DETECTING CONTRADICTION AND ENTAILMENT IN TEXT

A Project Report in partial fulfilment of the degree

Bachelor of Technology

in

Computer Science &Engineering / Electronics & Communication Engineering

By

19K41A0449	Mundru Sai Kumar
19K41A0536	Dharmula Akhil
19K41A0538	Goparaju Kalyani

Under the guidance of

Mr. D. RAMESH

Assistant Professor School of CS & AI SRU

Submitted to

DEPARTMENT OF COMPUTERSCIENCE&ENGINEERING

S.R. ENGINEERING COLLEGE (A), ANANTHASAGAR, WARANGAL

(Affiliated to JNTUH, Accredited by NBA) Nov-2022.

DEPARTMENT OFCOMPUTERSCIENCE& ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled "DETECTING CONTRADICTION AND ENTAILMENT IN TEXT" is a record of bonafide work carried out by the student(s) Dharmula Akhil, Goparaju Kalyani, Mundru Sai Kumar bearing Roll No(s) 19K41A0536, 19K41A0538, 19K41A0449 during the academic year 2022-23 in partial fulfillment of the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering/ Electronics & Communication Engineering** by the S.R. ENGINEERING COLLEGE, Ananthasagar, Warangal.

Supervisor

Head of the Department

External Examiner

ABSTRACT

As the world is progressing in terms of trends and technologies, we need to update ourselves with those trending technologies. Artificial Intelligence is one such most popular and advancing technology which is going to rule the world in future. AI is used to make the humans work easy without any hurdles. AI almost entrenched into each and every sector such as Health care, Finance, Transportation etc. But, can machines determine the relationships between sentences, or is that still left to humans? If NLP can be applied between sentences, this could have profound implications for fact-checking, identifying fake news, analyzing text, and much more. So we thought of developing an artificial intelligence application that helps us to classify the relationship between to text. The application takes the two texts as input and identifies the relationship present between two texts i.e contradiction, entailment or neutral. In order to develop this application. we have used a dataset from kaggle and build a model that helps us to identify the relationship of the two texts. The model performed well with an accuracy of 70% in classifying the relationship between the two texts.

Table of contents

1. INTRODUCTION

In the present indigenous world, each and every one are running behind the new advancing technologies in order to make their daily chores simpler. Artificial Intelligence (AI) is one such advancing technology that gained a lot if consideration in the present world. It is a part of computer science that focuses on designing intelligent computer systems that show the traits we re-late with human intelligence like comprehending languages, learning problem-solving, decision making, etc. One of the significant contributions of AI has remained in Natural Language Processing (NLP), which glued together linguistic and computational techniques to assist computers in understanding human languages and facilitating human-computer interaction. Machine Translation, Chat bots or Conversational Agents, Speech Recognition, Sentiment Analysis, Text summarization, etc., fall under the active research areas in the domain of NLP. However, in the past few years, Sentiment analysis has become a demanding realm.

Nowadays, Artificial Intelligence has spread its wings into Thinking Artificial Intelligence and Feeling Artificial Intelligence (Huang and Rust 2021). Thinking AI is de-signed to process information in order to arrive at new conclusions or decisions. The data are usually unstructured. Text mining, speech recognition, and face detection are all examples of how thinking AI can identify patterns and regularities in data. Machine learning and deep learning are some of the recent approaches to how thinking AI processes data. AI has made a big impact on the globe. AI was reintroduced in a significant manner in the twentieth century, and it inspired researchers to perform in-depth studies in domains like NLP, and machine learning. However, the domains of NLP remain ambiguous due to its computational methodologies, which assist computers in understanding and producing human-computer interactions in the form of text and voice.

Detection of relationship between two texts is one such area in which we use AI specifically Natural Language Processing (NLP) techniques to find the relationship of two sentences task much simpler.

Our brains process the meaning of a sentence like this rather quickly.

We're able to surmise:

- Some things to be true: "You can find the right answer through the process of elimination.”
- Others that may have truth: "Ideas that are improbable are not impossible!"

- And some claims are clearly contradictory: "Things that you have ruled out as impossible are where the truth lies."

If you have two sentences, there are three ways they could be related:

one could entail the other, one could contradict the other, or they could be unrelated. Natural

Our task is to create an NLP model that assigns labels of 0, 1, or 2 (corresponding to entailment, neutral, and contradiction) to pairs of premises and hypotheses.

