SENTIMENT ANALYSIS ON TEXT USING BERT NEURAL NETWORK

1,2,3,4 **Jella Eshwar, S Prasad, Penchala Divyasri, K. Anand**

5 **Y.Shiva Rao**

1,2,3,4 Ug scholars, MallaReddy college Of Engineering , Hyderabad - 500100

5 Assistant Professor, MallaReddy college Of Engineering , Hyderabad - 500100

ABSTRACT

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

Sentiment analysis has emerged as a vital tool for understanding public opinion and gauging sentiment within textual data. In the contemporary era of data-driven decision-making, sentiment analysis holds significant importance across various domains, including marketing, customer feedback analysis, and social media monitoring. The ability to accurately discern sentiment from text data can provide valuable insights for businesses and organizations. In this context, the utilization of advanced deep learning techniques, particularly BERT (Bidirectional Encoder Representations from Transformers) neural network, presents a compelling approach for sentiment analysis. BERT, renowned for its ability to capture contextual information and effectively process bidirectional sequences, has shown remarkable performance in natural language processing tasks.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

There has been a constant growth in the public and private information stored within the internet. This includes textual data expressing people's opinions on review sites, forums, blogs, and other social media platforms. Review based prediction systems allow this unstructured information to be automatically transformed into structured data reflecting public opinion.

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEMS

Historically, sentiment analysis has relied on traditional lexicon-based methods and statistical algorithms to interpret the sentiment of textual data. Lexicon-based techniques involved the usage of sentiment lexicons and dictionaries to assign sentiment scores to words within the text. However, these approaches often encountered difficulties in handling contextual nuances and keeping pace with evolving language usage trends.

3.2 DRAWBACKS

1. **Limited Contextual Understanding:** Traditional lexicon-based methods and statistical algorithms often struggle to capture the nuanced contextual understanding required for accurate sentiment analysis, leading to potential misinterpretation of sentiment in complex texts.

2. **Difficulty in Handling Sarcasm and Irony:** Existing systems may struggle to discern sentiment in instances of sarcasm and irony, as these rely heavily on contextual and cultural understanding, posing challenges for accurate sentiment classification.

3.3 PROPOSED SYSTEMS

In our project, we introduce a revolutionary shift in sentiment analysis by leveraging the power of BERT. By integrating BERT into sentiment analysis, our project aims to usher in a new era of precision and context-aware sentiment classification, enhancing decision support and insights for individuals and businesses. The landscape of sentiment analysis underwent a paradigm shift with the introduction of BERT (Bidirectional Encoder Representations from Transformers) and transformer-based architectures.

3.4 Advantages

1. **Enhanced Contextual Understanding:** The utilization of BERT facilitates a deeper comprehension of the contextual nuances within textual data, enabling the accurate interpretation of sentiment within diverse and complex language structures.

2. **Improved Accuracy and Consistency:** BERT's bidirectional processing empowers the model to capture intricate dependencies within the text, leading to heightened accuracy and consistency in sentiment analysis compared to traditional methods.

3.5 SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS :

IDE	:	Google colab
LANGUAGE	:	Python
LIBRARIES	:	BERT, SCI-KIT learn, Pandas,Numpy,Matplotlib

HARDWARE REQUIREMENTS:

