NLP With Tweet Emotions

Submitted by:
Lohith.A -20691A3128
Department -AI & DS

Mentor:
Mr. k. DURGACHARAN

Contents:

About Internship

Google

COURSE CERTIFICATE

Sep 28, 2022

AREDDULA LOHITH

has successfully completed

Crash Course on Python

an online non-credit course authorized by Google and offered through Coursera

EDUCATION FOR EVERYONE coursera COURSE CERTIFICATE

Google

Google

Verify at:
https://coursera.org/verify/DMRQWVKVLJF9

Coursera has confirmed the identity of this individual and their participation in the course.

Abstract:

- Emotion detection (ED) is a branch of sentiment analysis that deals with the extraction and analysis of emotions.
- The evolution of Web 2.0 has put text mining and analysis at the frontiers of organizational success.
- It helps service providers provide tailor-made services to their customers.
- Numerous studies are being carried out in the area of text mining and analysis due to the ease in sourcing for data and the vast benefits its deliverable offers.
- This article surveys the concept of ED from texts and highlights the main approaches adopted by researchers in the design of text-based ED systems.

Abstract:

- The article further discusses some recent state-of-the-art proposals in the field.
- The proposals are discussed in relation to their major contributions, approaches employed, datasets used, results obtained, strengths, and their weaknesses. Also, emotion-labeled data sources are presented to provide neophytes with eligible text datasets for ED.
- Finally, the article presents some open issues and future research direction for text-based ED.

Introduction:

- Since the birth of Artificial Intelligence in 1950 and its rebirth in the 20th century, it has contributed significantly to providing effective solutions to major human and societal problems under various fields including natural language processing (NLP), which employs computational and linguistics techniques to aid computers understand and sometimes generate human languages in the form of texts and speech/voice.

- The field of ED has also be applied in applications such as emotion retrieval from suicide notes,capturing emotions in multimedia tagging,detecting insulting sentences in conversations,and so on.

- However, whereas detecting emotions from voice/speech, images, and other multimodal methods have an exhaustive knowledge base, there exists great paucity in research for texts.

- This is because unlike multimodal methods, texts may not portray peculiar cues to emotions. Also, the hurdle of detecting emotions from short texts, emojis, and grammatical errors could be back-breaking coupled with the continuous evolution of new words as a result of language dynamics.

Text-Based ED:

- This section presents a brief introduction to emotion models that fundamentally define the detection of emotions. Some datasets are also highlighted for researchers needing data for their research in the field.

Emotion models:

- Emotion models are the foundations of ED systems; they define how emotions are represented. The models assume that emotions exist in various states thus the need to distinguish between the various emotion states.
- When undertaking any ED related activity, it is imperative to initially define the model of emotion for use.
- In Reference, various forms of representing emotions are identified; however, of utmost importance to this article is the discrete and dimensional emotion models (DEMs and DiEMs, respectively).

Emotion models:

1. Discrete emotion models (DEMs): The discrete model of emotions involves placing emotions into distinct classes or categories. Prominent among them include.

2. Dimensional emotion models (DiEMs): The dimensional model presupposes that emotions are not independent and that there exists a relation between them hence the need to place them in a spatial space.

ED from text:

- The emotions that prompt individuals to pen down certain words at particular times are what text-based ED is concerned about.

- The algorithm used as well as the results obtained after the study is presented in this section to highlight the paucity in text-based emotion research studies as hypothesized

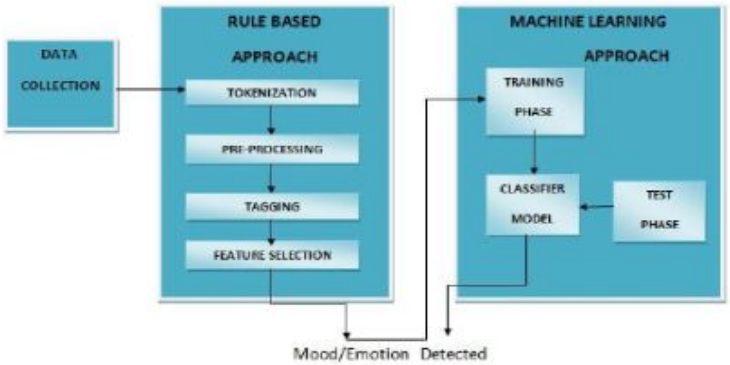
Requirements:

S.no	Software	Required
1	OS(Operating Systems)	Windows 10/11
2	Programming Language	Python
3	IDE(Integrated Development Environment)	Visual Studio Code,Jupyter Notebook.
4	APIs	Numpy,Pandas,Sklearn, Matlib,Seaborn, Pywhatkit.