In this we're classifying pairs of sentences (consisting of a premise and a hypothesis) into three categories - entailment, contradiction, or neutral. Let's take a look at an example of each of these cases for the following premise:

He came, he opened the door and I remember looking back and seeing the expression on his face, and I could tell that he was disappointed.

Hypothesis 1:

Just by the look on his face when he came through the door I just knew that he was let down.

We know that this is true based on the information in the premise. So, this pair is related by entailment.

Hypothesis 2:

He was trying not to make us feel guilty but we knew we had caused him trouble.

This very well might be true, but we can't conclude this based on the information in the premise. So, this relationship is neutral.

Hypothesis 3:

He was so excited and bursting with joy that he practically knocked the door off it's frame.

We know this isn't true, because it is the complete opposite of what the premise says. So, this pair is related by contradiction.

2. LITERATURE REVIEW

S. NO	Author	Methodology	Accuracy
1	Dorottya, Demszky, DanaMovshovitz-Attias, Jeongwoo Ko2	BERT based model Transfer learning	69%
2	Jacob Devlin	Pre-training BERT	92
3	Niklas Muennighoff	MTEB	100
4	PANTULKAR SRAVANTHI, DR. B. SRINIVASU	WordNet	80
5	Yuhua Li James Dominic O'Shea	Based on semantic and word order information	82
6	Xiaofei Sun	BERT and Glove	-
7	Marie-Catherine de Marneffe	RTE1	70
8	Kashif khan & Sher Hayat	Naive bayes classifier	80
9	Luyang Li, Bing Qin and Ting Liu	tailored neural network	85
10	Man Sherine Rady	Word2Vec CNN	84

3. DESIGN

3.1 REQUIREMENT SPECIFICATION(S/W & H/W)

Hardware Requirements

- ✓ **System** : Intel Core i3, i5, i7 and 2GHz Minimum
- ✓ **RAM** : 4GB or above
- ✓ **Hard Disk** : 10GB or above
- ✓ **Input** : Keyboard and Mouse
- ✓ **Output** : Monitor or PC

Software Requirements

- ✓ **OS** : Windows 8 or Higher Versions
- ✓ **Platform** : Jupyter Notebook, Google Colab
- ✓ **Program Language** : Python

