OPINION SPAM DETECTION USING ENSEMBLE MODELS

A PROJECT REPORT

Submitted by

SIMRAN KATHURIA [Reg.No:RA1811003030010]

Under the guidance of

Mr. NISHANT ANAND

(Assistant Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM

INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

DELHI-NCR CAMPUS, GHAZIABAD (U.P)

SRM INSTITUTE OF SCIENCE & TECHNOLOGY, NCR CAMPUS

MAY 2022

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**OPINION SPAM DETECTION USING ENSEMBLE MODELS**" is the bonafide work of " **SIMRAN KATHURIA [Reg.No:RA1811003030010]**", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Mr. NISHANT ANAND **GUIDE** Assistant Professor Dept. of Computer Science & Engineering	Dr. R. P. MAHAPATRA **HEAD OF THE DEPARTMENT** Dept. of Computer Science Engineering
Signature of the Internal Examiner	Signature of the External Examiner

ABSTRACT

People nowadays examine review websites, internet forums, and sites. Online surveys are frequently the essential worry in a customer's choice to buy an item or service and are a significant wellspring of data that can be utilized to decide popular assessment on these items or services. Mischievous individuals may use these reviews to promote or denigrate certain items or services. As a result, the reliability of reviews is questioned. Opinion (Review) Spam is a process in which spammers manipulate and poison reviews (by creating fraudulent, untruthful, or misleading surveys) for personal advantage. Since not all internet-based surveys are honest and dependable, it is critical to foster methods for distinguishing audit spam. To achieve a solution to this problem we will be using English text corpus to train the deep learning model and by using BERT transformer to extract meaningful features from given text i.e a word with multiple meanings can be determined using position embedding, that makes the most sense in the given context. and we will be expecting the output in binary format using a binary classifier. For the same problem statement, many different solutions have been proposed but we will ensemble different models so that we could achieve the highest accuracy.

ACKNOWLEDGEMENTS

I would like to convey my heartfelt thanks to my guide, Mr. Nishant Anand for his invaluable advice, constant encouragement, prompt assistance and provision of an ideal research environment. Throughout the project, despite his hectic schedule, he has been a cordial supporter of my efforts to complete this re-search task.

Dhawal Sarin [RA1811003030002]

Ekansh Sharma [RA1811003030008]

Simran Kathuria [RA1811003030010]

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

ELM	Ensemble Learning Models
NB	Naive Bayes
LR	Logistic Regression
CNN	Convolutional Neural Network
SVM	Support Vector Machine
KNN	K-Nearest Neighbours
NSP	Next Sentence Prediction
MLM	Masked-Language Modeling
TPR	True Positive Rate
TNR	True Negative Rate

CHAPTER 1

INTRODUCTION

1.1 What we are doing!

1. In this work, a new technique for detecting spam or fraudulent reviews is proposed. Firstly, we used a real-life data set from Kaggle (Kaggle), rather than relying solely on the semireal dataset, because the semi-real dataset is unreliable. Our corpus comprises of 20 hotel reviews in Chicago that are both honest and fraudulent. To our knowledge, only a few researchers, such as Mukherjee et al. (2013), have used a semireal dataset, in which spam identification is significantly easy.

2. Then ,we have employed Ensemble Learning Models (ELM) rather than using single base classifiers because ELM is more effective at detecting review spam than single baseclassifiers. Bert Encoder, Convolutional Neural Network (CNN), Max pooling, Relu function, Sigmoid function, and Normalising are all included in the proposed ELM.

3. Also, we have compared the performance of our proposed system to that of variousexisting supervised machine learning classifiers such as Naive Bayes (NB), Linear Classifier: Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbours (KNN)and so on.

1.2 Types of Spam Review

The spam review can be categorized in 3 different types:

1. **Deceptive(fake) reviews:** Deceptive (fake) reviews are reviews produced without any prior experience; their sole purpose is to promote or criticise a product/service.
2. **Brand-specific reviews:** which target a company or a brand rather than specific product or service for which review is intended.
3. **Non-reviews:** Inappropriate conversations, ads, and linkages to other product/service pages, the first type is more difficult to identify than the other two, although the other two can be easily discovered using various techniques or simply by reading.

1.3 Model Input/Output

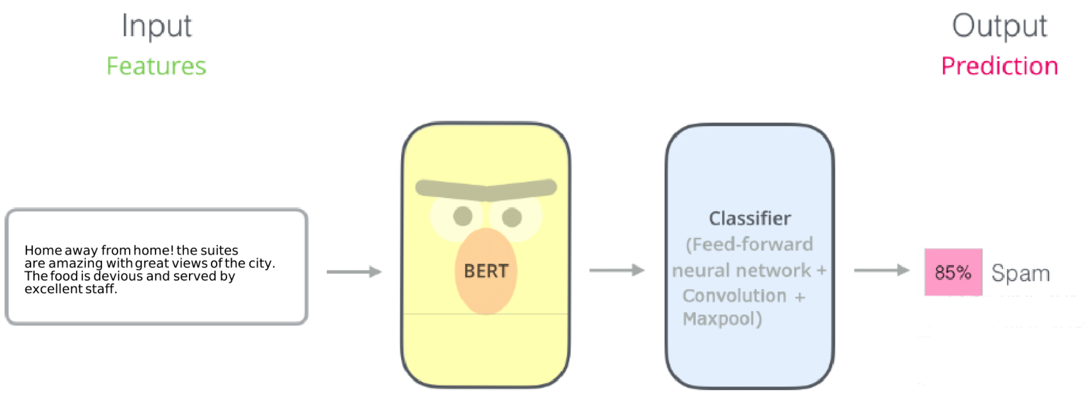

Input
Features
Output
Prediction
Home away from home! the suites are amazing with great views of the city. The food is devious and served by excellent staff.
BERT
Classifier (Feed-forward neural network + Convolution + Maxpool)
85%
Spam

Figure 1.1: Illustration showing example of Input / Output for the Neural Networks.

Simple working of model

Collection of text (review) is passed to the BERT model, which give classifiers (layers), input as in vector matrix form, and at the end it will show where the review is deceptive or not. In-depth working is in chapter 4.

The rest of the paper is organized as follows:

In Chapter 2, a quick review of the literature is presented. Chapter 4 covers methodology, a thorough description of the proposed ELM, Bert encoder, experimental setup, and data pre-processing. The results and their discussion are depicted in Chapter 5 and Chapter 6 and 7 deal with the conclusion and recommendations for future studies.

CHAPTER 2

LITERATURE SURVEY

1. **Enactment of Ensemble Learning for Review Spam Detection on Selected Features (Khurshid et al. (2018)).**
   On actual and semi-real-life datasets, they studied the performance and efficacy of ensemble learning with feature selection strategies for review spam identification. For the Yelp, M.OttPos-Pol, and M.Ott Neg-Pol datasets, they used the ChiSquared feature selection approach with ELM and obtained accuracy of 85.1%, 82.0%, and 77.4%, respectively.

2. **Attention-based Bidirectional L.S.T.M. for Deceptive Opinion Spam Classification(Salunkhe (2021)).**
   They employed many machine learning models in this work that may be employed to train the final model. Different classifiers, such as the NB Classifier, Linear Classifier: Logistic Regression, Support Vector Machine, and Deep Neural Networks, are applied, and their findings are compared to performance metrics to determine the final model for classification.

3. **Towards Accurate DeceptiveOpinions Detection based on Word Orderpreserving CNN(Zhao et al. (2018))**
   The CNN in the deep learning model is employed in this work to identify fraudulent opinions. And achieving an 84% accuracy rate.

4. **Opinion Spam and Analysis(Jindal and Liu (2008))**
   In this paper, they are using logistic regression and getting accuracy of 78%.

5. **Improving OpinionSpam Detection by Cumulative Relative Frequency Distribution (2020)(Fazzolari et al. (2020))**
   They employed numerous supervised machine-learning algorithms in this study, including Logistic Regression (L.R.), Support Vector Machine (S.V.M.), Decision Tree (D.T.), Naive Bayes (N.B.), and K-Nearest Neighbours(K.N.N.), as well as a Stratified k-Fold Cross Validation technique to assure greater results dependability.