S.no	Hardware	Required
1	RAM	8GB/16GB for faster output
2	Hard Disk	1TB
3	MicroProcessor	2.0GHZ
4	Micro Processor Type	Core i5/Core i7

Proposed system:

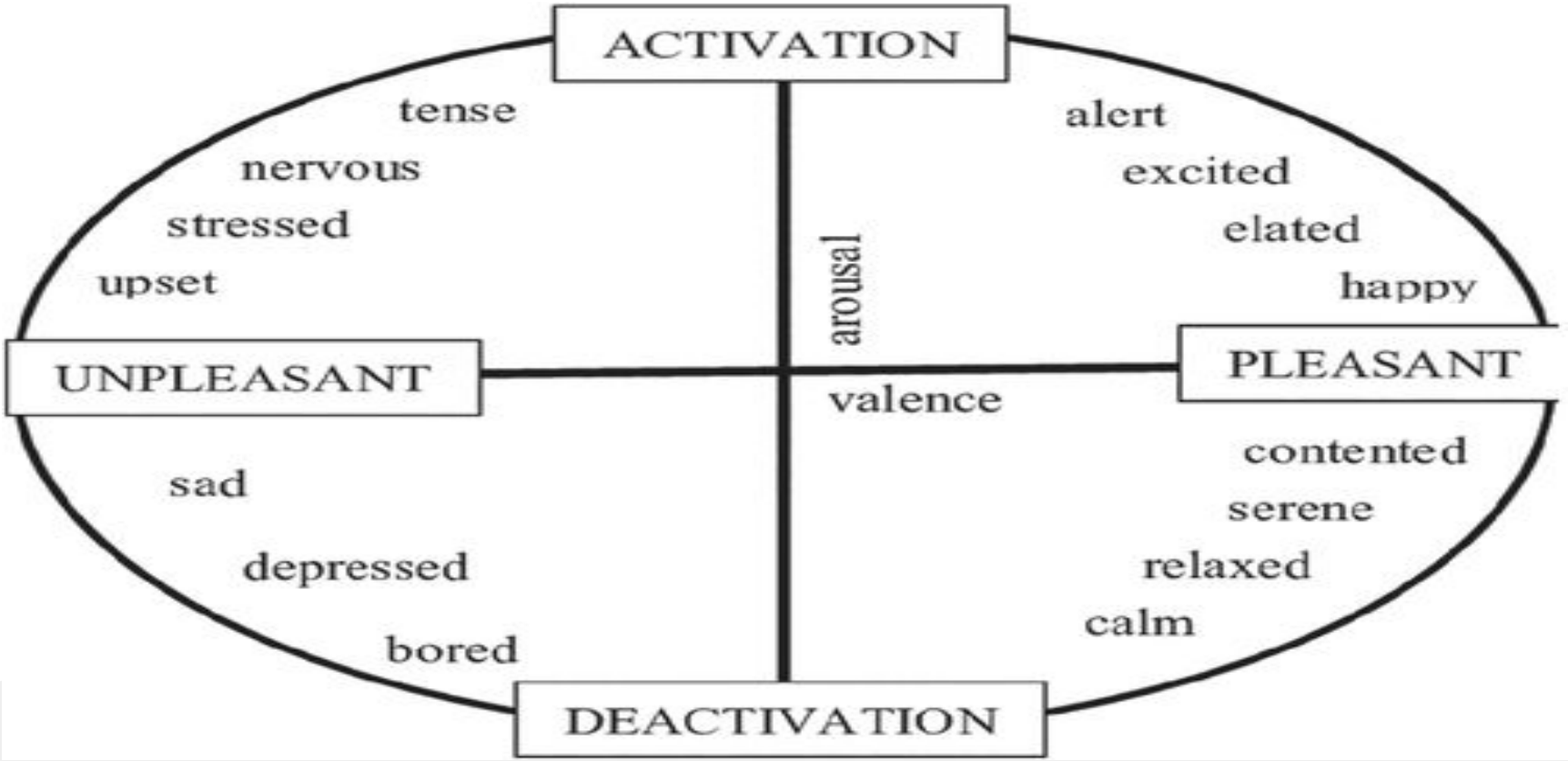
we have proposed a knowledge Based approach and machine learning approach for detecting the emotion or mood of the tweet.