3.2 FLOW CHART

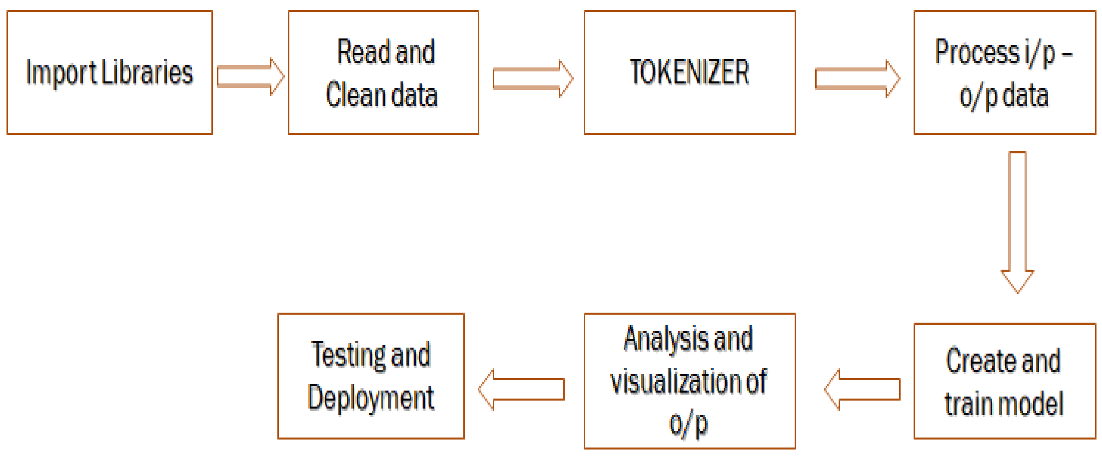


Figure 1: Flow chart

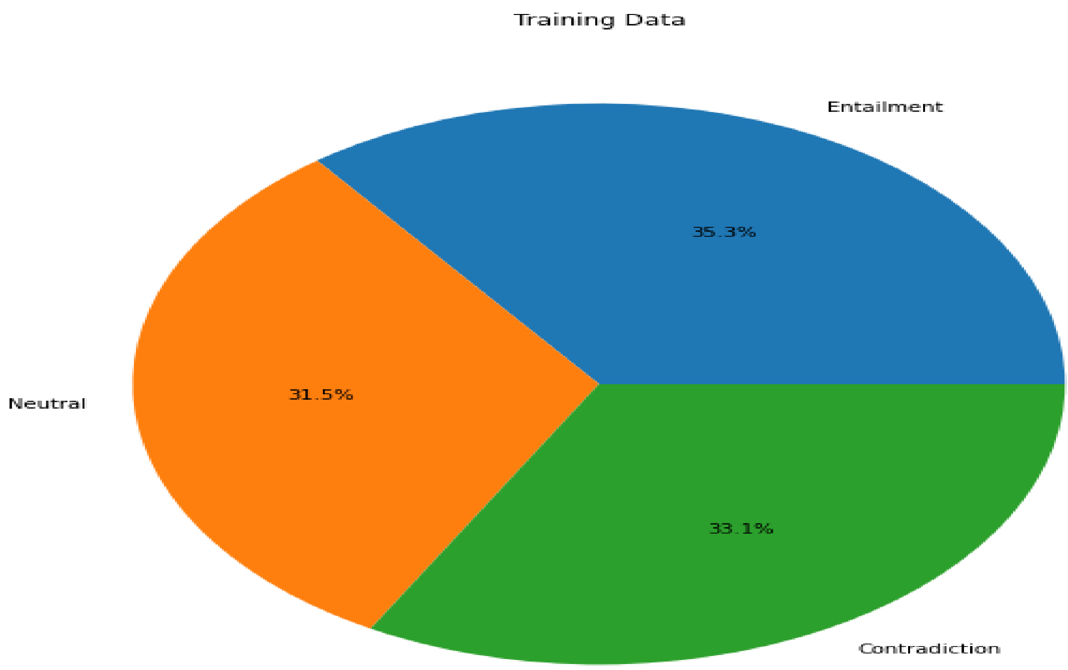
4. DATA SET

The Data set was obtained from open source Kaggle website. Data set contains 3 columns, Premise, Hypothesis and label Column has three categories like entailment, contradiction and neutral. Data set consist of 6871 rows unique sentences with their corresponding label. The data set was preprocessed and then it was model was trained using this data set. The test size is 0.20 and training size is 0.8 which means 20% used for testing and 80% for training. Our data set can recognize emotions like sadness, anger, love, surprise, fear, happy.

premise	hypothesis	label
and these comments were considered in formulat...	The rules developed in the interim were put to...	0
These are issues that we wrestle with in pract...	Practice groups are not permitted to work on t...	2
you know they can't really defend themselves l...	They can't defend themselves because of their ...	0
From Cockpit Country to St. Ann's Bay	From St. Ann's Bay to Cockpit Country.	2
Look, it's your skin, but you're going to be i...	The boss will fire you if he sees you slacking...	1

Figure.3 Data Set Sample

visualization the distribution of class labels over the data



5. DATA PRE-PROCESSING

```python
train_missing_values_count = train.isnull().sum() # we get the number of missing data point
print("Number of missing data points per column:\n")
print (train_missing_values_count)
```

```python
train["is_duplicate"] = train.duplicated()
train[train["is_duplicate"]==True].count()
```

```python
train.drop_duplicates(keep=False, inplace=True, ignore_index=True)
train.drop("is_duplicate", axis=1, inplace=True)
print("Number of data examples after dropping duplicates: {} \n".format(train.shape[0]))
```

Figure.4 Data pre-processing

Steps:

- Drop rows with NA values
- Drop NA may cause inconsistency in index so reset indexes
- Remove duplicate values

Split dataset:

Firstly split the dataset into features and target variable, then by using the train_test_split method, split the data into a training set and test set.

The test_size = 0.20 that is 20% of data for testing and remaining 80% for training purpose.

```python
from sklearn.model_selection import train_test_split
train, test = train_test_split(train, stratify=train.label.values,
                               random_state=42,
                               test_size=0.2, shuffle=True)


train.reset_index(drop=True, inplace=True)
test.reset_index(drop=True, inplace=True)
```

