# EMOTION DETECTION FROM TEXT

A Project Report in partial fulfilment of the degree

**Bachelor of Technology**

in

**Computer Science & Engineering / Electronics & Communication Engineering**

By


19K41A0450 N. Bhavith

19K41A0539 G. Anjali Kiran

19K41A0540 G. Sai srujan

Under the guidance of

**Mr. D. RAMESH**

Assistant Professor School of CS & AI SRU


**Submitted to**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

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

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

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## CERTIFICATE

This is to certify that the Project Report entitled "EMOTION DETECTION FROM TEXT" is a record of bonafide work carried out by the student(s) Anjali kiran, Sai srujan, Bhavith bearing Roll No(s) 19K41A0539, 19K41A0540, 19K41A0450 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

Artificial Intelligence has been used to interpret data to make decisions and understand the emotions of humans. Sentiment analysis is a technique for determining people's attitudes, feelings, and emotions about a certain purpose, such as people, or organizations. Since the emergence of the internet, individuals have used text messaging as a source to communicate with each other. As a result, in today's virtual community, robots must interpret emotions in views, comments, and textual exchanges to give emotionally aware replies to users. Emotion detection is a subtype of sentiment analysis in that it predicts a specific emotion rather than just reporting good, negative, or neutral. Many researchers have previously worked on voice and facial expressions for emotion identification in recent years. However, detecting emotions in the text is a time-consuming operation. Emotions are easily recognized by humans, but the true issue emerges with technology. To discern emotions in text, machines require an accurate algorithm. This study uses a data set from open-source Kaggle. Our algorithm, unlike previous research, can detect emotions from text with an at most accuracy of over 86%.

# Table of contents

# 1. INTRODUCTION

Artificial Intelligence (AI) is the 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). Figure 1 shows the sub domain of artificial intelligence. Thinking AI is designed 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. Emotion is a strong sensation caused by one's surroundings, mood, or interactions with others. Emotions are really complicated. Emotions, according to some beliefs, are a state of feeling that causes physical and psychological changes that impact our behavior.

The relevance of emotion recognition from text is rapidly increasing with the Internet. As a result, technologists, and political analysts have emphasized and wish to use this sector in all aspects of decision-making to better enterprises, reputations, and so on. The Internet provides online social networking sites such as Facebook, YouTube, and Twitter. Speech, facial expressions, and text-based emotion can all be used to express emotions. Text messages are used to communicate. Humans can quickly perceive emotion, while machines have a greater deal of difficulty. To discern emotion in text, machines require an accurate algorithm. Text-based recognition is also beneficial to psychologists.

Emotions include joy, sadness, rage, surprise, hatred, fear, and so forth. The emotion-detecting system formally divided human feelings into six groups at the main level, which are Love, Anger, Joy, Fear, Sadness, and Surprise. Certain additional terms are also classified as secondary levels. Emotion identification from text is a relatively new classification job, and advances in the textual analysis have made emotion detection a recent subject in the field of Natural language processing. The subject of how to identify emotion from text remains unanswered. So, our major goal is to extract emotion from the text at the sentence level with the greatest accuracy possible.

## 2. LITERATURE REVIEW

Based on the Research, There are several techniques that are used to identify emotions in text few are:

**Keyword-based detection:** Emotions are detected based on the related set(s) of keywords found in the input text.

**Learning-based detection:** Emotions are detected based on previous training result with respect to specific statistic learning methods.

**Hybrid detection:** Emotions are detected based on the combination of detected keyword, learned patterns, and other supplementary information.

Several research have been conducted using various methodologies to identify emotions in text.

Mostafa Aref et al [1], the authors offered an electronic deep research model that categorizes tweets with numerous sorts of embedded phrases CNN is the core component of our approach, and terminology are based on it. The achieved accuracy is 78. The word embedding started on its own, and the highest accuracy for Word2Vec is 84.99c.

Seal et al [2] emotion identification was carried out using a keyword-based strategy that primarily targeted phrasal verbs. They preprocessed the ISEAR data, employed the keyword-based technique, and then used the results. They identified several phrasal verbs that ought to be connected to emotion categories but weren't, so they created their own database. They identified phrasal verbs and terms that were interchangeable with different emotions and classified them using their database. They did, however, surpass the researcher's preexisting problems, such as an inadequate list of emotion keywords and a disregard for word semantics in meaning, by achieving a significantly greater accuracy of 65%.

