 Performation measure

Accuracy is the ratio of the number of correctly predicted data points to the number off all the data points.
Precision is the ratio of correctly predicted positive data points to the total number of correctly predicted positive points.
Recall is the ratio of correctly predicted positive datapoints to the total number of datapoints in the actual class
F1 Score is the weighted average of Precision and Recall.

		True value		
		Fake	True	Total
Predicted	Fake	28	8	36
	True	9	44	53
	Total	37	52	89

Table 5.1: Politifact: Confusion matrix

		True value		
		Fake	True	Total
Predicted	Fake	609	226	835
	True	237	1613	1850
	Total	846	1839	2685

Table 5.2: Gossipcop: Confusion matrix

5.4 Experiments and analysis

5.4.1 Baselines

To measure the success of our design, we compare our results with that of existing design for fake news detection. We compare to following baselines.

- **LIWC**[13]: LIWC is a psycho-linguistics lexicon that is widely accepted. It detects fake news considering only the textual features. Given a news story, LIWC counts the words that fall into various categories like linguistic, psychological and topical. This count is then used to predict fake news.

- **VGG-19**[14]: VGG-19 is a type of CNN with 19 layers that is mainly used for image classification. We use a fine-tuned VGG-19 as one of the baselines. In this design only the image features are considered.

- **att-RNN**[15]: att-RNN is a deep neural network that also consider multi-modality in detecting fake news. It uses LSTM for the text part and VGG-19 for the image part. This design is also a multi modal similar to ours.

Apart from these baselines we also compare our model FN-CNN (Fake News CNN) with slight variant of our model.

- **FN-CNN:NS** In this variant we don't consider the similarity between the news textual and visual information. In this variant, the extracted textual and visual features of each news article are concatenated. And only this concatenated vector is used for prediction.

- **FN-CNN:OS** In this variant we consider only the similarity between the news textual and visual information. And only this similarity measure is used in prediction.

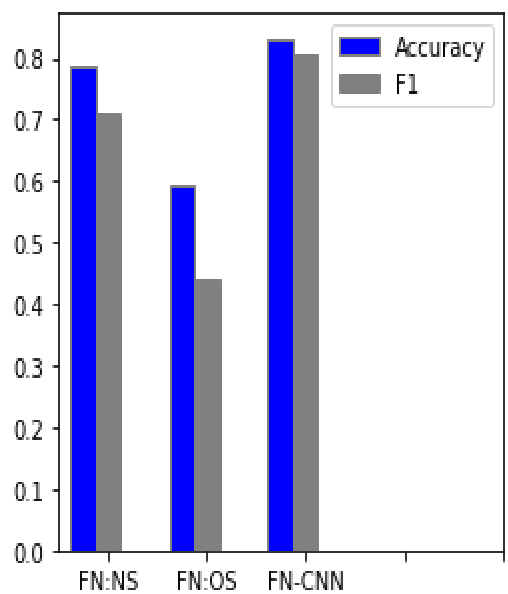
5.4.2 Performance analysis

We then compare our design with the baselines that are mentioned above. The table below shows the summary of comparison of our design with the existing ones.

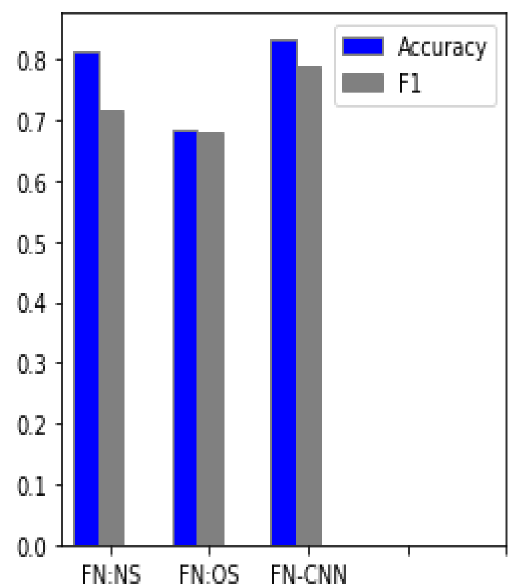
		LIWC	**VGG-19**	**att-RNN**	**FN-CNN:NS**	**FN-CNN:OS**	**FN-CNN**
Politifact	**Acc.**	0.772	0.649	0.769	0.796	0.738	0.834
	Pre.	0.775	0.668	0.735	0.726	0.752	0.789
	Rec.	0.814	0.787	0.842	0.701	0.644	0.803
	F1	0.792	0.720	0.782	0.713	0.695	0.806
Gossipcop	**Acc.**	0.836	0.775	0.743	0.814	0.812	0.838
	Pre.	0.7 78	0.775	0.788	0.775	0.753	0.797
	Rec.	0.317	0.970	0.823	0.772	0.701	0.817
	F1	0.466	0.862	0.796	0.774	0.726	0.805