Figure 5: Splitting the data

6. METHODOLOGY

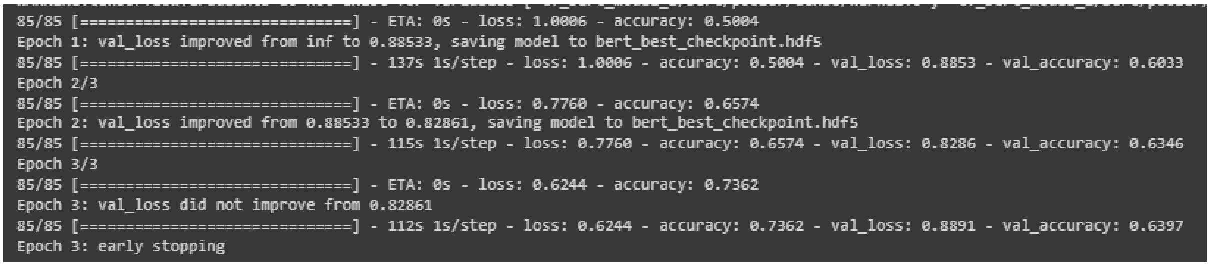
Steps:

1. Import the required libraries
2. Readthe dataset as .xlsx file
3. Perform missing value treatment by dropping columns with NA values
4. Duplicate values treatment
5. Add BERT tokenizer.
6. Performencoding
7. Define model
8. Add input layer
9. Add mask layer
10. Add attention layer
11. Adding embedding layer that can be used for neural networks on text data. It requires that the data to be integer encoded, so that each word is represented by unique value.it is initialized with random weights.
12. Adding dense layer with softmax activation function

```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_word_ids (InputLayer)    [(None, 100)]        0           []

 input_mask (InputLayer)        [(None, 100)]        0           []

 input_type_ids (InputLayer)    [(None, 100)]        0           []

 tf_bert_model_1 (TFBertModel)  TFBaseModelOutputWi  177853440   ['input_word_ids[0][0]',
                                thPoolingAndCrossAt               'input_mask[0][0]',
                                tentions(last_hidde               'input_type_ids[0][0]']
                                n_state=(None, 100,
                                 768),
                                 pooler_output=(Non
                                e, 768),
                                 past_key_values=No
                                ne, hidden_states=N
                                one, attentions=Non
                                e, cross_attentions
                                =None)

 tf.__operators__.getitem_1 (Sl  (None, 768)         0           ['tf_bert_model_1[0][0]']
 icingOpLambda)

 dense_1 (Dense)                (None, 3)            2307        ['tf.__operators__.getitem_1[0][0
                                                                 ]']

==================================================================================================
Total params: 177,855,747
Trainable params: 177,855,747
Non-trainable params: 0
__________________________________________________________________________________________________
```

Figure.8. Model Summary

7.RESULTS



```
85/85 [==============================] - ETA: 0s - loss: 1.0006 - accuracy: 0.5004
Epoch 1: val_loss improved from inf to 0.88533, saving model to bert_best_checkpoint.hdf5
85/85 [==============================] - 137s 1s/step - loss: 1.0006 - accuracy: 0.5004 - val_loss: 0.8853 - val_accuracy: 0.6033
Epoch 2/3
85/85 [==============================] - ETA: 0s - loss: 0.7760 - accuracy: 0.6574
Epoch 2: val_loss improved from 0.88533 to 0.82861, saving model to bert_best_checkpoint.hdf5
85/85 [==============================] - 115s 1s/step - loss: 0.7760 - accuracy: 0.6574 - val_loss: 0.8286 - val_accuracy: 0.6346
Epoch 3/3
85/85 [==============================] - ETA: 0s - loss: 0.6244 - accuracy: 0.7362
Epoch 3: val_loss did not improve from 0.82861
85/85 [==============================] - 112s 1s/step - loss: 0.6244 - accuracy: 0.7362 - val_loss: 0.8891 - val_accuracy: 0.6397
Epoch 3: early stopping
```

Figure.9. Accuracy

For evaluating the performance of the proposed model training and testing accuracies are very useful. To get better accuracy the model needs to be trained using different epochs. We trained the data set using our model. We used 3 epochs to train the data. We found the accuracy of our proposed model is around 63%.