RAM Minimum	:	16GB
CPU	:	I5 Processor
SSD	:	256 GB
GPU		

CHAPTER-4 SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

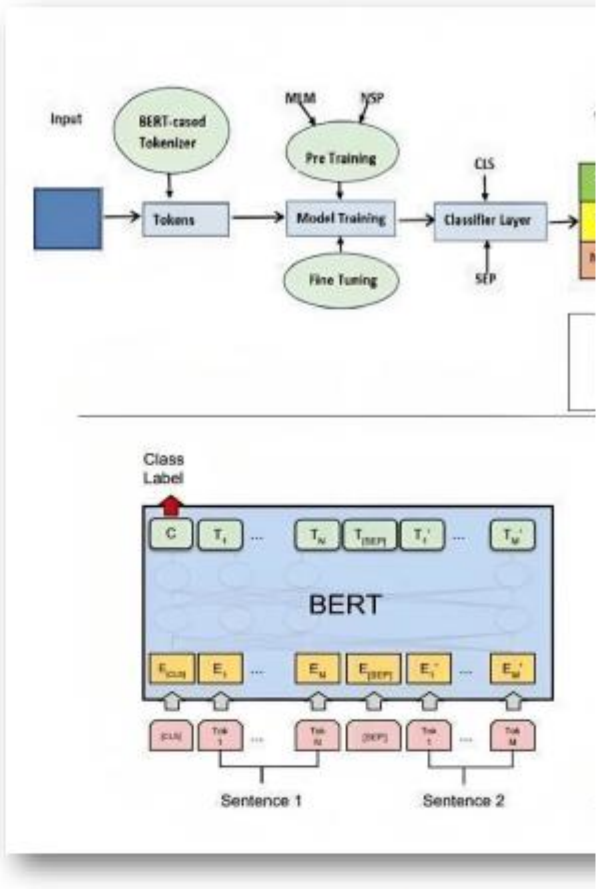


Fig 4.1.1

1. **Input:** the input is the text data you want to perform sentiment analysis on. This text data is what you want BERT to analyze and make predictions about.

2. **Token:** In NLP, a token is a unit of text, typically a word or a subword. BERT tokenizes your input text into these smaller units, which helps the model understand the text better.

4.2 Modules

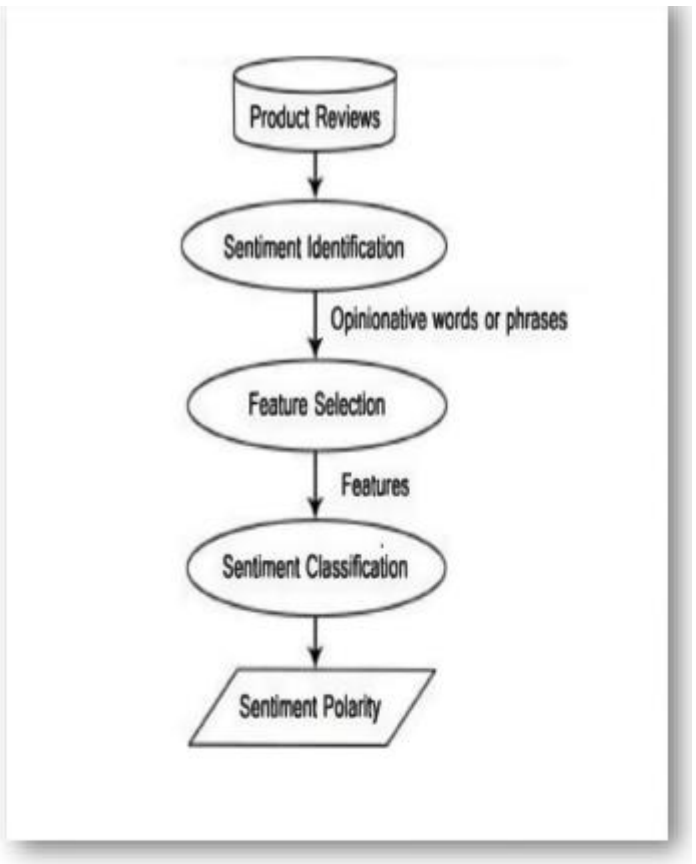


Fig 4.2.1

1. Product Review: This is the initial data source for your project. It refers to the text reviews or comments about a product, which serve as the input for sentiment analysis.

These reviews can come from various sources like e-commerce websites, social media, or customer feedback forms.

2. Sentiment Identification: This step involves identifying and extracting the sentiment from the product reviews. Sentiment identification can be thought of as the preliminary analysis, where you determine whether the sentiment is positive, negative, or neutral. BERT, as mentioned earlier, plays a crucial role in this step.

4.3 UML DIAGRAMS

1.CLASS DIAGRAM:

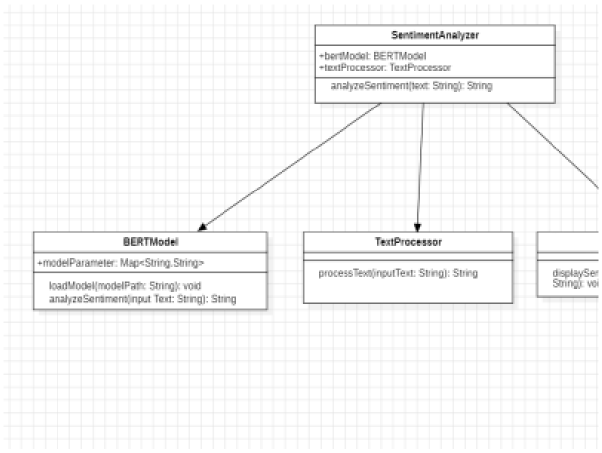

SentimentAnalyzer
+bertModel: BERTModel
+textProcessor: TextProcessor
analyzeSentiment(text: String): String
BERTModel
+modelParameter: Map<String,String>
loadModel(modelPath: String): void
analyzeSentiment(input Text: String): String
TextProcessor
processText(inputText: String): String
displaySer
String): voi

