SmartBridge AI Internship Project Report
Remote Summer Internship Program

Submitted By:
Geetha Manogna Goddu
RSIP Career Basic AI 052
Artificial Intelligence
geethamanogna9@gmail.com
Team: GG

Topic: Sentiment Analysis Of Twitter Data Using Deep Learning

Category: Deep Learning

Skills Required: Python,Python Web Frame Works,NLP

Project Description:

Twitter has grown in popularity during the past decades. It is now used by millions of users who share information about their daily life and their feelings. In order to automatically process and analyse these data, applications can rely on analysis methods such as sentiment analysis and topic modelling.Developing a program for sentiment analysis is an approach to be used to computationally measure customers' perceptions.

Solution :

The model takes the text as input, pre-processes the text and sends it to the neural network. The neural network classifies the text as 0 or 1 based on whether the text is negative or positive respectively. Using this the sentiment of the person who is sending the tweet can be understood and tweet can be analysed.

What is Sentiment Analysis?

Sentiment analysis is the interpretation and classification of emotions (positive, negative and neutral) within text data using text analysis techniques. Sentiment analysis allows businesses to identify customer sentiment toward products, brands or services in online conversations and feedback.

Why sentiment analysis?

Business: In marketing field companies use it to develop their strategies, to understand customers' feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some
products.
Politics: In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!
Public Actions: Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

Process for developing the Web-Application:

1. Data Collection
    1. Collect the Dataset
2. Text Pre-Processing
    1. Import the dataset
    2. Remove punctuations, numbers
    3. Convert each word into its Lower Case
    4. Stemming
    5. Splitting Data into Train and Test Set
    6. Importing the mode Building Libraries
    7. Initializing the model
    8. Adding Input Layer, Hidden Layer and Output Layer
    9. Configure the Learning Process
    10. Train and Test the model
    11. Optimize the model
    12. Save the model
    13. Predictions
3. Application Building
    1. Create a HTML File
    2. Build Python code

Literature Survey:

Sentiment analysis is an evolving field of study which involves the process of evaluating and distinguishing the opinions or emotions expressed in a given text. Twitter promotes unregulated communication by providing an easily accessible medium where millions of people tweet everyday to contribute their thoughts and viewpoints to the world. This paper presents a review on the techniques of Sentiment Analysis on the Twitter Data. With this paper, we present a brief review of all the work done on twitter sentiment analysis so far and elaborate the models and their methodologies used. We have surveyed all the papers published in this field and focused on the recent approach so as to facilitate the development of promising avenues of future projects and research. Index Terms sentiment analysis, twitter data.

Existing Problem:

- Existing system takes a stored dataset into consideration.
- It fails to determine the impact the results might or will have in the respective field.
- Existing system does not allow the retrieval of data bases in the query entered by the user.
- Existing system does not provide accurate feature selection.

Proposed Solution:

- Proposed system gives you the impact the results and statistics will have on the respective fields.
- Proposed system allows retrieval of data based on the query entered by the user.
- Proposed system will provide accurate feature selection.

Theoretical analysis:

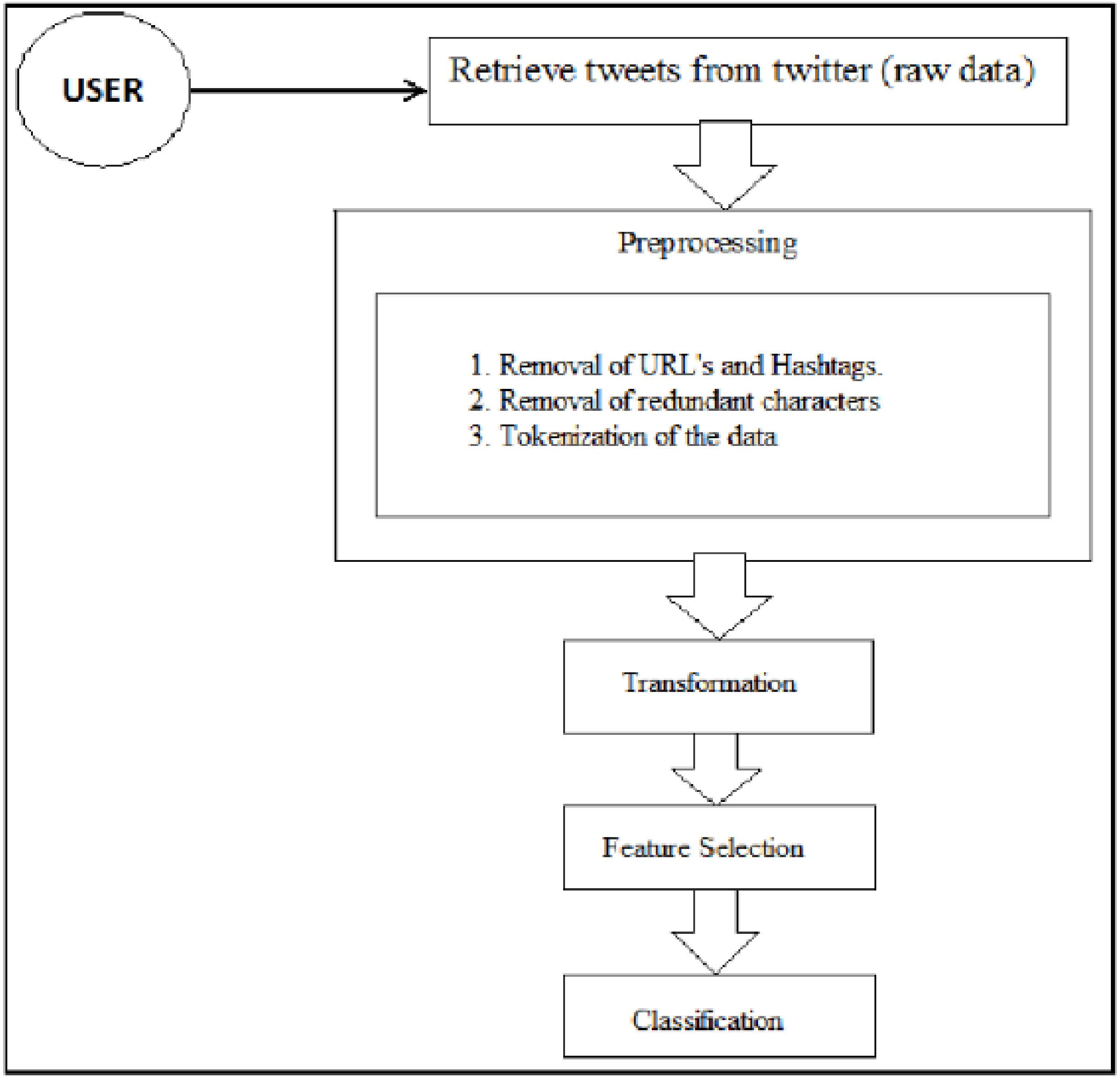

USER
Retrieve tweets from twitter (raw data)
Preprocessing
1. Removal of URL's and Hashtags.
2. Removal of redundant characters
3. Tokenization of the data
Transformation
Feature Selection
Classification

Fig. 1: System architecture

Advantages and Dis-Advantages:
Advantages and Disadvantages of Sentiment Analysis:

Advantages:

- The use of this information can be applied to make wiser decisions related to the use of resources, to make improvements in organizations.
- Tracking people's feelings on products, services and events, which allow enterprise managers to have knowledge and parameters to decision-making.

Disadvantages:

- For they are usually coupled with hashtags, emoticons and links, creating difficulties in determining the expressed sentiment.

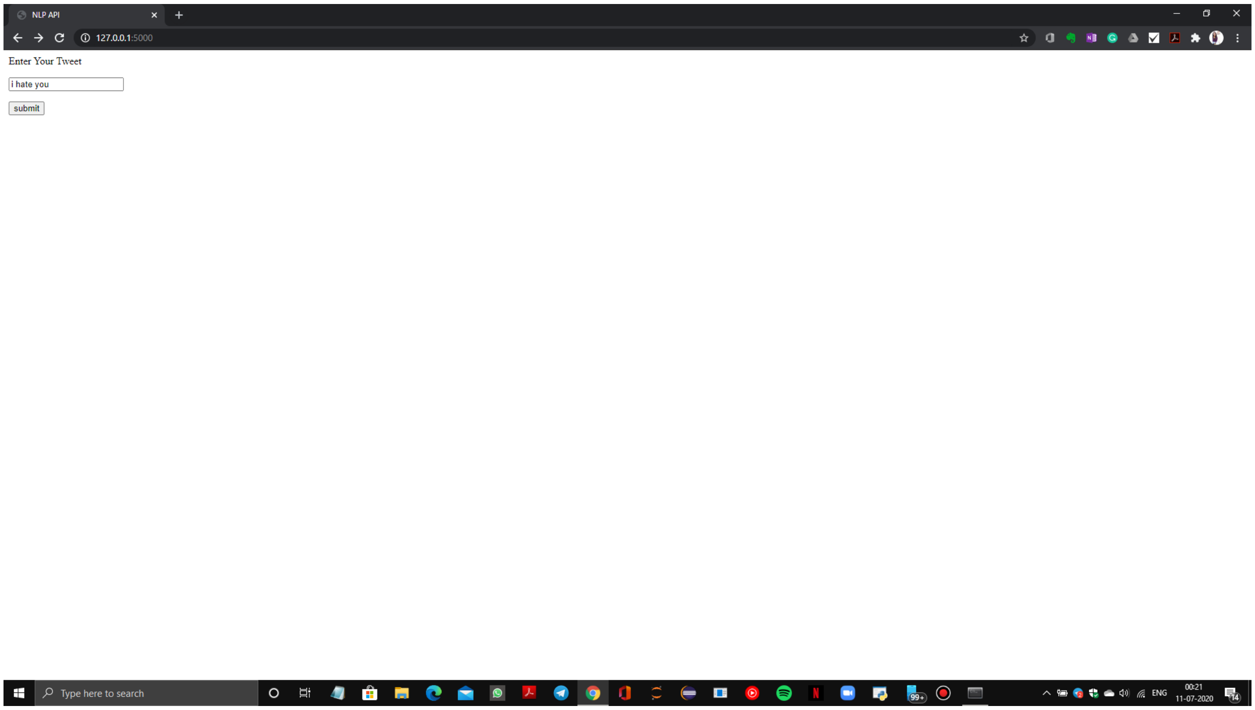
Applications :