Santosh Kumar & Rajeev Kumar Gupta et al [3] proposed a model that the pipeline first received the text as an input, and it then transformed the text into a vector. The ML Classifier was trained using these vectors. Second, the pre trained word vector was used to extract the features, and the embedding matrix (18210, 300) served as the DL model's input layer. The DL model was used to train the padded vector. Used a hybrid model (deep learning + machine learning) SVM, CNN, BI-GRU and found an accuracy over 80.11%.

Kashif khan & Sher Hayat et al [4] Conducted experiments on the text emotion dataset to test the efficiency of the proposed method for social emotion detection. It includes thirteen reader feelings. Preprocessing begins by breaking each text into sentences using sentence boundary recognition, followed by deleting stop terms, numbers, person names, punctuation, and words that occur in less than three documents. Naive bayes classifier, linear SVM, Logistic regression and Random forest are used and found greater accuracy of over 80% using random forest.

Jianpei Zhang, Jing Yang, Yong Wang, et al [5] and the current research suggests a hybrid controlled sentiment analysis method based on one of the two approaches. A training phrase is used to gather the classifier from a small collection of labelled data using the Lexicon-based technique for learning confidence parameters. An evaluation set in Naive Bayes is then used to create a classifier for feelings. The classification for lexical sensory polarity and the experienced classifier come last. The Naive Bayes framework for investigating sensations is described.

Man Sherine Rady & Mostafa Aref et al [6] the authors proposed an electronic profound model of study that classifies tweets with various types of embedded words. CNN is the key part of their model, and terms are rooted in the central facets. The accuracy reached is 78. The word embedding spontaneously begun and maximum precision is 84.99c for Word2Vec.

Aditya Dave, Santosh Bharti et al [7] in order to provide a framework for the detailed analysis of sentiment that provides us with real-time sentiments for a case, business, or individual in the audience on the microblogging website Twitter, this article leverages existing sentiment analysis techniques. With the use of this technique, people may understand the perceived image of the scenario. This input might be utilised to improve the current situation if it is given in a timely manner.

# 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

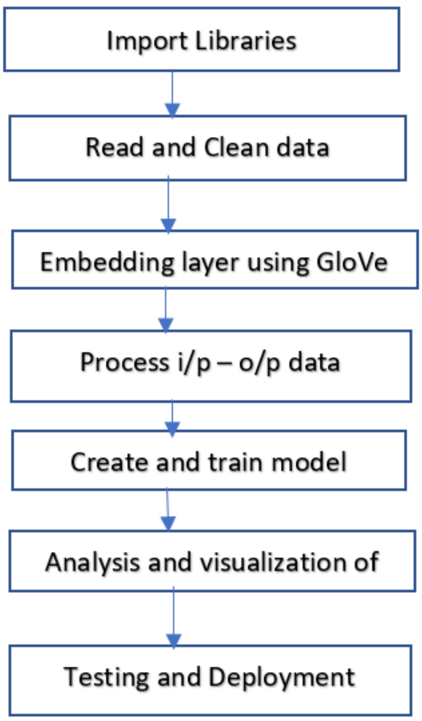


**Figure 1:** Flow chart

## 4. DATA SET

The Data set was obtained from open source Kaggle website. Data set contains 2 columns, Text, and Emotions. Emotions Column has various categories like sadness, anger, love, surprise, fear and happy. Data set consist of 5880 unique sentences with their corresponding emotion. 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.