Fig 4.3.1

2.USECASE DIAGRAM:

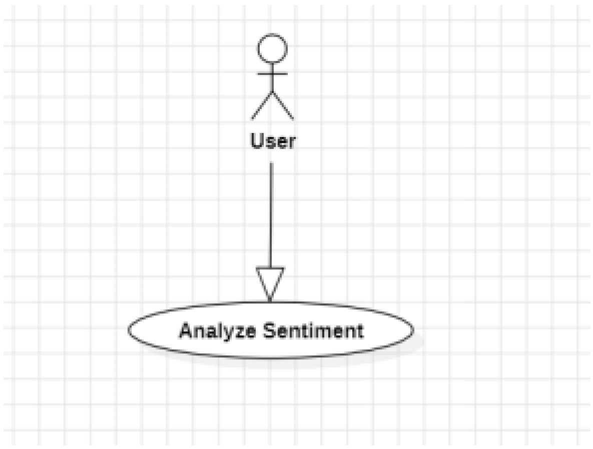


Fig 4.3.2

CHAPTER-5

SYSTEM IMPLEMENTATION

2.1 DEEP LEARNING

Deep learning is a subset of machine learning that involves neural networks with three or more layers. These neural networks attempt to simulate the behaviour of the human brain to "learn" from large amounts of data. Deep learning algorithms strive to automatically learn hierarchical representations of data, leading to better feature learning and abstraction.

2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) that focuses on the interaction between computers and humans through natural language. The primary goal of NLP is to enable machines to understand, interpret, and generate human-like language.

2.3 Python

Python programming language is used for building the machine learning model

2.3.1 Introduction

Python is an object-oriented, high level language, interpreted, dynamic and multipurpose programming language. Python is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development. Python's syntax and dynamic typing with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas. Python supports multiple programming pattern, including object oriented programming, imperative and functional programming or procedural styles. Python is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc.

2.4 FUNDAMENTALS OF PYTHON

This section contains the basic fundamentals of Python.

2.4.1 Tokens

Tokens can be defined as a punctuator mark, reserved words and each individual word in a statement. Token is the smallest unit inside the given program. Tokens include Keywords, Identifiers, Literals, Operators.

2.5 Google Colab

Google Colab, short for Google Colaboratory, is a cloud-based platform provided by Google that allows users to write, execute, and share Python code in a collaborative environment.

5.7 Algorithm

The BERT (Bidirectional Encoder Representations from Transformers) model is being used as part of the algorithm for sentiment analysis in the project. BERT is a powerful natural language processing model developed by Google, capable of understanding context and meaning in language. It has been widely adopted for various NLP tasks, including sentiment analysis, due to its ability to capture complex language patterns.

5.8 Packages

5.8.1. pandas (pd): This module is used for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series.

5.8.2. numpy (np): NumPy is a fundamental package for scientific computing with Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

5.9 Source Code

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set(style='whitegrid',font_scale=1.2)

sns.set_palette(sns.color_palette("rocket
"))

from sklearn.model_selection import
train_test_split

from sklearn.metrics import
confusion_matrix, classification_report

import transformers

from transformers import BertModel,
BertTokenizer, AdamW,
get_linear_schedule_with_warmup

import torch

from torch import nn, optim

from torch.utils.data import Dataset,
DataLoader

 from collections import defaultdict

# ignore the warnings

import warnings

warnings.filterwarnings('ignore')

# Let's start by defining some key
variables that will be used later on in the
training/evaluation process

RANDOM_SEED = 50

BATCH_SIZE = 16 # Note that
increasing the batch size reduces the
training time significantly, but gives you
lower accuracy.
```

