Sentiment Analysis from Text

A Project Report in partial fulfillment of the degree

Bachelor of Technology

in

Electronics & Communication Engineering/Computer Science & Engineering

By

20K45A0405	**G.Tharun Teja**
20K45A0414	**G.Shiva**
20K45A0507	**M.Srujala**

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 JNTU, Accredited by NBA)Nov-2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled **"Sentiment Analysis from Text"** is a record of bonafide work carried out by the student(s) **G.Tharun Teja, G.Shiva, M.Srujala** bearing Roll No(s) 20K45A0405,20K45A0414,20K45A0507 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

Over the past ten years, sentiment analysis and opinion mining have seen significant growth. Customer reviews are one of the most important elements that determine the satisfaction of customers about the products. Also, it gives a comprehensive picture of products to business owners. This field of study makes an effort to ascertain, among other things, what people think, feel, and feel about something or someone. Natural language processing methods and machine learning algorithms are employed in this. The initial goal is to automatically identify a review's orientation and compare it to the judgment the article reviewer made. This would make it possible for researchers to categorize and contrast reviews across the board and more impartially support the overall evaluation of a scientific work. To assess reviews, a hybrid methodology is developed that combines a machine learning algorithm with methods from natural language processing.

Table of Contents

1. INTRODUCTION

The development of technology and the rise of E-commerce, internet buying, and the information age among internet users has greatly increased. Online purchasing has been a great success, thus there are many clients relocated or changed to alternative buying methods from purchases made online. Despite the benefits that online shopping and ecommerce provide, there are several drawbacks like the requirement for the customer to find measures to assure the quality of the goods since occasionally there are differences when a website's merchandise was shown in either an image or a photograph and the product's quality.

Reading or examining customer evaluations before making a purchase is one of the most crucial techniques for customers to verify the quality of products. Users can publish their thoughts and comments on the things they sell on numerous websites, forums, and blogs like Amazon. Reviews from consumers are very useful information for business and marketing owners as well as for the customers themselves. Business and marketing owners should examine customer reviews to determine the best-selling item, the item with the highest rating, the item with the most positive customer feedback, and the item with the most negative feedback.

User-generated content concerning users' opinions about goods or services is known as customer reviews. Additionally, a product or service receives comments or evaluations at a very fast rate. It is now exceedingly challenging for customers and business owners to get a full picture of opinions due to the expansion and rise in customer reviews. To address these issues, natural language processing (NLP) and data mining approaches were developed. There are several approaches that may be used to analyze customer evaluations, but a sentimental analysis is another useful one. Determine whether a piece of literature is favorable, negative, or neutral by performing a sentimental analysis.

A technique to sentiment analysis that combines technology learning NLP methods for polarity or weighted assignment themes, a portion of topics, and entities' sentiment ratings within a word or phrase. Aside from that, Sentimental analysis aids in the ability of data analysts in major businesses and enterprises to customer and public opinion measurement experiences, carry out precise market research, and keep an eye on the product's reputation and brand.

Sentiment analysis is frequently used to assess customer satisfaction by assessing opinions of people about various objects, including goods, services, themes, etc. It also goes by the moniker "opinion mining," and it has a broad problem domain. Natural Language

Processing (NLP), text analysis, and computational linguistics are all used in sentiment analysis to find and extract subjective information from source materials, such as determining if a review is good or negative. Subjectivity analysis, emotion analysis, review mining, and other terms for similar but slightly distinct activities include analysis of sentiments, opinion extraction, mining of sentiments and opinions, and subjectivity analysis.

Additionally, sentiment analysis can be performed at four levels: word-level, sentence-level, document-level, aspect-level, and concept-level (Appel et al.). The following text is examined to determine the polarity of the affective word, subjective sentence, and entire document, respectively, at the word-level, sentence-level, and document levels of sentiment analysis. Features or elements of a product or any other item that is being analyzed are chosen as targets for aspect-level sentiment analysis, and attitude toward these targets is discovered. Words in opinionated texts with similar meanings are recognized and taken into account as the same notion in concept-level sentiment analysis. Additionally, concept-based sentiment analysis provides new approaches to perform sentiment analysis and opinion mining that go beyond word-level sentiment analysis of text (Cambria et al.). Additionally, concept level methods use semantic networks like Concept Net to categorize text based on its semantics.

Text preprocessing was the original application for natural language processing. These days, NLP applications are widely used. In the domain of NLP, transfer learning is able to train a model on one task and then modify it to execute other NLP functions on other tasks. It is put into practice using pre-trained models. Instead of creating a new model from scratch for our objective, a pre-trained model can be used. This model can be improved a little to conserve computing time and resources. There are numerous pre-trained models that can complete various jobs.