6. **Review Spam Detection (Jindal and Liu (2007))**
   In this paper, they have used the statistical package R to perform logistic regression that gives the average AUC value of 78% by applying 10fold cross validation on the data.

7. **Learning to Identify ReviewSpam(Li et al. (2011))**
   In this paper, They use supervised learning approaches initially, then 2-view semi-supervised methods to leverage the vast quantity of unlabeled data in this research. The findings of the experiment reveal that 2-view co-training algorithms obtain better outcomes than singleview algorithms, with an accuracy of 63.1%.

8. **What Yelp Fake Review Filter Might Be Doing? (Mukherjee et al. (2013))**
   This research looked at the in-depth nature of phoney reviews in a business scenario. Although linguistic approaches showed high accuracy i.e. 90 percent, on crowd-sourced deceptive reviews, this study reveals that they do not function well on actual-life deceptive reviews. The accuracy of behavioural characteristics was 86%.

9. **SpamReview Detection Techniques: A Systematic Literature Review(Hussain et al. (2019))**
   This literature review presents an over all discussion about different feature extraction approaches from review datasets. Research gaps and future directions in the area of spam review detection are also presented.

10. **Spam Detection in Online Social Networks Using Feed-Forward Neural Network (Kaur and Sabharwal (2018))**
    In this paper they compare various algorithms with FFNN + ICA such as Random Tree, Random Forest, Naive Bayes. Their proposed model reaches an accuracy of 90%.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Supervised learning problem

Ground truth values are known, making this a supervised learning problem. This is a binary-label classification problem as opposed to a multi-class classification problem each output label is independent and acts as a binary classification problem.

3.2 Selection of appropriate loss function

Binary cross-entropy is the chosen loss function taken element wise over the entire output. Loss formula is given below.

$$loss = -(ylog(p) + (1-y)log(1-p)) \tag{3.1}$$

* where y is the ground truth value, and p is the predicted value

3.3 Model specific details

Our model have 14 layers with sigmoid activation to make sure the output value for each label stays inbetween one and zero.

$$sigmoid(x) = \frac{1}{1-e^{\text{-x}}} \tag{3.2}$$

Internal layers use ReLU as the activation function to make the training process faster.

$$relu(x) = \begin{cases} 0, & \text{if } value \leq 0 \\ x, & \text{if } value > 0 \end{cases} \tag{3.3}$$

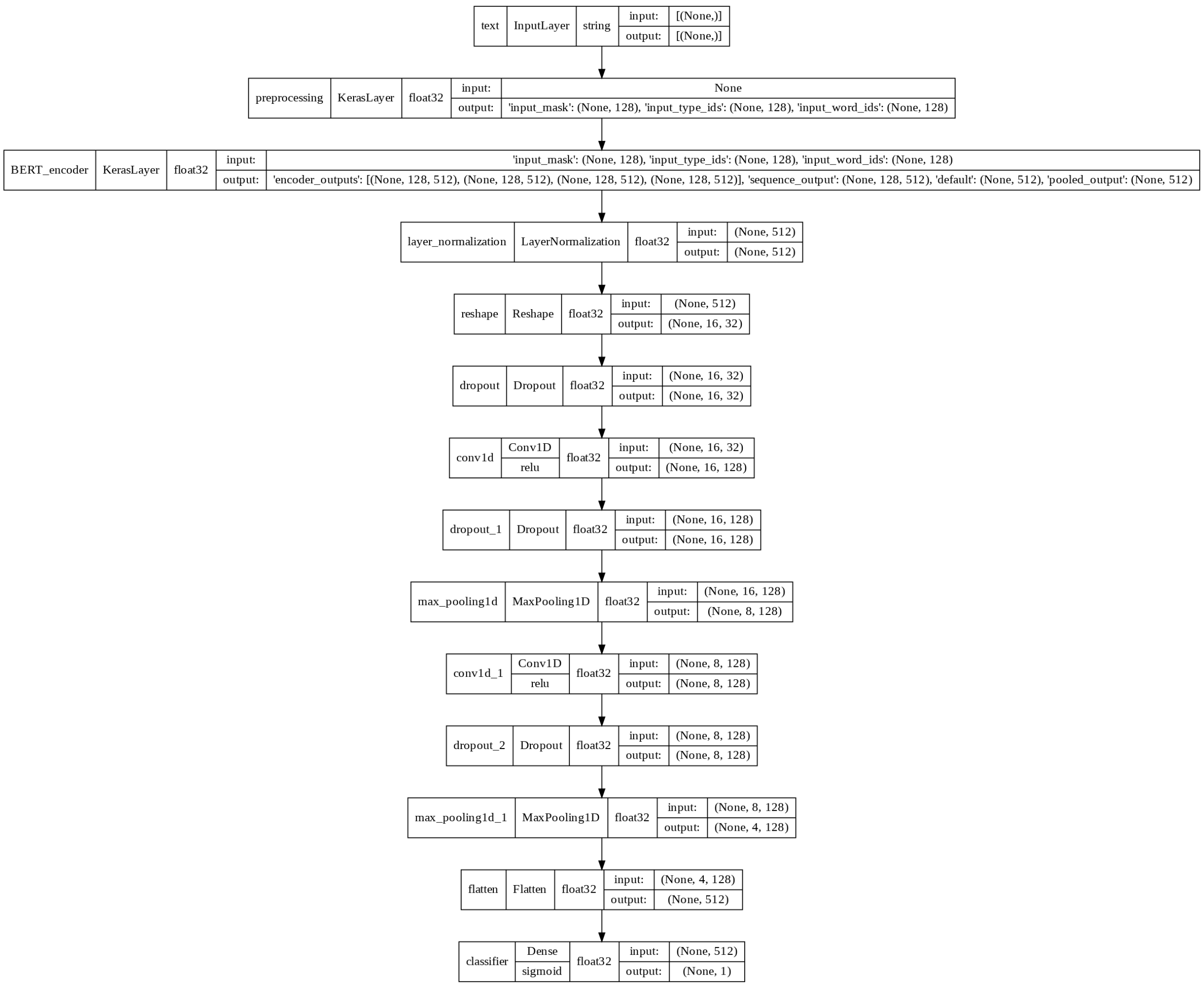
Selected optimizer is AdamW. To avoid the problem of exploding and vanishing gradients from the network being too deep, GlorotUniform is the kernel and bias initializer. Dropout is used to prevent overfitting. Keras is the chosen framework with Tensorflow 2.0 as backend.

CHAPTER 4

SYSTEM DESIGN

4.1 Architecture

A Neural Network takes the text sequence (review) as input and makes a prediction.



text | InputLayer | string | input: [(None,)] | output: [(None,)]
preprocessing | KerasLayer | float32 | input: None | output: 'input_mask': (None, 128), 'input_type_ids': (None, 128), 'input_word_ids': (None, 128)
BERT_encoder | KerasLayer | float32 | input: 'input_mask': (None, 128), 'input_type_ids': (None, 128), 'input_word_ids': (None, 128) | output: 'encoder_outputs': [(None, 128, 512), (None, 128, 512), (None, 128, 512), (None, 128, 512)], 'sequence_output': (None, 128, 512), 'default': (None, 512), 'pooled_output': (None, 512)
layer_normalization | LayerNormalization | float32 | input: (None, 512) | output: (None, 512)
reshape | Reshape | float32 | input: (None, 512) | output: (None, 16, 32)
dropout | Dropout | float32 | input: (None, 16, 32) | output: (None, 16, 32)
conv1d | Conv1D relu | float32 | input: (None, 16, 32) | output: (None, 16, 128)
dropout_1 | Dropout | float32 | input: (None, 16, 128) | output: (None, 16, 128)
max_pooling1d | MaxPooling1D | float32 | input: (None, 16, 128) | output: (None, 8, 128)
conv1d_1 | Conv1D relu | float32 | input: (None, 8, 128) | output: (None, 8, 128)
dropout_2 | Dropout | float32 | input: (None, 8, 128) | output: (None, 8, 128)
max_pooling1d_1 | MaxPooling1D | float32 | input: (None, 8, 128) | output: (None, 4, 128)
flatten | Flatten | float32 | input: (None, 4, 128) | output: (None, 512)
classifier | Dense sigmoid | float32 | input: (None, 512) | output: (None, 1)

Figure 4.1: Detailed Architecture.