5.4.3 Module analysis

We have also considered variants of our own model to show how important it is to consider multiple modalities when designing a model. The variants of the model we have chosen are (1) OS, in this only the similarity measure is considered, (2) NS, in this no similarity measure is considered and only the concatenated image and text vector is considered. The following graph shows how accuracy is varying in each variants. Accuracy is maximum when both the modalities and its similarities are taken into account. Here B refers



(a) Politifact



(b) Gossipcop

Figure 5.1: Model variant analayasis

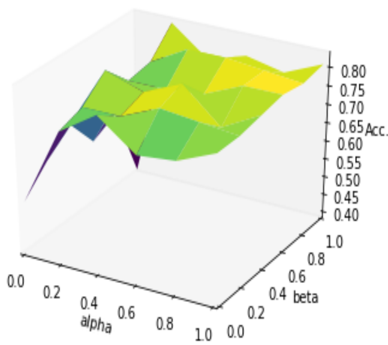
to Both that is both the modalities as well the similarity measure between them. And the accuracy is maximum for this.

5.4.4 Parameter analysis

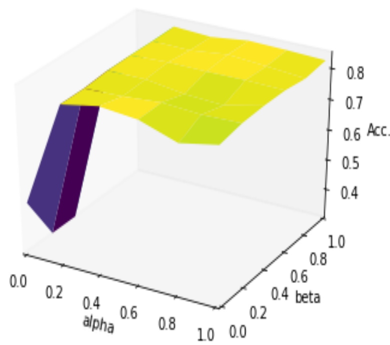
When looking at the loss function equation we can see two parameters α and β which are used to denote the relative importance between the concatenated textual and visual features (α) and the similarity between the textual and visual features (β). To understand how these parameters affect the model performance we calculate the accuracy by iterating the value of α and β over the range from zero to one with a gap of 0.2. The results are then plotted in the graph as shown below in Fig 5.2. From the figure it is clear that various combination of parameters leads to the accuracy of model ranging from 0.75 to 0.84. From the graph it is clear that, when $\alpha : \beta = 0 : 6 : 0 : 4$ in PolitiFact and $\alpha : \beta = 0 : 4 : 0 : 6$ in GossipCop, the accuracy and F1 score is high. This shows the importance of considering various modalities and also the the importance of considering the relationship between the two modalities.

5.4.5 Case study

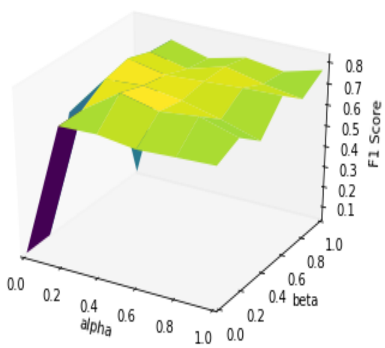
In our case study we want to ensure if our model can identify the real-world fake news depending on the dissimilarity between the image and visual features. We wanted to check if FN-CNN can identify the dissimilarity and thereby recognise the news as fake. For this purpose we went through some real world news stories and compared their ground truth news label to the cosine similarity that we get from our model. Some examples are shown below. It can be observed that gap between image and text features exist and this dissimilarity is captured by our model. For example in Fig. 5.3(a) the image is declared fake because the image is not related to voting or bills.. Also not in Fig. 5.3(b) images are fake news are mostly collages or random group photos.Using attractive images to gain the attraction of the images is clearly noticeable here Such images don't have any relationship with the news content. For example, the fake news in Fig. 5.3(c) includes an image