2. LITERATURE REVIEW

[1] Laksono, R. A., Sungkono, K. R., Sarno, R., & Wahyuni, C. S. (2019). Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor using Naïve Bayes 2019 12th International Conference on Information & Communication Technology and System (ICTS). This study tries to classify Surabaya restaurant customer satisfaction using Naive Bayes. Data sampling is crawling by using WebHarvy Tools. The result from this research shows that these two methods get the customer response accurately and Naive Bayes method is more accurate than Text-Blob sentiment analysis with a different accuracy of 2.9%.

[2] Hassan et.al conducted research for improvement of sentiment analysis models for Twitter. They proposed a bootstrapping ensemble framework. They depend on comprising between flowing classifier: SVM, Logistic Regression, Naïve Bayes, Bayes Net, REP Tree, Random Tree, and RBF Neural Network. The result showed a bootstrapping ensemble framework produced a more balanced and accurate sentiment polarity representations in social media.

[3] Wang et.al conducted a study for sentiment classification. They depend on using five classifiers: Support Vector Machine, K Nearest Neighbor, Decision Tree, Maximum Entropy and Naive Bayes. Moreover, they compare the performance of three ensemble methods: Random Subspace, Boosting, and Bagging. The result showed the Random Subspace give the best performance.

[4] Fersini et.al Development a new ensemble method based on Bayesian Model Averaging. They compare five classifiers: Conditional Random Fields, Maximum Entropy, Support Vector Machines, Naïve Bayes and Dictionary. Moreover, they are based on using Bagging and Simple Voting as combination rules. The result showed the Bayesian Model Averaging gave the best performance.

[5] Chalothom et.al conducted research on Sentiment analysis on Twitter. They depend on using minority voting as a combination rule for the ensemble method. Moreover, they are based on the following classifiers: Stacking, SentiStrength Naive Bayes and Support Vector Machine.

[6] The localized Twitter opinion mining using sentiment analysis analyzes tweets about iPhone 6 using SentiWordNet, part of SNLP which is an open-source natural language processing tool developed by Stanford University.

[7] Twitter sentiment analysis training data contains a corpus of already classified tweets in terms of sentiment analysis training and testing where it contains more than 1,500,000 classified tweets, each row is marked as “1” for positive sentiment and “0” for negative sentiment.

[8] Sentiment analysis on travel reviews using three machine learning models namely, Naïve Bayes, SVM and character-based N-gram model has been performed in which SVM and N-gram approaches have better performance than Naïve Bayes. It has been observed that in the maximum number of cases SVM showcases best performance in comparison to other classification models.

[9] Fan et al develops decision support for vehicle defect discovery by making use of consumer feedback reviews (textual online discussion forums). The research text corpus comes from the 2010 snapshot of the United States Department of Transportation, National Highway Traffic Safety Administration (NHTSA), Office of Defect Investigations, vehicle safety complaint database. The min review is at least 50 words, and the max review is 8586 words, the review contains an average of 502 words.

[10] Kasthuriarachchy in his research compared the various fields of reviews which came from movies, DVD, phone and tweets, aimed to detect the different reflection in the semantic classification. In the data set, the average number of words per sentence is 17-22.2 and at least one sentence per review, even the average sentences per review in movies review are 35.8.

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** : Google Colab,Jupyter Notebook,vs code
- ✔ **Program Language** : Python

2.2 Flow chart

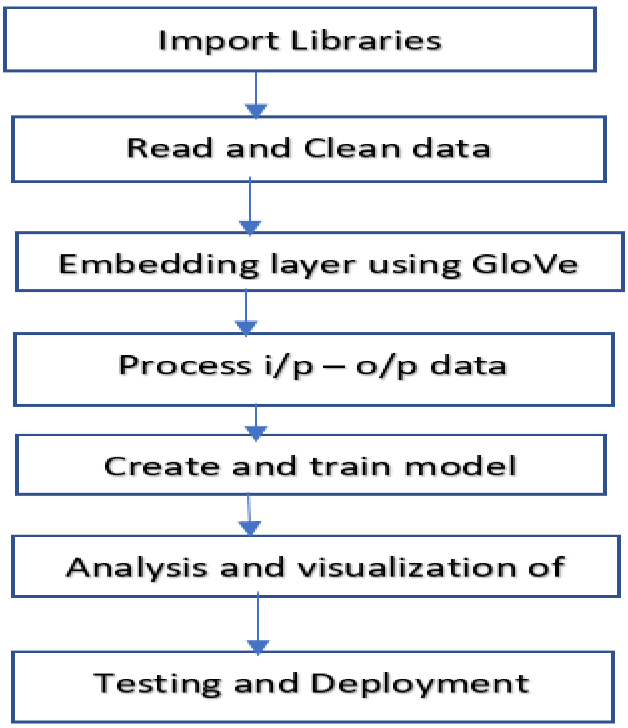


Fig 1: Work flow of the project.