4.2 RNN Model

This neural network model **Fig. 4.1** is divided into 14 major layers. First three layers includes text pre-processing and tokenizing layer (BERT). The fourth layer is Standard normalization layer followed by a reshaping and dropout layer. The next seven layers consists of multiple convolution , max-pooling , dropout layers whereas the last layer is the classifier(sigmoid).

Layer 1 takes sequence of text (review) of size n ([None]) as input.

Layer 2 takes the output from the previous layer of dimension n ([None]) as an input, the purpose of this layer is to standardise the input by usually lowercasing + punctuation stripping, eliminating white spaces, eliminating numeric characters. Layer then gives the output of dimension (None x 512) i.e n sentences of 512 words each.

Layer 3 take the output from previous layer of dimension Nonex512 as input, the working of this layer is explained in detail in section 4.3 this layer gives the output of dimension Nonex512. (This dimension depends upon the type of BERT transformer you are using.)

Layer 4 is the normalization layer this layer take the output from the preceding layer of dimension Nonex512 asinput,This layer is used to make preparing a model more straight-forward and quicker. Standardization helps with keeping input information inside a scope of values that work well with the default trigger capacities, introduction loads, and other foundation settings. This layer gives the output of dimension Nonex512.

Layer 5 is the reshape layer this layer take the output from previous layer of dimension Nonex512 as input, this reshape layer changes the input shape to Nonex16x32 just to reduce time and increase performance.

Layer 6,8,11 are the dropout layer these layers take output from the previous layer as input and give output in the same dimension, function of these layers to prevent over fitting of the model.

Layer 7 take the output from the previous layer of dimension Nonex16x32 as input, it has one convolution layer with 128 filters of dimension (kernel_size) 9x9 and gives the output of dimension Nonex6x128.

Layer 9 is the max pooling layer that takes the output from the previous layer of dimension Nonex16x128 as input it then reduces the size of the input in half by using a 2x2 pool size with stride of 2x2 and give output of dimension Nonex8x128.

Layer 10 take the output from previous layer of dimension Nonex8x32 as input, it has one convolution layer with 128 filters of dimension (kernel_size) 7x7 and gives the output of dimension Nonex8x128.

Layer 12 is the max pooling layer that take output from previous layer of dimension Nonex8x128 as input it then reduces the size of the input in half by using a 2x2 pool size with stride of 2x2 and give output of dimension Nonex4x128.

Layer 13 flattens the output from the previous layer to get an output of Nonex512.

Layer 14 The final / output layer.The activation function in the final layer is sigmoid. The output shape is [1].

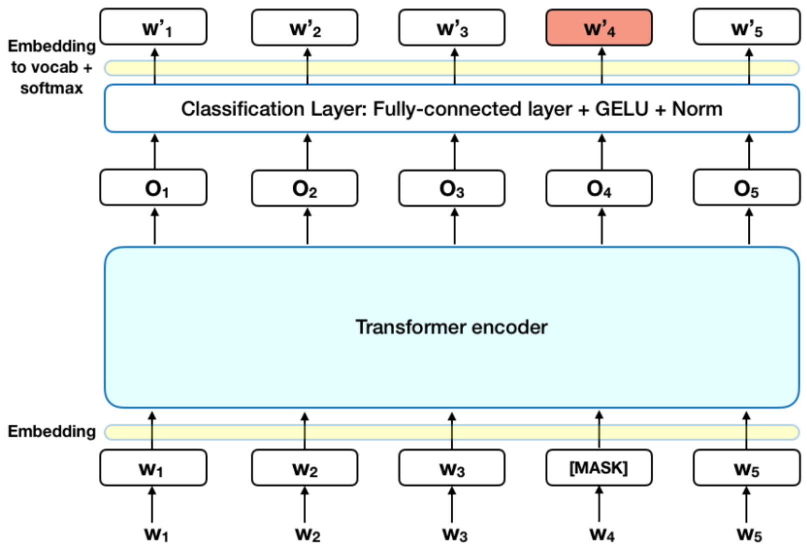
4.3 How BERT works

The Transformer goes through the entire sequence of words without pausing (right-to-left or left-to-right). It is thought to be bidirectional along these lines, however it is more accurate to state it is non-directional. Characteristic enables the model to become acquainted with a word's situation in light of all of its surrounding features (left-to-right of the word). The level portrayal of the Transformer encoder in the graph below is unmistakable. The data is made up of a series of tokens that are first entered into vectors and then processed by the brain. The end output is a collection of H-dimensional vectors, each of which corresponds to a data tokens with a similar file. There is a test for describing an expectation target when developing language models. Next is predicted by many models in a sequence(for example, "The youngster arrived home from _________"), which is a directed technique that limits setting learning. BERT employs two methods of preparing to pass this test:

Masked-Language Modeling (MLM)
Fig. 4.2 Masked Language Modeling (MLM) Before feeding word sequences into BERT, 15% of the words in each sequence were removed.A MASKed token is substituted for each succession. The model, in turn, Attempt to predict the initial worth of the covered words at that time.The other, non-veiled, words in the arrangement provide light to the environment.The anticipation of the outcome words, in specialist terms, necessitates:

1. Applying a grouping layer to the encoder output.
2. Using the installation grid, duplicate the result vectors and change them to the jargon aspect.

3. Using softmax, calculate the chance of each word in the jargon.



Embedding to vocab + softmax
W'1
W'2
W'3
W'4
W'5
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
W1
W2
W3
[MASK]
W5
W1
W2
W3
W4
W5

Figure 4.2: Masking in BERT.

The BERT loss function solely considers the forecast of veiled attributes and ignores the nonmasked terms' anticipation.Resulting in, converging of model more slowly than directed models, which is offset by its higher context awareness.

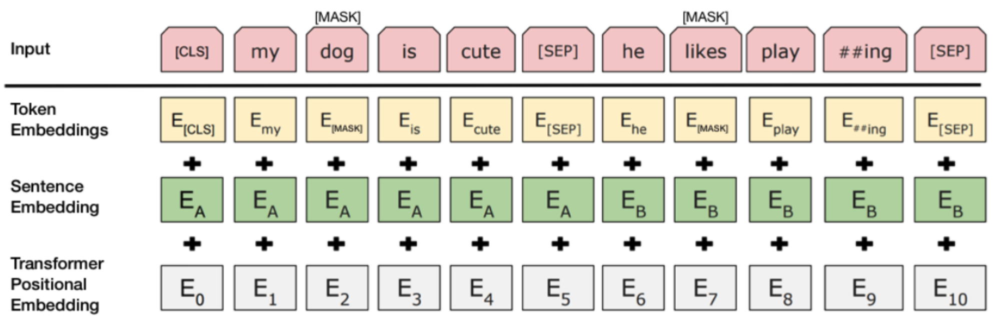
Next Sentence Prediction (NSP)

Fig. 4.3The model learn foretell, yet if the 2nd sentence in a pair is the sub-sequent sentence in the actual document file through the BERT training procedure, which takes pairs of sentences as input. During training, half of the inputs are sets of sentences, with the second statement being the following statement in the original text, while the other half are arbitrary sentences from the corpus. It is believed that from the first sentence, the random sentence will be separated. In order to aid model with recognizing the 2 sentences in preparing, the information is handled in an accompanying way prior to entering to model:

1. A [CLS] marker placed at every beginning of each statement, and [SEP] marker placed at every end of each statement.

2. Every token has a statement insertion indicating Statement A or Statement B. In concept, sentence embedding is similar to token embedding with a vocabulary of 2.

3. Each marker (token) is given a positional inserting to indicate where it belongs in the arrangement. The Transformer paper introduces the concept and implementation of positional installation.



Input
[CLS]
my
[MASK]
dog
is
cute
[SEP]
he
[MASK]
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} $E_{[MASK]}$ E_{is} E_{cute} $E_{[SEP]}$ E_{he} $E_{[MASK]}$ E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Sentence Embedding
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Transformer Positional Embedding
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 4.3: Tokenization in BERT

For predicting whether, the 2nd sentence is connected to the 1st sentence, these steps are executed:

1. Transformer model receives the whole input sequence.

2. Result of [CLS] token is changed into 2×1 formed vector, utilizing a basic arrangement layer.

3. Employing softmax to calculate the likelihood of IsFurtherSequence. (MLM) and (NSP) are trained simultaneously in the BERT model so as to reduce cumulative loss function of two procedures just to minimum.