```
# Set seed for reproducibility.
np.random.seed(RANDOM_SEED)
torch.manual_seed(RANDOM_SEED)
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
df_reviews = pd.read_csv("bert_sentiment.csv")
# We have all building blocks required to create a torch dataset. Let's do it...
class dataset(Dataset):
  def __init__(self, reviews, targets, tokenizer, max_len):
    self.reviews = reviews
    self.targets = targets
    self.tokenizer = tokenizer
    self.max_len = max_len

  def __len__(self):
    return len(self.reviews)

  def __getitem__(self, item):
    # step 1: get the reviews and targets
    review = str(self.reviews[item])
    target = self.targets[item]
```

CHAPTER-6
TESTING

6.1 TESTING

The various levels of testing are

1. White Box Testing
2. Black Box Testing
3. Unit Testing
4. Functional Testing

1. White Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing).

2. Black Box Testing

- Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing).

3. Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application.

4. Functional Testing

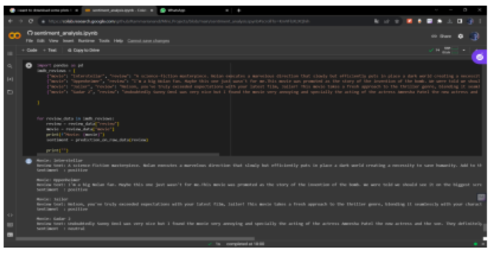
Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes *what* the system does.

CHAPTER-7

RESULTS

7.1 SCREENSHORTS

sentiment analysis on movie reviews

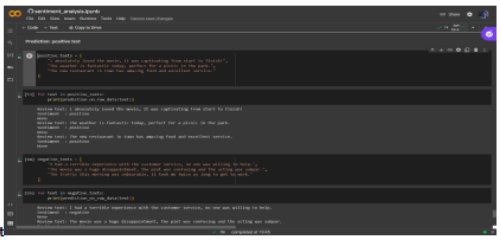


```
import pandas as pd
imdb_reviews = [
    {"movie": "Interstellar", "review": "A science-fiction masterpiece. Nolan executes a marvelous direction that slowly but efficiently puts in place a dark world creating a necessit
    {"movie": "Oppenheimer", "review": "I'm a big Nolan fan. Maybe this one just wasn't for me.This movie was promoted as the story of the invention of the bomb. We were told we shoul
    {"movie": "Jailer", "review": "Nelson, you've truly exceeded expectations with your latest film, Jailer! This movie takes a fresh approach to the thriller genre, blending it seaml
    {"movie": "Gadar 2", "review": "Undoubtedly Sunny Deol was very nice but I found the movie very annoying and specially the acting of the actress Ameesha Patel the new actress and

]


for review_data in imdb_reviews:
    review = review_data["review"]
    movie = review_data["movie"]
    print(f"Movie: {movie}")
    sentiment = prediction_on_raw_data(review)

    print("")
```

```
Movie: Interstellar
Review text: A science fiction masterpiece. Nolan executes a marvelous direction that slowly but efficiently puts in place a dark world creating a necessity to save humanity. Add to th
Sentiment  : positive

Movie: Oppenheimer
Review text: I'm a big Nolan fan. Maybe this one just wasn't for me.This movie was promoted as the story of the invention of the bomb. We were told we should see it on the biggest scre
Sentiment  : positive

Movie: Jailer
Review text: Nelson, you're truly exceeded expectations with your latest film, Jailer! This movie takes a fresh approach to the thriller genre, blending it seamlessly with your charact
Sentiment  : positive

Movie: Gadar 2
Review text: Undoubtedly Sunny Deol was very nice but I found the movie very annoying and specially the acting of the actress Ameesha Patel the new actress and the son. They definitely
Sentiment  : neutral
```

Fig 7.1.1 prediction on movie reviews

3. Prediction on positive

text


Prediction: positive text

```
positive_texts = [
    "I absolutely loved the movie, it was captivating from start to finish!",
    "The weather is fantastic today, perfect for a picnic in the park.",
    "The new restaurant in town has amazing food and excellent service."
]
```

```
[13] for text in positive_texts:
         print(prediction_on_raw_data(text))
```

```
Review text: I absolutely loved the movie, it was captivating from start to finish!
Sentiment  : positive
None
Review text: The weather is fantastic today, perfect for a picnic in the park.
Sentiment  : positive
None
Review text: The new restaurant in town has amazing food and excellent service.
Sentiment  : positive
None
```

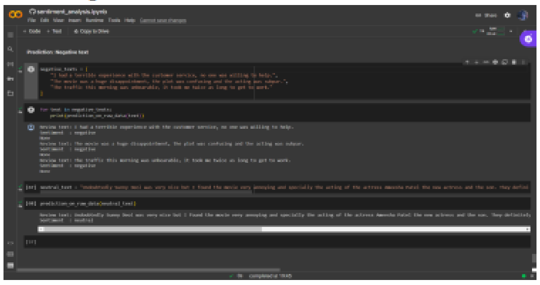
```
[14] negative_texts = [
         "I had a terrible experience with the customer service, no one was willing to help.",
         "The movie was a huge disappointment. The plot was confusing and the acting was subpar.",
         "The traffic this morning was unbearable, it took me twice as long to get to work."
     ]
```

