SENTIMENT ANALYSIS FOR MARKETING

810621104019:PRIYANGA.K

PHASE 2 SUBMISSION DOCUMENT

INTRODUCTION

- Human decision making or thinking is always affected by others thinking,ideas and opinions.The growth of social web gives a huge amount of user generated data such as comments,opinions and reviews about products,services and events.
- This data will be userpful for consumers as well as manufacturer .While buying any product online consumers usually check comments or opinion of others about the product.
- Sentiment analysis does the classification of opinions in the text into catogories like"positive" or "negative" or "neutral".

HOW TO CLASSIFY SENTIMENT

1. Machine Learning/Automatic

This approach ,employes a machine-learning technique and diverse features to construct a classifier that can identify text that expresses sentiment.

2. Lexical-based/Rule-based

This method uses a variety of words annotated by polarity score,to decide the general assessment score of a given content.

3.Hybrid

The combination of machine learning and lexicon-based approaches to address sentiment analysis is called hybrid.Though not commonly used,this method usually produces more promising results than the approches mentioned above.

WHAT DOES SENTIMENT ANALYSIS MEAN:

The process of computationally identifying and categorizing opinions expressed in a piece of text,especially in order to determine whether the writer's attitude towards a particular topics,product,etc.is positivenegative,neutral.

SENTIMENT ANALYSIS CAN USED AS FOLLOWS:

- Social media monitoring
- Brand monitoring
- Voice of customer(VoC)
- Workforce analytics and voice of employee
- Product analytics
- Market research and analysis

WHAT IS TEXTBLOB?

Textblob is a python library and offers a simple

API to access its methods and perform basic NLP tasks.

The sentiment function of textblob returns two properties,polarity,and subjectivity.

Polarity is float which lies in the range of [-1,1]where 1 means a negative ststement.Subjective sentences generally refer to personal opinion,emotion or judgement whereas objective refers to factual information.Subjectivity is also a float which lies in the range of[0,1].

ENSEMBLE METHODS:

Bagging: Techniques like Random Forest or Bagged Decision Trees can be employed to create an ensemble of sentiment classifiers. Each classifier is trained on a subset of the data, and their predictions are combined to produce a final sentiment score.

Boosting: Algorithms like AdaBoost or Gradient Boosting can improve sentiment analysis by giving more weight to misclassified data points in each iteration, leading to better overall accuracy.

Deep Learning Architectures:

1.Convolutional Neural Networks (CNNs):

CNNs can be used for sentiment analysis by treating text as an image, converting words into vectors, and using convolutional layers to detect important features.

2.Recurrent Neural Networks (RNNs):

RNNs, particularly LSTM (Long Short-Term

Memory) and GRU (Gated Recurrent Unit) variants, are effective for sequence modeling, making them suitable for sentiment analysis tasks where the order of words matters.

3.Transformer Models:

State-of-the-art models like BERT, GPT, and RoBERTa have revolutionized natural language understanding tasks, including sentiment analysis. They can capture context and nuances in text effectively.

Bagging:

I/N:

Accuracy = accuracy_score(y_test, y_pred)

Report = classification_report(y_test, y_pred)

Print(fAccuracy: {accuracy})

Print(Classification Report:\n, report)

Boosting:

I/N:

Y_pred = ada_boost_classifier.predict(X_test_tfidf)

Accuracy = accuracy_score(y_test, y_pred)

Print(fAccuracy: {accuracy:.2f})

Recurrent neural network:

I/N:

Model.compile(optimizer=adam,
loss=binary_crossentropy,

metrics=[accuracy])

Model.fit(X_train, y_train, epochs=5, batch_size=64,

validation_data=(X_test, y_test))

Loss, accuracy = model.evaluate(X_test, y_test)

Print(fLoss: {loss}, Accuracy: {accuracy})

Print(fText: {text}\nSentiment: {Positive if sentiment > 0.5 else Negative})

Convolution neural network:

I/N

Texts = [This is a positive review., Negative sentiment in this one., …]

Model = keras.Sequential

Test_loss, test_acc = model.evaluate(x_test, y_test)

Print(Test accuracy:, test_acc)

BERT:

I/N:

Model_name = bert-base-uncased # You can choose different BERT variants

Tokenizer = BertTokenizer.from_pretrained(Tweet)

Model =BertForSequenceClassification.from_pretrained(Tweet)

With torch.no_grad():

Outputs = model(**inputs)

Sentiment_labels = {0: Negative, 1: Neutral, 2: Positive}

Sentiment = sentiment_labels[predicted_label]

Text_to_analyze = Positive

Result = analyze_sentiment(text_to_analyze)

Print(fSentiment: {result})

<u>RoBERTa:</u>

I/N:

Tokenizer = RobertaTokenizer.from_pretrained(Tweet)

ModelRobertaForSequenceClassification.from_pretrained(Tweet)

Logits = outputs.logits

Sentiment_labels = [Negative, Neutral, Positive]

Sentiment = sentiment_labels[predicted_class]

Return sentiment, logits.tolist()

Text = Positive

Sentiment, scores = analyze_sentiment(Positive)

Print(fSentiment: {sentiment})

Print(fSentiment Scores: {scores})

<u>GPT-2:</u>

I/N:

Model_name = gpt2 # You can also specify other variants like gpt2-medium, gpt2-large, etc.