4.4 Normalization

Normalization permits you to resize imported information prior to preparing the model. While highlight coding is empowered, you can pick the standardization procedure to utilize.

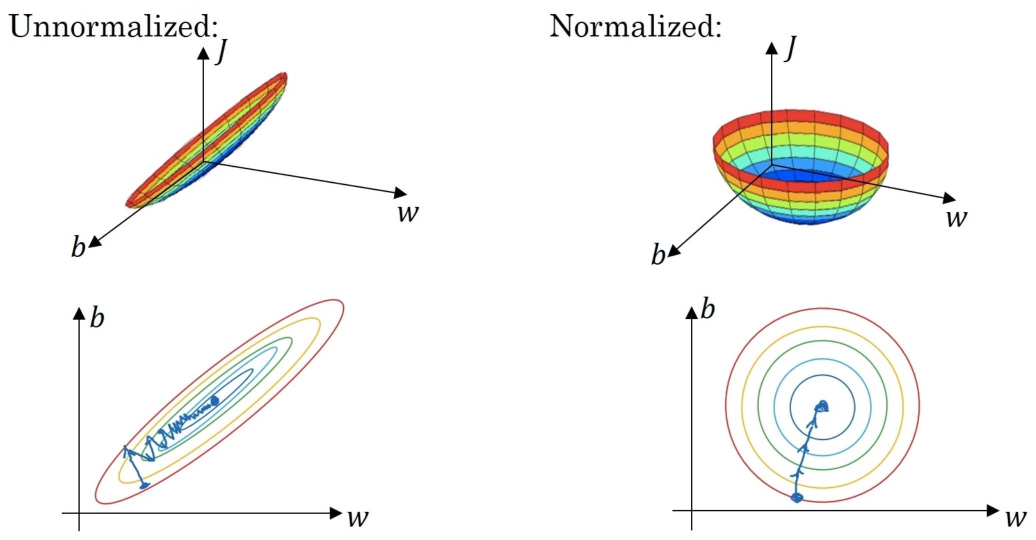


Figure 4.4: Difference b/w normalized and un-normalized data.

Standardization

Standardization (Normalization) is the method involved with changing a bunch of crude information over to zero mean & unit standard deviation. Values over the component mean will be scored emphatically, while values underneath the mean will be scored adversely. Standardization is performed with the understanding that the information is consistently disseminated. The information will be Gaussian after standardization, with a mean of 0 and a standard deviation of one. There can be values that are very far off from 0 (for instance, 5, 10, 20), however assuming the dissemination is unary Gaussian, these qualities have exceptionally low likelihood. The benchmark score, often known as the z score.

Why we used standardization ?

You might standardize the dataset to make preparing model more straight forward and quicker. Standardization helps with keeping input information inside a scope of values that work well with the default trigger capacities, introduction loads, and other foundation settings. Standardization appropriates the information esteems consistently, causing it doubtful that one information trademark will overwhelm the other.

Formula of standardization

The conventional score of a crude information esteem x is determined:

$$Z = \frac{x - \mu}{\sigma} \tag{4.1}$$

4.5 Convolution 1D

The Convolution 1D block is a layer for detecting features in a vector.

The 1D block is made up of a configurable number of filters, each of which has a fixed size; the vector and the filter are convolutioned, yielding a new vector with the same number of channels as the number of filters as an output. To induce non-linearity, each value in the tensor is passed through an activation function.

How does convolution 1D work?

When executing convolutions, the default is to shift filters of a certain Width by one element at a time, this is known as Horizontal stride, and

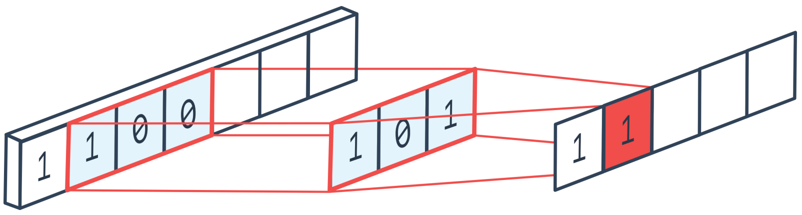


Figure 4.5: A 1D convolution with a kernel size of 3 and stride of 1.

it may be changed by the user. The output vector will be smaller as the stride increases. This can be used to decrease the amount of parameters and the amount of memory needed, but it results in data loss.

4.6 One-Dimension Maxpool

One-Dimension Max pooling block reduces the size of data without comromising with the quality , the number of parameters, and the amount of computation required, as well as controlling modeling error (overfitting).

The mechanism behind maximum pooling?

One-Dimension Max pooling block sweeps a pool (window) of a a given size over the incoming data with a predefined stride, determining the maximum in each window.

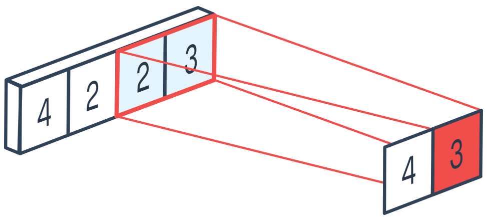


Figure 4.6: A One-Dimension Max pooling with a pool size=2 and stride=2.

When to employ One-Dimension Max pooling

After employing one or more convolution layers, max pooling layers are used to allow inner convolution layers to gather input from a larger percentage of the original data vector. When convolutional layers are used as detectors for a single feature, max pooling only keeps the "strongest" result from the pooling rectangle for that feature. Each channel (and hence each feature) receives its own treatment.

CHAPTER 5

EXPERIMENT RESULTS

5.1 Coding

Model compilation code:

1. Below are the output for RNN Model discussed in System Design's RNN Model section 4.2.
2. It uses the tensorflow library to define and compile the model.
3. Following which the model is saved into.tf file named"Bert_Model.tf"

5.2 Coding Workflow

I) Importing Modules

- **numpy** is imported for setting random seed for pandas & sklearn so as to make it easier to reproduce results
- **pandas** is imported for reading/loading are dataset which is in a tabular form (comma separated value file format)
- **matplotlib** is imported for creating graphs to visualize the training of our model. It's easier to see if it overfits or underfits
- We are using **sklearn** for dividing our dataset into 3 parts, ie; training, validation and test sets

  - main tensorflow module for making, compiling and training the model.
  - **tensorflow-hub** for Bert preprocessor and text encoder
  - **tensorflow-text** as a workaround for a OP call error in **tensorflowhub** possibly due to scoping issue or not being listed as a dependency
  - **tf-models-official** or official (as imported here) is used to get the AdamW optimizer for stochastic gradient descent based training of out model

II) Dataset Handling

- We are dropping **hotel** and review **source** attributes from our dataset so that our model learns to classify purely based on text and doesn't become brittle wherein it breaks if it encounters a new hotel name or source for the review fed to it.
- By that we mean it doesn't end up overfitting on our dataset and be completely impractical for use.
- **polarity** is removed, it's being data that isn't readily available in the text but rather an add-on value that has been provided in the dataset. We could make another classifier for polarity and then graft it's output with original text to use in our model but that's an unnecessary complication.
- We are making the text lowercase since the classification is going to be case insensitive for ease, since people may capitalise things for various reasons other than grammatical as well as do it by mistake as is common on most user content on the internet.
- We are also removing URLs from the reviews (if found) since they aren't relevant and could confuse the model as a feature for identifying as it as spam

III) Train, Test, Validation splits

- Test set size: 160
- Validation set size: 144
- Train set size: 1296

IV) Bert Encoder

We are using **small_Bert** with 4 hidden layers of 512 pooled output with 8 Attention heads

- L = number of hidden layers, valid values being one of [2, 4, 6, 8, 10, 12]
- H = pooled output size, valid values being one of [128, 512, 768]
- A = number of Attention head, valid values being one of [2, 4, 8, 12]

V) Model building

our classifier model composed of Bert Encoder (includes it's own preprocessing as well)
and 1 Dimensional Convolution layers **(tf.keras.layers.Conv1D)**.