- Social media monitoring
- People analytics and voice of employees
- Voice of customer & Customer Experience Management
- Regulatory Compliance

Conclusion:

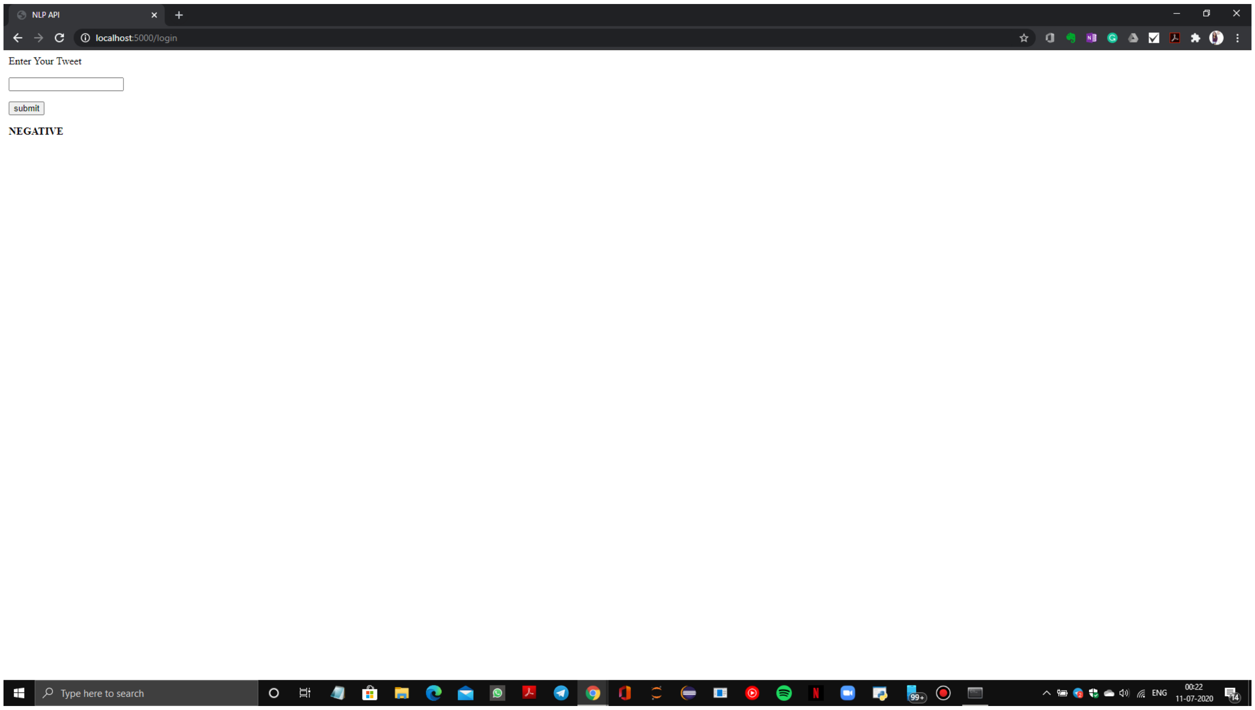
Sentiment analysis refers to the management of sentiments, opinions, and subjective text. The demand of sentiment analysis is raised due to the requirement of analysing and structuring hidden information, extracted from social media in form of unstructured data. The sentiment analysis is being implementing through deep learning techniques. Deep learning consists of numerous effective and popular models, these models are used to solve the variety of problems effectively. Different studies have been discussed in this review to provide a deep knowledge of the successful growing of deep learning applications in the field of sentiment analysis. Numerous problems have been resolved by having high accuracy of both fields of sentiment analysis and deep learning.