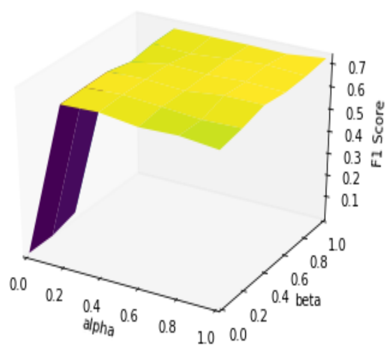
(a) Politifact



(b) Gossipcop



(c) Politifact



(d) Gossipcop

Figure 5.2: Accuracy/F1 vs parameter analaysis

with a smiling individual that has no relationship with that of the death news. Thus our model is capable of identifying such dissimilarity. Also note the cosine similarity values, for true news the value is less signifying that image and visual features are someway related. But in the case of fake news it is noticed to be large.

(a) c = 0.0236

(b) c = 0.0021

(c) c = 0.0441

Figure 5.3: Similarity measure for Fake news

(a) c = 0.8399

(b) c = 0.99421

(c) c = 0.7210

Figure 5.4: Similarity measure for True news

Chapter 6

Conclusion and Future Direction of Research

The influence of social media is increasing rapidly now so that people nowadays prefer news from the social media more than the traditional news sources. Hence fake news creators turned their attention towards this online sources. Therefore, the identification of fake news grows in importance. In this paper, we have explored the fake news problem by reviewing the existing literature. The model that we proposed can capture such fake news articles with decent accuracy. We have also shown the importance of considering visual features and also the importance of considering relationship between the various modalites. The model we designed also works decently well in real time news articles.

The proposed model can be improved by exploring and adding more explicit features that support classification. There are many social media platforms like Twitter, quora, facebook, etc, which have an extensive collection of datasets with images and data. Testing this model on such datasets can also be explored in the future.

ACKNOWLEDGEMENT

We would like to express our sincere thanks to our mentor **Dr. Jay Prakash** who gave us this opportunity to do this project on the topic-"Detection Of Fake News using Deep Learning" and also for his able guidance and support. Also, we would like to express our deepest appreciation to all those who helped us to complete this report. We thank FRG, National Institute of Technology, Calicut for all the support.

References

[1] S. Kai, S. Amy, W. Suhang, T. Jiliang, and L. Huan, "Fake news detection on social media: A data mining perspective,"

[2] L. Yang and B. Yi-Fang, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

[3] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, "Fake news detection: A deep learning approach," *SMU Data Science Review*, vol. 1, 2018.

[4] Y. Yang, Z. Lei, Z. Jiawei, C. Qingcai, Z. Xiaoming, L. Zhoujun, and Y. Philip S, "Ti-cnn: Convolutional neural networks for fake news detection," 2018.

[5] S. R. Sangwan and M. Bhatia, "Denigration bullying resolution using wolf search optimized online reputation rumour detection," *Procedia Computer Science*, vol. 173, pp. 305 – 314, 2020. International Conference on Smart Sustainable Intelligent Computing and Applications under ICITETM2020.

[6] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy," p. 729–736, 2013.

[7] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 598–608, 2017.

[8] P. Qi, J. Cao, T. Yang, J. Guo, and J. Li, "Exploiting multi-domain visual information for fake news detection," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 518–527, 2019.

[9] J. Pennebaker and L. King, "Linguistic styles: Language use as an individual difference," *Journal of personality and social psychology*, vol. 77, pp. 1296–312, 01 2000.

[10] M. Newman, J. Pennebaker, D. Berry, and J. Richards, "Lying words: Predicting deception from linguistic styles," *Personality social psychology bulletin*, vol. 29, pp. 665–75, 06 2003.

[11] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014.

[12] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164, 2015.

[13] J. Pennebaker, R. Boyd, K. Jordan, and K. Blackburn, *The development and psychometric properties of LIWC2015*. University of Texas at Austin, 2015.

[14] A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[15] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, (New York, NY, USA), p. 795–816, Association for Computing Machinery, 2017.

[16] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *CoRR*, vol. abs/1103.0398, 2011.

[17] A. Jivani, "A comparative study of stemming algorithms ms .," 2011.

[18] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[19] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.