- We normalize the pooled_output from Bert before feeding it into the classifier
- It is then reshaped from (1, 512) to (16, 32) for convolution layers
- kernels(layer parameters) are initialised with tf.keras. initializers. GlorotUniform for use with relu (rectified ELU) activation
- tf.keras.layers.Dropout for regularization.
- tf.keras.layers.MaxPool for using only the most pronounced features of encoded text
- tf.keras.layers.Flatten is used to flattening the output from prior layers for the final perceptron

- tf.keras.layers.Dense is the decision perceptron (Also referred to as FullyConnected) with sigmoid activation to get an output between (0, 1)

VI) Model Training

In the training step for our model history variable stores the per epoch results for training and validation dataset accuracy and loss step = number of batches the the data is fed to the model for loss evaluation and parameter optimization epoch = 1 complete run of all available steps, whole training data has been fed at the end of an epoch.

VII) Training Graph

Then the Graph plots the progress of the model's training based accuracy & loss for both training and validation datasets per epoch that are shown below.

5.3 Testing

5.3.1 Test Results

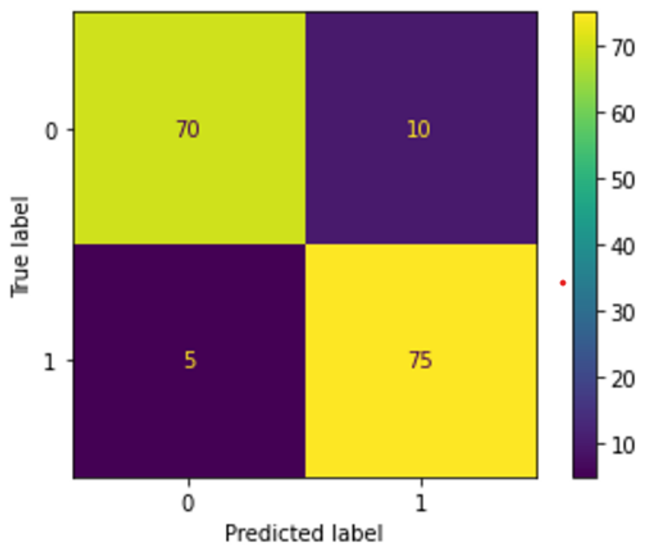


Figure 5.1: Confusion Matrix

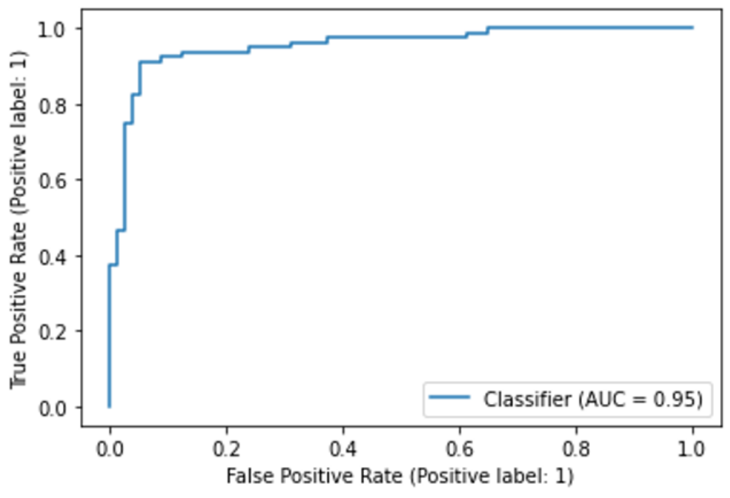


Figure 5.2: ROC Curve

5.3.2 Validation Results

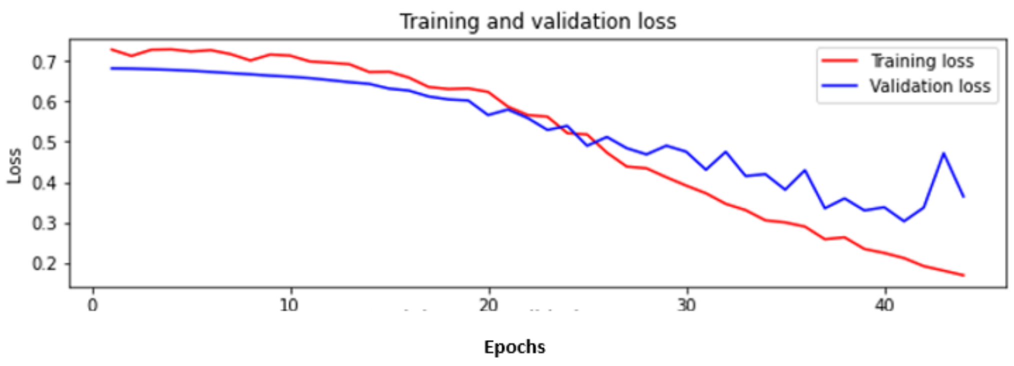


Figure 5.3: Loss Graph.

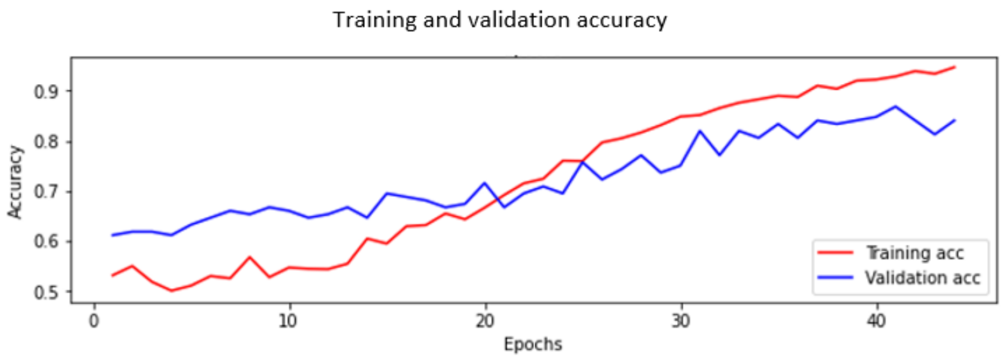


Figure 5.4: Accuracy Graph.