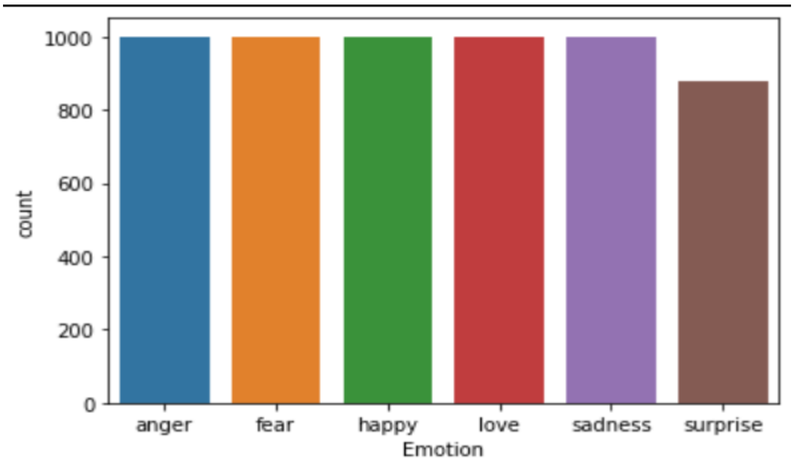


**Figure 2.** Emotion and count Graph

| | Text | Emotion |
|---|---|---|
| 0 | i didnt feel humiliated | sadness |
| 1 | i can go from feeling so hopeless to so damned... | sadness |
| 2 | im grabbing a minute to post i feel greedy wrong | anger |
| 3 | i am ever feeling nostalgic about the fireplac... | love |
| 4 | i am feeling grouchy | anger |

**Figure.3** Data Set Sample

# 5. DATA PRE-PROCESSING

```
df=df.dropna() #Drop rows with NA values
X=df.drop('Emotion',axis=1) #Input
y=df['Emotion'] #Output
```

```
messages=X.copy()
messages.reset_index(inplace=True) #Drop NA may cause inconsistency in index
```

```
nltk.download('stopwords')
ps = PorterStemmer()
corpus = []
for i in range(0, len(messages)):
    review = re.sub('[^a-zA-Z]', ' ', messages['Text'][i]) #Remove Special Characters
    review = review.lower() #Lower case
    review = review.split()
    review = [ps.stem(word) for word in review if not word in stopwords.words('english')] #Remove stopwords
    review = ' '.join(review)
    corpus.append(review)
```

**Figure.4** Data pre-processing

**Steps:**

- Drop rows with NA values
- Drop NA may cause inconsistency in index so reset indexes
- Remove special characters in text
- Convert into lower letters
- Remove stop words

**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.

# 6. METHODOLOGY

In our Proposed Model we are using GloVe embedding technique and BI-LSTM. Bidirectional long-short term memory is the process of making any neural network have the sequence information in both directions backwards (future to past) or forward (past to future). GloVe stands for Global Vectors for word representation, is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. The proposed model consists of data collection and data preprocessing. After preprocessing the data by removing stop words, the corpus is passed to glove embedding layer and converted to one hot representation and finally it is passed to CNN Model which uses BI-LSTM.

We created a NLP model by using GloVe embedding technique and BI-LSTM. Firstly, we have imported the required libraries and read Csv file. We then performed missing value treatment by dropping columns with NA values. Removed special Characters and Stop words and then added glove word embedding layer it aims to generate word embedding's by aggregating global word occurrences matrices from corpus. Performed One hot representation for input and initialized our model to sequential () and used a CNN architecture for BI-LSTM. Added 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. Added dropout layer. It is a regulation technique where randomly selected neurons are ignored during training and added bi directional LSTM layer. We added dense layer with relu activation function and later added dense layer with SoftMax activation and compile**.**

**Steps:**

1. Import libraries and read Csv file
2. Perform missing value treatment by dropping rows with NA values
3. Remove special Characters and Stop words
4. Add glove word embedding layer it aims to generate word embedding's by aggregating global word occurrences matrices from corpus.
5. Perform One hot representation for input
6. Initializing model to sequential()
7. 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.