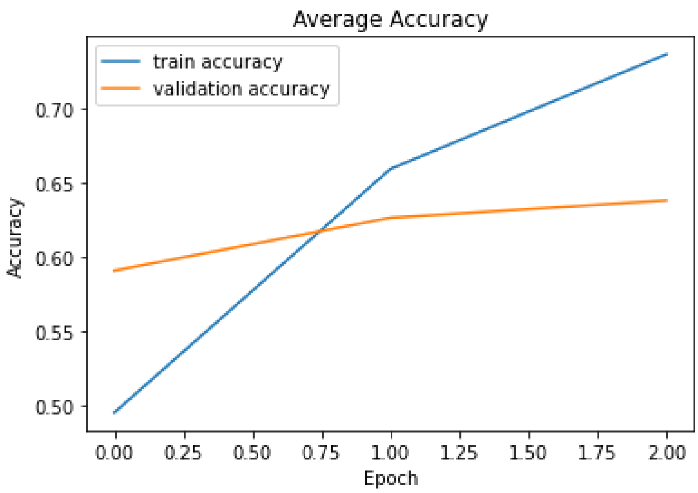


Figure.10. Model accuracy train and validation vs epochs

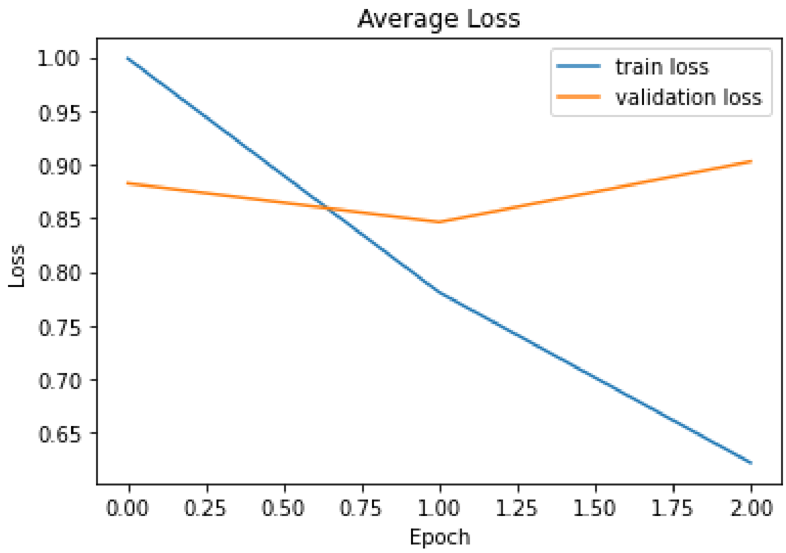


Figure.11. Model loss train and validation vs epochs

result :

```
premise = 'Why is it you wish to die?'
hypothesis = 'Is there any value to this thing called living?'

predict_inference(premise, hypothesis, model, device)
```

```
'entailment'
```

8. CONCLUSION

We developed an application that is used to identify relationship between the two texts or sentences. It classifies the relationship into three categories i.e one could entail the other (entailment), one could contradict the other (contradiction), or they could be unrelated (neutral). The model performed well in task with an accuracy of 70%. In future the application can be developed to identify the relationship of the sentences of different languages.

9. REFERENCES

[1]Sherine Rady, and Mostafa Aref, “A Deep Learning Architecture with Word Embeddings to Classify Sentiment in Twitter”, Springer Nature Switzerland AG 2021, A. E. Hassanien et al. (Eds.): AISI 2020, AISC 1261, pp. 115-125,0032021

[2] D. Seal, U. K. Roy, and R. Basak, “Sentence-level emotion detection from text based on semantic rules,” Information and Communication Technology for Sustainable Development, Springer, Singapore, pp. 423–430, 2020.

[3] Santosh Kumar Bharti, S Varadhaganapathy, Rajeev Kumar Gupta, Prashant Kumar Shukla, Mohamed Bouye, Simon Karanja Hingaa, Amena Mahmoud, "Text-Based Emotion Recognition Using Deep Learning Approach", *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2645381, 8 pages, 2022. https://doi.org/10.1155/2022/2645381

[4] Khan, Kashif & Ejaz, Muhammad. (2016). Emotion Detection through Text.

[5]Hongyu Han, Yongshi Zhang, Jianpei Zhang, Jing Yang, and Yong Wang, “A Hybrid Sentiment Analysis Method”, Springer Nature Singapore Pte Ltd. 2021, Q. Liu et al. (Eds.): CENet 2020, AISC 1274, pp. 1135-1146, 2021.

[6]NourahAlswaidan, Mohamed El Bachir Menai, “A survey of state-ofthe-art approaches for emotion Recognition in the text”, ©SpringerVerlagLondonLtd.,partofSpringerNature2020.

[7]ditya Dave, Santosh Bharti, Samir Patel, and Sam bit Kumar Mishra, “A Real-Time Sentiments Analysis System Using Twitter Data”, O Springer Nature Singapore Pte Ltd. 2021, D. Mishra ct al. (eds.), liitelligetitu+id Cloud Computing, Srnan Innovation, Systeriis and Technologies 153