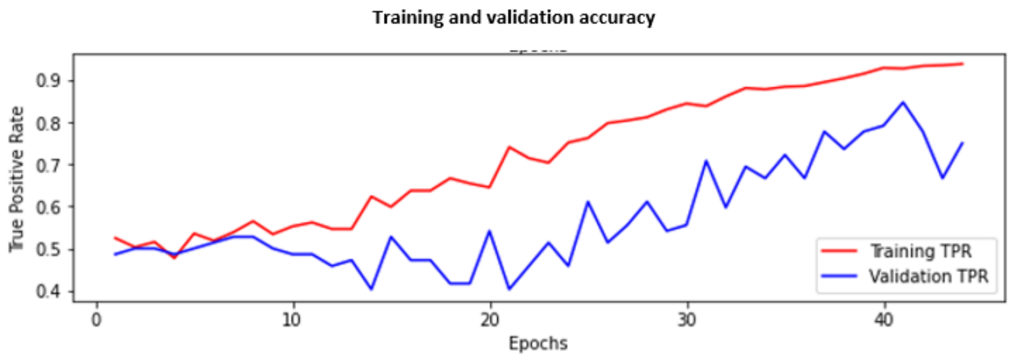


Figure 5.5: TPR Graph

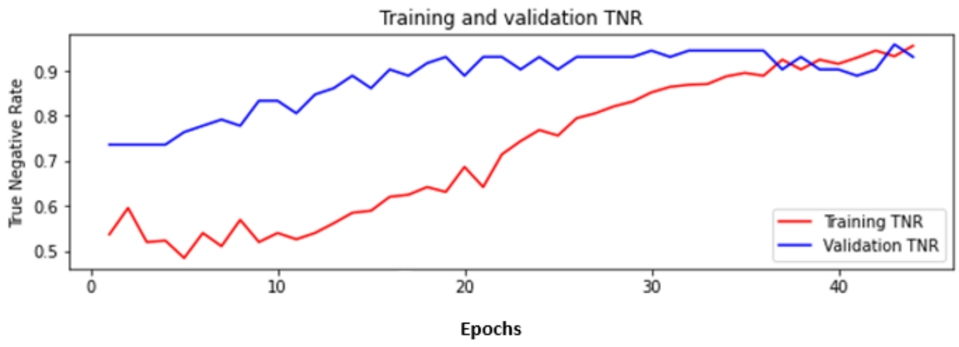


Figure 5.6: TNR Graph

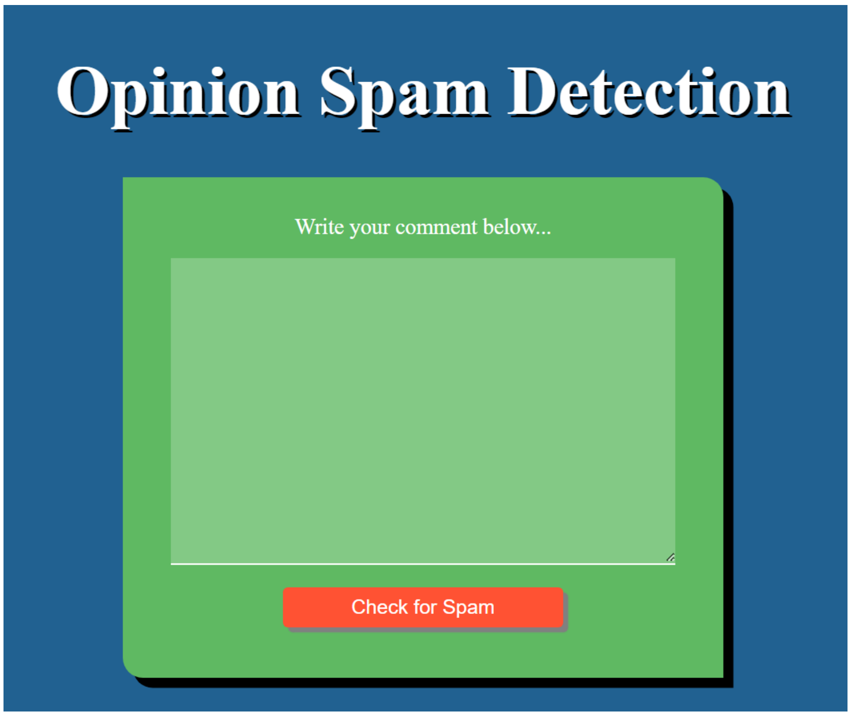
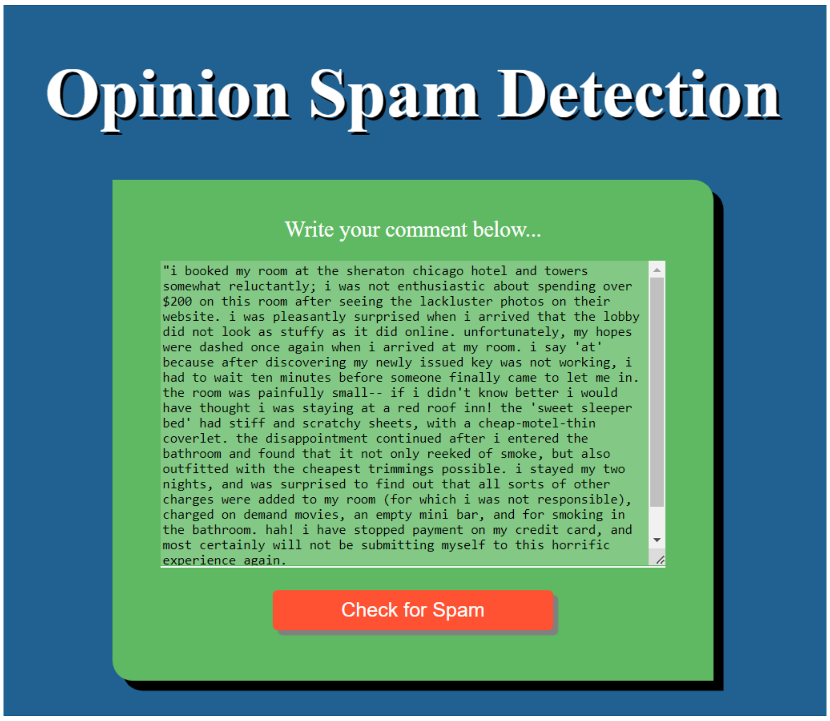


Figure 5.7: GUI (I)



"i booked my room at the sheraton chicago hotel and towers somewhat reluctantly; i was not enthusiastic about spending over $200 on this room after seeing the lackluster photos on their website. i was pleasantly surprised when i arrived that the lobby did not look as stuffy as it did online. unfortunately, my hopes were dashed once again when i arrived at my room. i say 'at' because after discovering my newly issued key was not working, i had to wait ten minutes before someone finally came to let me in. the room was painfully small-- if i didn't know better i would have thought i was staying at a red roof inn! the 'sweet sleeper bed' had stiff and scratchy sheets, with a cheap-motel-thin coverlet. the disappointment continued after i entered the bathroom and found that it not only reeked of smoke, but also outfitted with the cheapest trimmings possible. i stayed my two nights, and was surprised to find out that all sorts of other charges were added to my room (for which i was not responsible), charged on demand movies, an empty mini bar, and for smoking in the bathroom. hah! i have stopped payment on my credit card, and most certainly will not be submitting myself to this horrific experience again.

Figure 5.8: GUI (II)

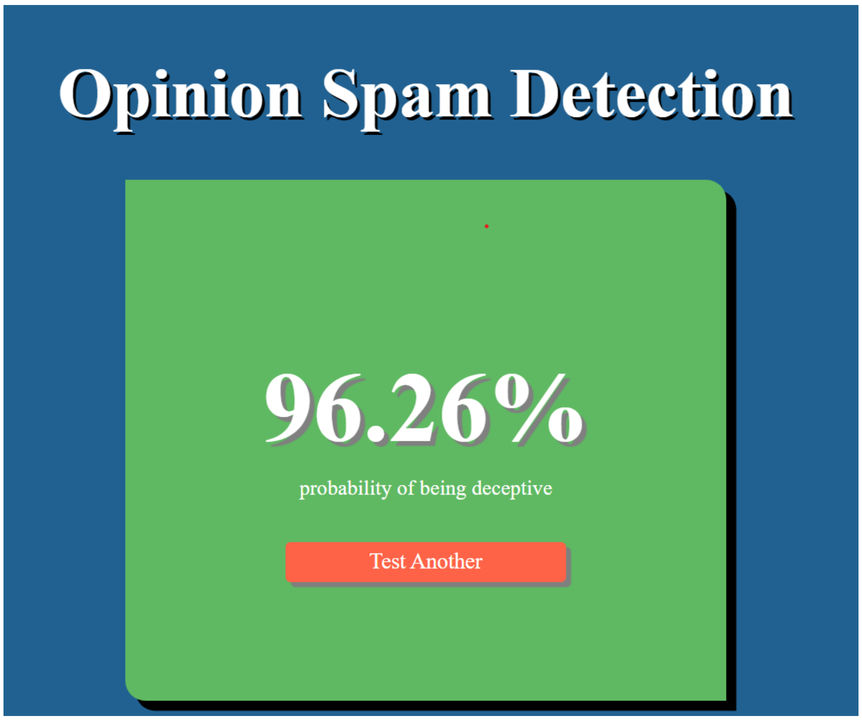


Figure 5.9: GUI (III)

CHAPTER 6

CONCLUSION

From the results mentioned in the previous section we can conclude the following:

1. This particular neural network has a good sensitivity and accuracy i.e., to correctly identify deceptive reviews.
2. From Fig. 5.3 and 5.4 we can conclude the loss and accuracy part. For every increasing epoch loss is decreasing and accuracy is increasing.
3. We are getting TPR: 0.94 and TNR: 0.88, from Fig 5.6 and 5.5, i.e., proposed model is more accurately identifying that review is deceptive.

This model can be incorporated into things like a web application, mobile application or a desktop application. It aims at using deep learning for helping people in making quick and accurate decisions on the basis of others reviews.

CHAPTER 7

FUTURE ENHANCEMENTS

While this neural network is very good in detecting that how likely the given review is a spam, there are several areas where it needs further improvement.