The system is based on 2 approaches:

1. Rule Based Approach
2. Machine Learning Approach

Diagrams


ACTIVATION
tense
nervous
stressed
upset
alert
excited
elated
happy
arousal
UNPLEASANT
PLEASANT
valence
sad
depressed
bored
contented
serene
relaxed
calm
DEACTIVATION

SENTIMENT ANALYSIS

NEGATIVE

Totally dissatisfied with the service. Worst customer care ever.

NEUTRAL

Good Job but I will expect a lot more in future.

POSITIVE

Brilliant effort guys! Loved Your Work.

Nlp with tweet emotions detection of python:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Data Prepartion:

Data Prepartion

```
### load pkgs
import pandas as pd
import numpy as np
```

```
df=pd.read_csv("/content/drive/MyDrive/Dataset/tweet_emotions.csv")
# Dataset is now stored in a Pandas Dataframe
```

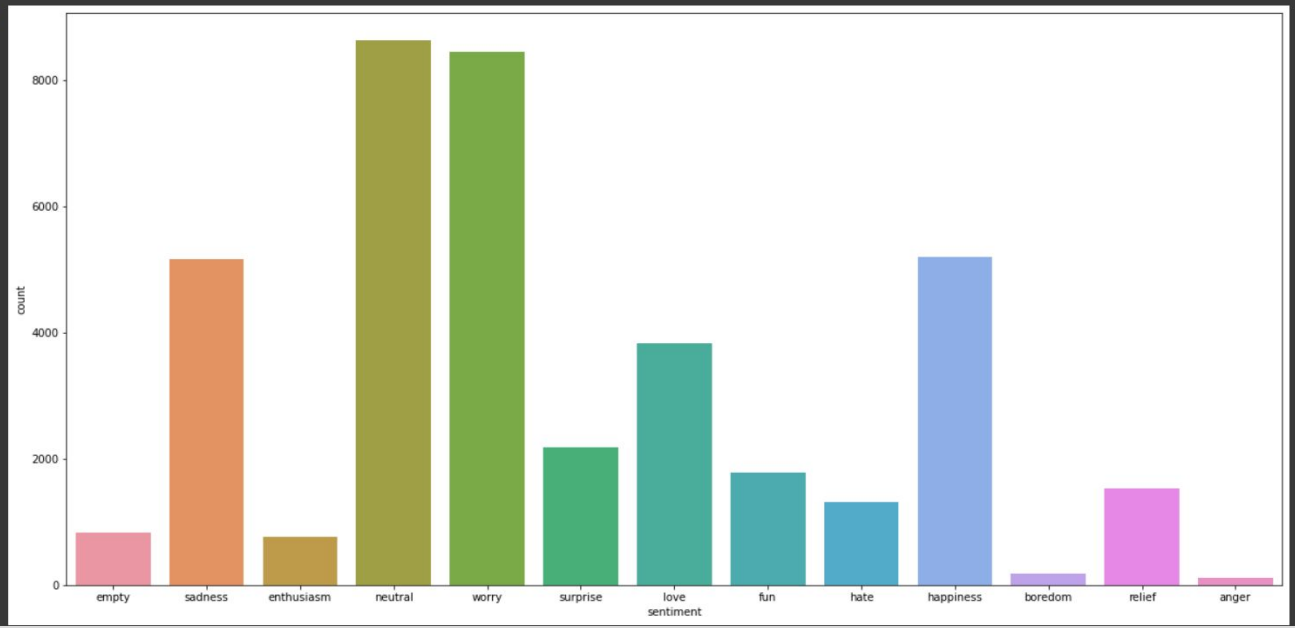
```
# load data viz pkgs
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#value counts of emotions
df['sentiment'].value_counts()
```

```
#value counts of emotions
df['sentiment'].value_counts()
```

```
neutral       8638
worry         8459
happiness     5209
sadness       5165
love          3842
surprise      2187
fun           1776
relief        1526
hate          1323
empty          827
enthusiasm     759
boredom        179
anger          110
Name: sentiment, dtype: int64
```