```
[15] for text in negative_texts:
         print(prediction_on_raw_data(text))
```

```
Review text: I had a terrible experience with the customer service, no one was willing to help.
Sentiment  : negative
None
Review text: The movie was a huge disappointment. The plot was confusing and the acting was subpar.
```

Fig 7.1.2 prediction on positive sentence

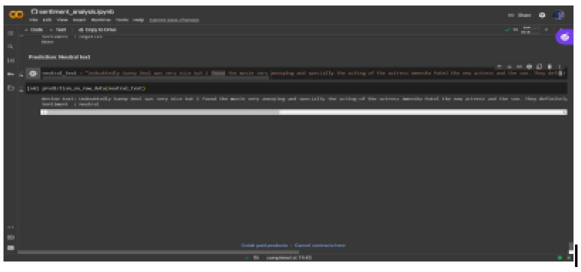
3.Prediction on negative text


sentiment_analysis.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Share
Prediction: Negative text

```
negative_texts = [
    "I had a terrible experience with the customer service, no one was willing to help.",
    "The movie was a huge disappointment, the plot was confusing and the acting was subpar.",
    "The traffic this morning was unbearable, it took me twice as long to get to work."
]

for text in negative_texts:
    print(prediction_on_raw_data(text))

Review text: I had a terrible experience with the customer service, no one was willing to help.
Sentiment  : negative
None
Review text: The movie was a huge disappointment, the plot was confusing and the acting was subpar.
Sentiment  : negative
None
Review text: The traffic this morning was unbearable, it took me twice as long to get to work.
Sentiment  : negative
None

[41] neutral_text = "Undoubtedly Sunny Deol was very nice but I found the movie very annoying and specially the acting of the actress Ameesha Patel the new actress and the son. They defini

[48] prediction_on_raw_data(neutral_text)

Review text: Undoubtedly Sunny Deol was very nice but I found the movie very annoying and specially the acting of the actress Ameesha Patel the new actress and the son. They definitely
Sentiment  : neutral

[ ]
```

Fig 7.1.3 prediction on negative sentence

4.Prediction on neutral text


sentiment_analysis.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Share
Prediction: Neutral text

```
neutral_text = "Undoubtedly Sunny Deol was very nice but I found the movie very annoying and specially the acting of the actress Ameesha Patel the new actress and the son. They defini

[48] prediction_on_raw_data(neutral_text)

Review text: Undoubtedly Sunny Deol was very nice but I found the movie very annoying and specially the acting of the actress Ameesha Patel the new actress and the son. They definitely
Sentiment  : neutral
```

Fig 7.1.4 prediction on neutral sentence

CHAPTER-8

CONCLUSION

8.1 CONCLUSION

In summary, our project, 'Sentiment Analysis on Text Using BERT Neural Network,' marks a significant leap forward in the realm of sentiment analysis. By embracing the power of BERT, we've addressed the limitations of traditional sentiment analysis methods. Our approach offers unparalleled accuracy, making it a valuable tool for assessing and understanding sentiment in text. Importantly, this is not just an academic endeavour; it has real-world implications. Our solution is designed for practical applications, offering insights and decision support to individuals and businesses. BERT's adaptability across languages and domains, combined with its reduced data dependency, positions it as a versatile and scalable solution.

CHAPTER-9

FUTURE ENCHANTMENT

9.1FUTURE ENHANCEMENT

1. Multi-lingual Sentiment Analysis: Extending the BERT-based sentiment analysis model to support multiple languages, thereby broadening its applicability to diverse linguistic contexts.

2. Fine-grained Sentiment Classification: Refining the model to categorize sentiments into more nuanced categories, such as sentiment intensity levels or specific emotions, to provide deeper insights into textual sentiment**.**

REFERENCES :

- Rishi Meridian, “Analysis of Google Play Store Data set and predict the popularity of an app on Google Play Store", Texas a&M University College Station, Texas, June-2020. Jong, J. (2011). Predicting rating with sentiment analysis

- William B Dolan and Chris Brockett. 2005. Automati cally constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005).

- Harman, M., Jiao, Y., and Zhang, Y. (2012). App store mining and analysis: Mrs for app Stores. In 2012 9th IEEE Working Conference on Mining Software Repositories (MSR), Pages 108–111.

- Https://developer.mozilla.org/en-US/docs/Learn/Serverside/Django/Introduction