They are mentioned below:

1. We plan to expand our idea to detect not just individual spammers, but also groups of people that work closely to provide credit or disparage a product (or a service). Once a group of malicious reviewers has been identified, the objective is to look for overlap between the items that the malicious reviewers have reviewed. Colluders might be groups of users who have a lot of overlap (i.e., who have updated the same goods).

2. For future study we can use datasets with standard labels to train classifiers, and can add more attributes to dataset, so as to increase the accuracy and reliability of the model like IP address, signed in location of a reviewer from where that review is written. In addition, greater study into spam detection in multilingual reviews is required.

REFERENCES

1. Al-Adhaileh, M. H. and Alsaade, F. W. (2022). “Detecting and analysing fake opinions using artificial intelligence algorithms.” *INTELLIGENT AUTOMATION AND SOFT COMPUTING*, 32(1), 643–655.

2. Alsubari, S. N., Shelke, M. B., and Deshmukh, S. N. (2020). “Fake reviews identification based on deep computational linguistic.” *International Journal of Advanced Science and Technology*, 29, 3846–3856.

3. Fazzolari, M., Buccafurri, F., Lax, G., and Petrocchi, M. (2020). “Improving opinion spam detection by cumulative relative frequency distribution.” *arXiv preprint arXiv:2012.13905*.

4. Gan, S. H. “How to design a spam filtering system with machine learning algorithms. [Online]. Available from https://towardsdatascience.com/email-spam-detection-1-2-b0e06a5c0472.

5. Hussain, N., Turab Mirza, H., Rasool, G., Hussain, I., and Kaleem, M. (2019). “Spam review detection techniques: A systematic literature review.” *Applied Sciences*, 9(5), 987.

6. Jindal, N. and Liu, B. (2007). “Review spam detection.” *Proceedings of the 16th international conference on World Wide Web*. 1189–1190.

7. Jindal, N. and Liu, B. (2008). “Opinion spam and analysis.” *Proceedings of the 2008 international conference on web search and data mining*. 219–230.

8. Kaggle. “Deceptive opinion spam corpus. [Online]. Available from https://www.kaggle.com/datasets/rtatman/deceptive-opinion-spam-corpus.

9. Kaur, J. and Sabharwal, M. (2018). “Spam detection in online social networks using feed forward neural network.” *RSRI conference on recent trends in science and engineering*, Vol. 2. 69–78.

10. Khurshid, F., Zhu, Y., Zhuang, X., Ahmad, M., and Ahmad, M. (2018). “Enactment of ensemble learning for review spam detection on selected features.” *International Journal of Computational Intelligence Systems*, 12(1), 387.

11. Li, F. H., Huang, M., Yang, Y., and Zhu, X. (2011). “Learning to identify review spam.” *Twenty-second international joint conference on artificial intelligence*.

12. Mukherjee, A., Venkataraman, V., Liu, B., and Glance, N. (2013). “What yelp fake review filter might be doing?.” *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 7.

13. Narayan, R., Rout, J. K., and Jena, S. K. (2018). “Review spam detection using semi-supervised technique.” *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*, Springer, 281–286.

14. Ren, Y. and Ji, D. (2017). “Neural networks for deceptive opinion spam detection: An empirical study.” *Information Sciences*, 385, 213–224.

15. Salunkhe, A. (2021). “Attention-based bidirectional lstm for deceptive opinion spam classification.” *arXiv preprint arXiv:2112.14789*.

16. Savage, D., Zhang, X., Yu, X., Chou, P., and Wang, Q. (2015). “Detection of opinion spam based on anomalous rating deviation.” *Expert Systems with Applications*, 42(22), 8650–8657.

17. Tensorflow. “Keras. [Online]. Available from https://www.tensorflow.org/guide/keras.

18. Tensorflow. “Tensorflow. [Online]. Available from https://www.tensorflow.org/.

19. Tensorflow. “tf.keras.initializers.glorotuniform. [Online]. Available from https://www.tensorflow.org/api_docs/python/tf/keras/initializers/GlorotUniform.

20. Tensorflow. “tf.keras.optimizers.adamw. [Online]. Available from https://www.tensorflow.org/addons/api_docs/python/tfa/optimizers/AdamW.

21. Wang, Y., Liu, B., Wu, H., Zhao, S., Cai, Z., Li, D., and Fong, C. C. (2020). “An opinion spam detection method based on multi-filters convolutional neural network.” *CMC-COMPUTERS MATERIALS & CONTINUA*, 65(1), 355–367.

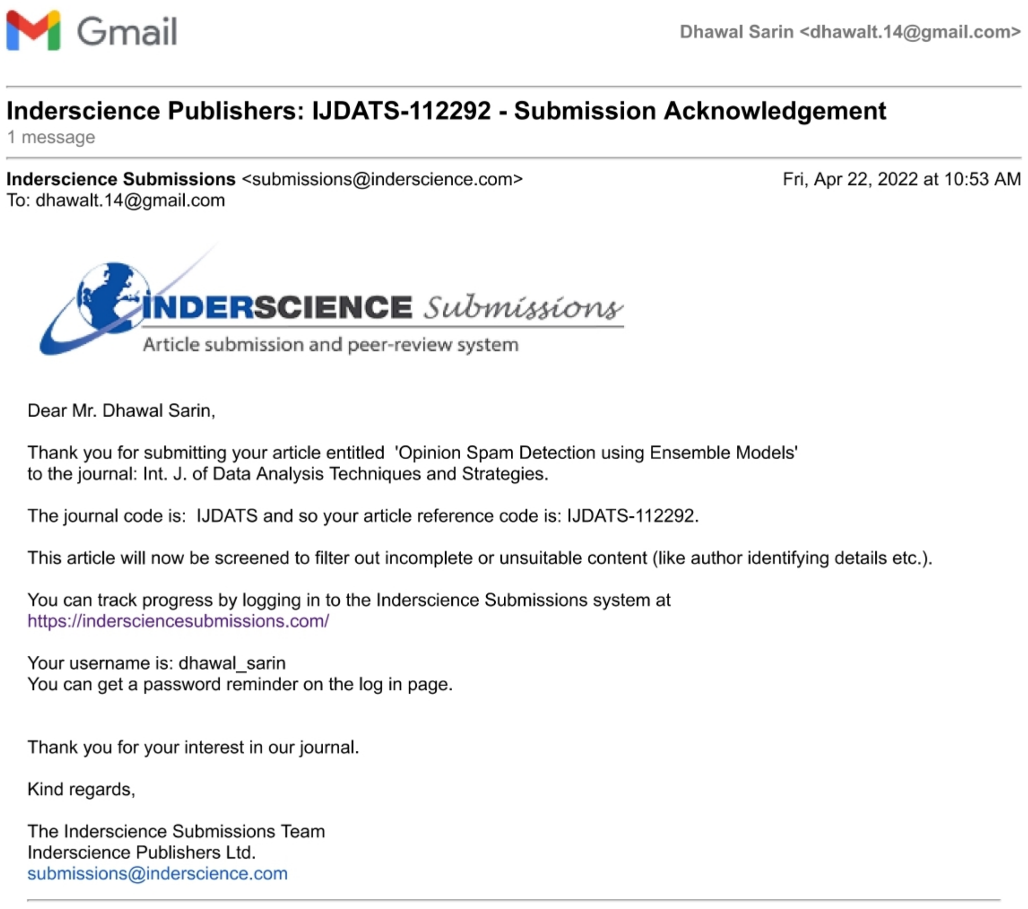
22. Wikipedia. “Cross entropy. [Online]. Available from https://en.wikipedia.org/wiki/Cross_entropy.

23. Xiang, L., Guo, G., Li, Q., Zhu, C., Chen, J., and Ma, H. (2020). "Spam detection in reviews using lstm-based multi-entity temporal features." *Intelligent Automation and Soft Computing*.

24. Xiang, L., Guo, G., Yu, J., Sheng, V. S., and Yang, P. (2020). "A convolutional neural network-based linguistic steganalysis for synonym substitution steganography." *Mathematical Biosciences and Engineering*, 17(2), 1041–1058.

25. Zhao, S., Xu, Z., Liu, L., Guo, M., and Yun, J. (2018). "Towards accurate deceptive opinions detection based on word order-preserving cnn." *Mathematical Problems in Engineering*, 2018.