Output:



Enter Your Tweet

i hate you



Enter Your Tweet

NEGATIVE

<u>Code:</u>

app4.py:

```
from tensorflow.keras.models import load_model
import numpy as np
import tensorflow as tf
from tensorflow.python.framework import ops
graph = ops.get_default_graph()


from flask import Flask, request, render_template
app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/login', methods=['POST'])
def login():
    name = request.form['name']
    s = str(name)
    with graph.as_default():
        from tensorflow.keras.models import load_model
        from tensorflow.keras.preprocessing.text import Tokenizer
        from tensorflow.keras.preprocessing.sequence import pad_sequences
        import numpy as np
        model = load_model('nlp2.h5')
        tokenizer = Tokenizer(num_words=3000)
        x_test = pad_sequences(tokenizer.texts_to_sequences([s]), maxlen=101)
        score = model.predict([x_test])[0]
        if np.argmax(score) == 2:
            a = "POSITIVE"
        elif np.argmax(score) == 0:
            a = "NEGATIVE"
        elif np.argmax(score) == 1:
            a = "NEGATIVE"

    print(s)
    return render_template('index.html', abc = a)

def prediction(text):
    tokenizer = Tokenizer(num_words=3000)
    x_test = pad_sequences(tokenizer.texts_to_sequences([text]),
maxlen=101)
    score = model.predict([x_test])[0]
    if np.argmax(score) == 2:
        a = "POSITIVE"
    elif np.argmax(score) == 0:
        a = "NEGATIVE"
    elif np.argmax(score) == 1:
        a = "NEUTRAL"
    return print(a)


if __name__ == '__main__':
    app.run(debug = True)
```

sentiment_analysis.py:

```
# Importing libraries
import numpy as np
import pandas as pd

# Importing the Dataset
dataset = pd.read_csv('train.tsv', sep="\t")
```

```
#dataset2 = dataset

import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
c=[]

def remove_pat(inpt, pat): #for removing the @names and links in
the comments
    n = re.findall(pat, inpt)
    for i in n:
        inpt = re.sub(i, ' ', inpt)
    return inpt

for i in range(0, 7589):
    rev = dataset['tweet'][i]
    rev = remove_pat(rev, '@[\w]*') #user name
    rev = rev.replace('(', '') #bracket one
    rev = rev.replace(')', '') #bracket two
    rev = remove_pat(rev, r'https?://[A-Za-z0-9./]+')
    #rev = remove_pat(rev, r"http\S+") #links
    rev = re.sub('[^a-zA-Z]', ' ',rev) #removing special characters
    rev = rev.lower() #lower case
    rev = rev.split()
    rev = [word for word in rev if not word in
set(stopwords.words('english'))] #getting rid of stopwords
    ps = PorterStemmer()
    rev = [ps.stem(word) for word in rev if not word in
set(stopwords.words('english'))] #Stemming
    rev = ' '.join(rev)
    c.append(rev)

#dataset2['tweet'] = c

# Tokenizing for words into sequence
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import
pad_sequences
tokenizer = Tokenizer(num_words=3000)
tokenizer.fit_on_texts(c)
x = tokenizer.texts_to_sequences(c)
x = pad_sequences(x)

# OneHotEncoding the target
y = pd.get_dummies(dataset['label']).values

# Train Test Split
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.2,
random_state=0)

# Model building
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import SpatialDropout1D
model = Sequential()
model.add(Embedding(3000, 200,input_length = 101))
model.add(SpatialDropout1D(0.25))
model.add(LSTM(150, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(3,activation='softmax'))
model.compile(loss = 'categorical_crossentropy',
optimizer='adam',metrics = ['accuracy'])

model.fit(x_train, y_train, epochs = 4, batch_size = 32)

y_pred = model.predict(x_test)

# Prediction
def prediction(text):
    x_test = pad_sequences(tokenizer.texts_to_sequences([text]),
maxlen=101)
    score = model.predict([x_test])[0]
    if np.argmax(score) == 2:
        a = "POSITIVE"
    elif np.argmax(score) == 0:
        a = "NEGATIVE"
    elif np.argmax(score) == 1:
        a = "NEUTRAL"
    return print(a)

prediction('I hate mondays.')

# Saving
model.save('nlp2.h5')
```

index.html:

```
<html>
<head>
<meta charset="UTF-8">
<title>NLP API</title>
<link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
```

```
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condense
d:300' rel='stylesheet' type='text/css'>
</head>
<body>
<form action = "http://localhost:5000/login", method = "post">
<p>Enter Your Tweet </p>
<p> <input type="text" name="name" /></p>
<p> <input type="Submit" value="submit"/><p>
</form>
<b> {{abc}}</b>
<body>
<html>
```