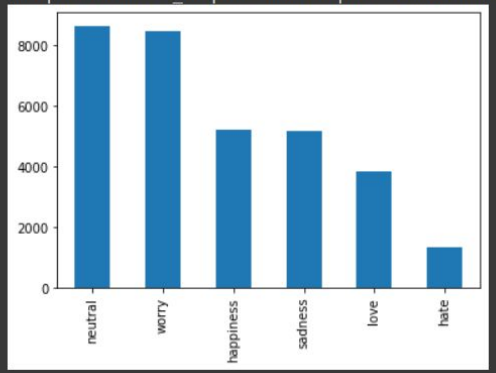
```
plt.figure(figsize=(20,10))
sns.countplot(x='sentiment',data=df)
plt.show()
```



```
df=df[ (df['sentiment']=='neutral') | (df['sentiment']=='worry') | (df['sentiment']=='happiness') | (df['sentiment']=='sadness') | (df['sentiment']=='love')| (df['sentiment']=='hate')]
```

```
#value counts of emotions
df['sentiment'].value_counts().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fada4c45910>
```




```
training["sentiment"].value_counts()
```

```
neutral      8638
worry        8459
happiness    5209
sadness      5165
love         3842
hate         1323
Name: sentiment, dtype: int64
```

```
total = training.isnull().sum().sort_values(ascending=False)
percent = (training.isnull().sum()/training.isnull().count()).sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
missing_data.head(20)
```

	Total	Percent
tweet_id	0	0.0
sentiment	0	0.0
content	0	0.0

```
training.sentiment = pd.Categorical(pd.factorize(training.sentiment)[0])
```

Training:

Training

TfidfVectorizer Count Vectorizer give number of frequency with respect to index of vocabulary where as tf-idf consider overall documents of weight of words.

```
[ ]  from sklearn.feature_extraction.text import TfidfVectorizer
```

```
[ ]  tfidf_vect = TfidfVectorizer()

     text1 = tfidf_vect.fit_transform(training["content"])
```

```
[ ]  y = training["sentiment"]
```

```
[ ]  from sklearn.model_selection import train_test_split

     X_train, X_test,y_train,y_test = train_test_split(text1, y, test_size=0.3, random_state=123)
```

```
[ ]  from sklearn.svm import SVC
     from sklearn.metrics import classification_report
```

```
[ ]  print(X_train.shape)
```

```
(22845, 35149)
```

```
model = SVC()
model.fit(text1,y)
```

```
SVC()
```

```
[ ]  pred_svm = model.predict(X_test)
     print(classification_report(pred_svm, y_test))
```

```
              precision    recall  f1-score   support

           0       0.81      0.93      0.87      1372
           1       0.94      0.84      0.89      2897
           2       0.93      0.84      0.89      2838
           3       0.78      0.87      0.82       981
           4       0.44      0.93      0.60       184
           5       0.87      0.89      0.88      1519

    accuracy                           0.87      9791
   macro avg       0.79      0.89      0.82      9791
weighted avg       0.88      0.87      0.87      9791
```

```
[ ]  pred_svm
```

```
array([2, 5, 2, ..., 2, 0, 3])
```

Input:

```
[ ] ### load pkgs
    import pandas as pd
    import numpy as np
```

```
df=pd.read_csv("/content/drive/MyDrive/Dataset/tweet_emotions.csv")
# Dataset is now stored in a Pandas Dataframe
```

```
#value counts of emotions
df['sentiment'].value_counts()
```

```
neutral       8638
worry         8459
happiness     5209
sadness       5165
love          3842
surprise      2187
fun           1776
relief        1526
hate          1323
empty          827
enthusiasm     759
boredom        179
anger          110
Name: sentiment, dtype: int64
```

Output:

```
[ ] pred_svm = model.predict(X_test)
    print(classification_report(pred_svm, y_test))
```

```
              precision    recall  f1-score   support

           0       0.81      0.93      0.87      1372
           1       0.94      0.84      0.89      2897
           2       0.93      0.84      0.89      2838
           3       0.78      0.87      0.82       981
           4       0.44      0.93      0.60       184
           5       0.87      0.89      0.88      1519

    accuracy                           0.87      9791
   macro avg       0.79      0.89      0.82      9791
weighted avg       0.88      0.87      0.87      9791
```

Reference:

- Shadi Shaheen, Wassim El-Hajj et.al, “Emotion Recognition from Text Based on Automatically Generated Rules”, in 2014 IEEE International Conference on Data Mining Workshop.
- Hema Krishnan, M. Sudheep Elayidom, T.Santhanakrishnan, “ Sentiment Analysis of tweets for inferring popularity of mobile phones”, in International Journal of Computer Applications, Jan 2017 edition


Thank You