8. Add dropout layer. It is a regulation technique where randomly selected neurons are ignored during training
9. Add bi directional LSTM layer
10. Adding dense layer with relu activation function
11. Adding dense layer with SoftMax activation and compile.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 100)         1309100

 dropout (Dropout)           (None, None, 100)         0

 bidirectional (Bidirectiona  (None, 128)              84480
 l)

 dropout_1 (Dropout)         (None, 128)               0

 dense (Dense)               (None, 64)                8256

 dropout_2 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 6)                 390

=================================================================
Total params: 1,402,226
Trainable params: 1,402,226
Non-trainable params: 0
```

**Figure.8.** Model Summary

## 7. RESULTS

```
#Accuracy score
print("Training accuracy: ",accuracy_score(y_train,y_pred_train))
print("Testing accuracy: ",accuracy_score(y_test,y_pred_test))
print("Validation accuracy: ",accuracy_score(y_val,y_pred_val))
```

```
Training accuracy:  0.9918444622442292
Testing accuracy:  0.8657968313140727
Validation accuracy:  0.8581828771112405
```

**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 40 epochs to train the data. We found the accuracy of our proposed model is around 86%.

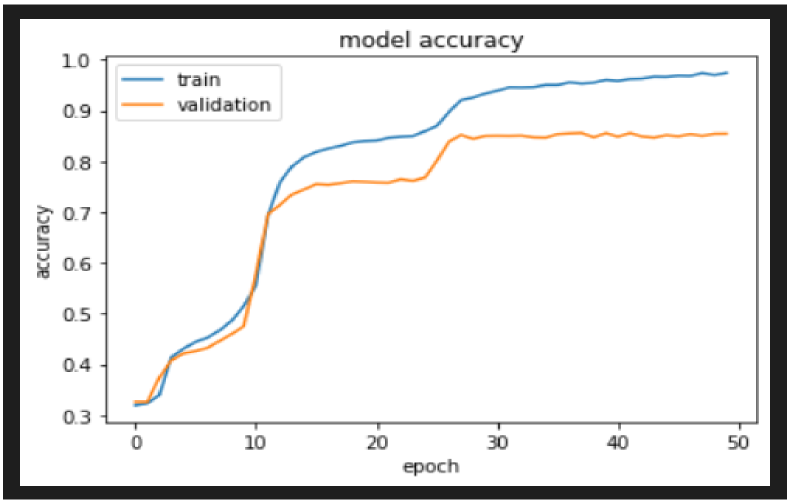


**Figure.10.** Model accuracy train and validation vs epochs

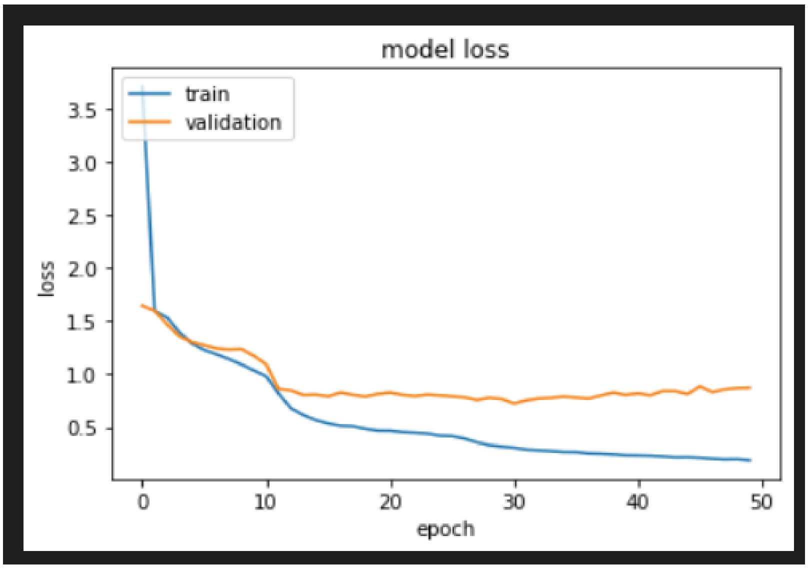


**Figure.11.** Model loss train and validation vs epochs

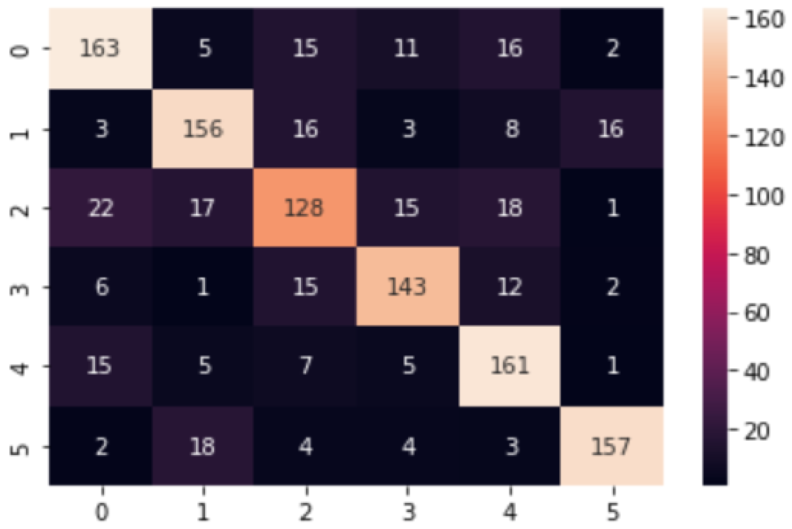

**Figure 12:** Confusion matrix

### Comparative result analysis:

```
predict_emotion('He was speechless when he found out he was accepted to this new job')
```

'happy'

```
predict_emotion("This is outrageous, how can you talk like that?")
```

'anger'

```python
predict_emotion('i feel like a miserable piece of garbage')
```

```
'sadness'
```

```python
predict_emotion('i just don t feel as impressed and as happy with things like i used to')
```

```
'surprise'
```

The above text consists of words like ‘impressed’ and ‘happy’ which are related to emotion ‘happy’ but this statement belongs to emotion ‘surprise’.

```python
predict_emotion('you look scary and strange')
```

```
'fear'
```

```python
predict_emotion('oops it is strange')
```

```
'surprise'
```

In the above texts we can observe ‘strange’ is the common word in both of them but, when same words are used in different scenarios their emotion and meaning changes.

```python
predict_emotion('i feel like some of you have pains and you cannot imagine becoming passionate about the group or the idea that is causing pain')
```

Python

```
'love'
```

```python
predict_emotion('it is painful to me')
```

Python

```
'sadness'
```

```python
predict_emotion('im passionate about my future')
```

```
'happy'
```

In the above texts we can observe the words ‘pain’ and ‘passionate’ are common but their emotion are different because they are used in different scenarios or sentences.

```
predict_emotion('i feel vulnerable and alone')
```

```
'fear'
```

```
predict_emotion('i feel alone')
```

```
'sadness'
```

In the above texts the word ‘alone’ is common but give different emotion for different sentences based on the scenario.

```
predict_emotion('im feeling good and happy')
```

```
'happy'
```

```
predict_emotion('He is not so happy today')
```

```
'sadness'
```

In the above texts the word ‘happy’ is common, generally it refers to emotion happy but in the second text we can observe it is ‘not happy’ which gives the output as ‘sad’.

```
predict_emotion('The tears came , and rapidly dissolved her angry expression to one of utter despai')
```

```
'sadness'
```

```
predict_emotion('Shelley sat down beside the patient and smiled at him , the cheerfulness coming from somewhere behind her deep despai')
```

```
'sadness'
```

```
predict_emotion('The look of deep sorrow on her face cut into him far more deeply than any aggresive words could have done')
```

```
'sadness'
```

In the above text we can observe the words angry, cheerfulness and aggressive doesn’t belong to the emotion ‘sadness’ but, the output is sad because these words in those sentences frame as a sad statement.

When we use methods based on synonyms they can’t recognize mixed emotions in the sentences. So, embedding techniques like GloVe is suitable to identify the sentences with more than one emotion because they calculate the weights based on the word to word co-occurrences and predict the accurate result.

## 8. CONCLUSION

Emotion detection is an important area of study in human-computer interaction. Researchers, have done enough work to recognize emotion from audio and visual data, but understanding emotions from textual data is still a new and active study topic. This paper discusses that the novelty of discovering relevant texts emotions plays a critical role in this work. In this study, we explore the model's ability to predict sentiments. In this article, we employed the BI-LSTM and GloVe embedding technique and created a machine learning model that predicts the emotion from the text. The accuracy obtained was over 86%.

## 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]Nourah Alswaidan, 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.), liitelligetit u+id Cloud Computing, Srnan Innovation, Systeriis and Technologies 153