ANNEXURE-I



Gmail

Dhawal Sarin <dhawalt.14@gmail.com>

Inderscience Publishers: IJDATS-112292 - Submission Acknowledgement

1 message

Inderscience Submissions <submissions@inderscience.com> Fri, Apr 22, 2022 at 10:53 AM

To: dhawalt.14@gmail.com

INDERSCIENCE Submissions
Article submission and peer-review system

Dear Mr. Dhawal Sarin,

Thank you for submitting your article entitled 'Opinion Spam Detection using Ensemble Models'
to the journal: Int. J. of Data Analysis Techniques and Strategies.

The journal code is: IJDATS and so your article reference code is: IJDATS-112292.

This article will now be screened to filter out incomplete or unsuitable content (like author identifying details etc.).

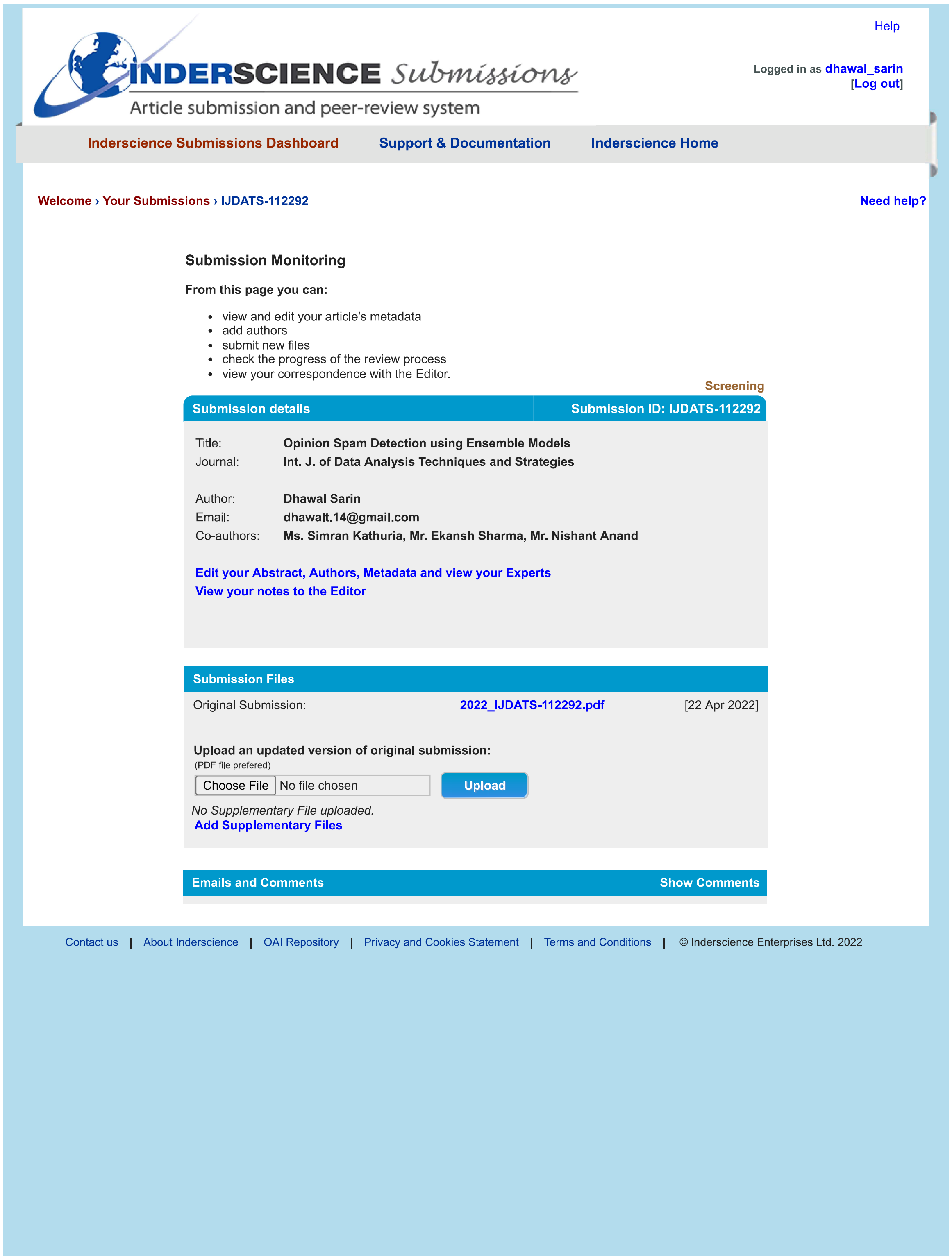
You can track progress by logging in to the Inderscience Submissions system at
https://indersciencesubmissions.com/

Your username is: dhawal_sarin
You can get a password reminder on the log in page.

Thank you for your interest in our journal.

Kind regards,

The Inderscience Submissions Team
Inderscience Publishers Ltd.
submissions@inderscience.com



Help

Logged in as dhawal_sarin
[Log out]

Inderscience Submissions Dashboard | Support & Documentation | Inderscience Home

Welcome › Your Submissions › IJDATS-112292

Need help?

Submission Monitoring

From this page you can:

- view and edit your article's metadata
- add authors
- submit new files
- check the progress of the review process
- view your correspondence with the Editor.

Screening

Submission details | **Submission ID: IJDATS-112292**

Title:	**Opinion Spam Detection using Ensemble Models**
Journal:	**Int. J. of Data Analysis Techniques and Strategies**
Author:	**Dhawal Sarin**
Email:	**dhawalt.14@gmail.com**
Co-authors:	**Ms. Simran Kathuria, Mr. Ekansh Sharma, Mr. Nishant Anand**

Edit your Abstract, Authors, Metadata and view your Experts
View your notes to the Editor

Submission Files

Original Submission:	**2022_IJDATS-112292.pdf**	[22 Apr 2022]

Upload an updated version of original submission:
(PDF file prefered)

Choose File No file chosen Upload

No Supplementary File uploaded.
Add Supplementary Files

Emails and Comments | **Show Comments**

Contact us | About Inderscience | OAI Repository | Privacy and Cookies Statement | Terms and Conditions | © Inderscience Enterprises Ltd. 2022

ANNEXURE-II

Opinion_spam_detection_using_ensemble_models

ORIGINALITY REPORT

8%	5%	6%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	download.atlantis-press.com Internet Source	1%
2	"Computational Intelligence in Information Systems", Springer Science and Business Media LLC, 2021 Publication	1%
3	"Encyclopedia of Social Network Analysis and Mining", Springer Science and Business Media LLC, 2018 Publication	1%
4	www.mdpi.com Internet Source	1%
5	Yuying Li, Yang Feng, Rui Hao, Di Liu, Chunrong Fang, Zhenyu Chen, Baowen Xu. "Classifying crowdsourced mobile test reports with image features: An empirical study", Journal of Systems and Software, 2022 Publication	<1%
6	Rajvardhan Oak, Min Du, David Yan, Harshvardhan Takawale, Idan Amit. "Malware Detection on Highly Imbalanced Data through	<1%

	Sequence Modeling", Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security - AISec'19, 2019 Publication	
7	pk.freedissertation.com Internet Source	<1 %
8	www.frontiersin.org Internet Source	<1 %
9	"Data Processing Techniques and Applications for Cyber-Physical Systems (DPTA 2019)", Springer Science and Business Media LLC, 2020 Publication	<1 %
10	Faisal Khurshid, Yan Zhu, Jie Hu, Muqeet Ahmad, Mushtaq Ahmad. "Battering Review Spam Through Ensemble Learning in Imbalanced Datasets", The Computer Journal, 2021 Publication	<1 %
11	arxiv.org Internet Source	<1 %
12	solar.njit.edu Internet Source	<1 %
13	espace.etsmtl.ca Internet Source	<1 %
14	link.springer.com Internet Source	

		<1%
15	psasir.upm.edu.my Internet Source	<1%
16	www.coursehero.com Internet Source	<1%
17	www.hindawi.com Internet Source	<1%
18	Naveed Hussain, Hamid Turab Mirza, Ghulam Rasool, Ibrar Hussain, Mohammad Kaleem. "Spam Review Detection Techniques: A Systematic Literature Review", Applied Sciences, 2019 Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		