Tokenizer = GPT2Tokenizer.from_pretrained(Tweet)

```
Model = GPT2LMHeadModel.from_pretrained(Tweet)
Input_ids = tokenizer.encode(prompt, return_tensors=pt)
Generated_text = tokenizer.decode(output[0],
skip_special_tokens=True)
Print(generated_text)
From mpl_toolkits.mplot3d import Axes3D
From sklearn.preprocessing import StandardScaler
Import matplotlib.pyplot as plt
Import numpy as np
Import os
Import pandas as pd
Print(os.listdir(../input))
nRowsRead =
Df1 = pd.read_csv(../input/Tweets.csv, delimiter=,, nrows =
nRowsRead)
Df1.dataframeName = Tweets.csv
nRow, nCol = df1.shape
print(fThere are {nRow} rows and {nCol} columns)
```

O/p

Bagging:

I/N:

Accuracy = accuracy_score(y_test, y_pred)

Report = classification_report(y_test, y_pred)

Print(fAccuracy: {accuracy})

Print(Classification Report:\n, report)

Boosting:

I/N:

Y_pred = ada_boost_classifier.predict(X_test_tfidf)

Accuracy = accuracy_score(y_test, y_pred)

Print(fAccuracy: {accuracy:.2f})

Recurrent neural network:

I/N:

Model.compile(optimizer=adam,
loss=binary_crossentropy,

metrics=[accuracy])

Model.fit(X_train, y_train, epochs=5, batch_size=64,

validation_data=(X_test, y_test))

Loss, accuracy = model.evaluate(X_test, y_test)

Print(fLoss: {loss}, Accuracy: {accuracy})

Print(fText: {text}\nSentiment: {Positive if sentiment > 0.5
else

Negative})

Convolution neural network:

I/N

Texts = [This is a positive review., Negative sentiment in this one., …]

Model = keras.Sequential

Test_loss, test_acc = model.evaluate(x_test, y_test)

Print(Test accuracy:, test_acc)

BERT:

I/N:

Model_name = bert-base-uncased # You can choose different BERT

variants

Tokenizer = BertTokenizer.from_pretrained(Tweet)

Model = BertForSequenceClassification.from_pretrained(Tweet)

With torch.no_grad():

Outputs = model(**inputs)

Sentiment_labels = {0: Negative, 1: Neutral, 2: Positive}

Sentiment = sentiment_labels[predicted_label]

Text_to_analyze = Positive

Result = analyze_sentiment(text_to_analyze)

Print(fSentiment: {result})

RoBERTa:

I/N:

Tokenizer = RobertaTokenizer.from_pretrained(Tweet)

Model = RobertaForSequenceClassification.from_pretrained(Tweet)

Logits = outputs.logits

Sentiment_labels = [Negative, Neutral, Positive]

Sentiment = sentiment_labels[predicted_class]

Return sentiment, logits.tolist()

Text = Positive

Sentiment, scores = analyze_sentiment(Positive)

Print(fSentiment: {sentiment})

Print(fSentiment Scores: {scores})

GPT-2:

I/N:

Model_name = gpt2 # You can also specify other variants like gpt2-

medium, gpt2-large, etc.

Tokenizer = GPT2Tokenizer.from_pretrained(Tweet)

Model = GPT2LMHeadModel.from_pretrained(Tweet)

Input_ids = tokenizer.encode(prompt, return_tensors=pt)

Generated_text = tokenizer.decode(output[0],

skip_special_tokens=True)

Print(generated_text)

From mpl_toolkits.mplot3d import Axes3D

From sklearn.preprocessing import StandardScaler

Import matplotlib.pyplot as plt

Import numpy as np

Import os

Import pandas as pd

Print(os.listdir(../input))

nRowsRead =

Df1 = pd.read_csv(../input/Tweets.csv, delimiter=„ nrows = nRowsRead)

Df1.dataframeName = Tweets.csv

nRow, nCol = df1.shape

print(fThere are {nRow} rows and {nCol} columns)

O/p

Tweet_id Airline_sentiment AS_confidence Negative_reason airline

1 5787676523440432 neutral 1.0000 NaN Virgin America

2 5766756565436587 positive 0.7843 Bad Flight VirginAmerica

3

.

.

577653245768987 negative 0.9879 Cant tell

VirginAme