3. DATASET:

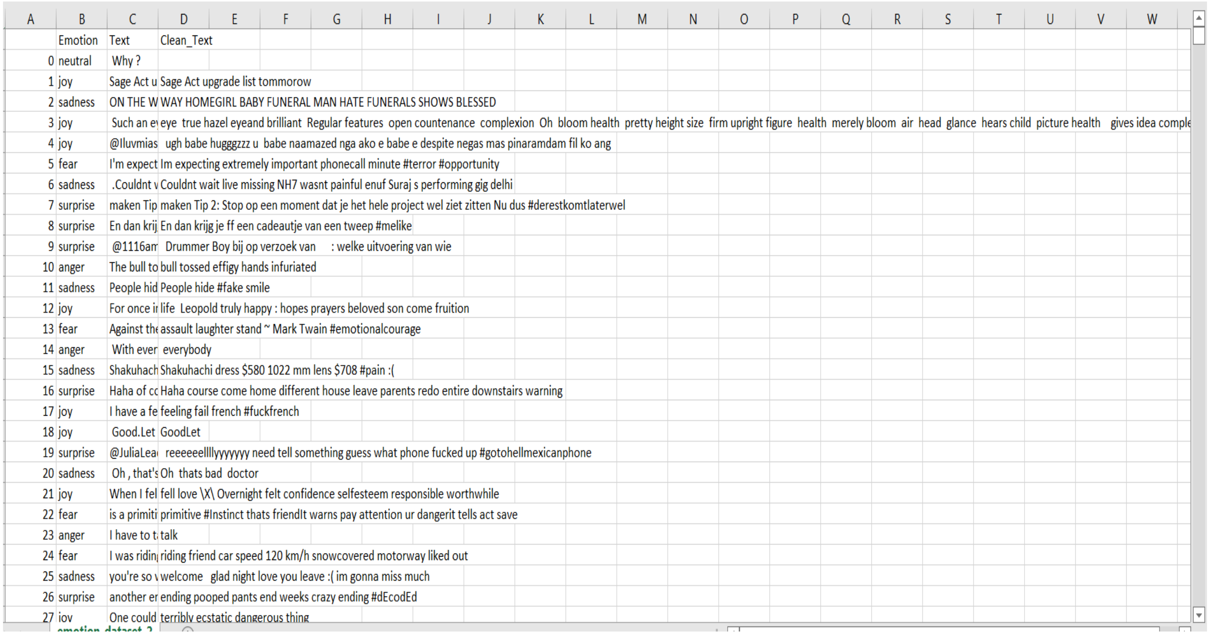
We have collected the data from known sources. Dataset consists of two columns – sentiment and text.

Input feature:

Text

Output feature:

Sentiment



	Emotion	Text	Clean_Text
0	neutral	Why ?	
1	joy	Sage Act u	Sage Act upgrade list tommorow
2	sadness	ON THE W	WAY HOMEGIRL BABY FUNERAL MAN HATE FUNERALS SHOWS BLESSED
3	joy	Such an ey	eye true hazel eyeand brilliant Regular features open countenance complexion Oh bloom health pretty height size firm upright figure health merely bloom air head glance hears child picture health gives idea comple
4	joy	@Iluvmias	ugh babe hugggzzz u babe naamazed nga ako e babe e despite negas mas pinaramdam fil ko ang
5	fear	I'm expect	Im expecting extremely important phonecall minute #terror #opportunity
6	sadness	.Couldnt v	Couldnt wait live missing NH7 wasnt painful enuf Suraj s performing gig delhi
7	surprise	maken Tip	maken Tip 2: Stop op een moment dat je het hele project wel ziet zitten Nu dus #derestkomtlaterwel
8	surprise	En dan krij	En dan krijg je ff een cadeautje van een tweep #melike
9	surprise	@1116am	Drummer Boy bij op verzoek van : welke uitvoering van wie
10	anger	The bull to	bull tossed effigy hands infuriated
11	sadness	People hid	People hide #fake smile
12	joy	For once ir	life Leopold truly happy : hopes prayers beloved son come fruition
13	fear	Against the	assault laughter stand ~ Mark Twain #emotionalcourage
14	anger	With ever	everybody
15	sadness	Shakuhach	Shakuhachi dress $580 1022 mm lens $708 #pain :(
16	surprise	Haha of cc	Haha course come home different house leave parents redo entire downstairs warning
17	joy	I have a fe	feeling fail french #fuckfrench
18	joy	Good.Let	GoodLet
19	surprise	@JuliaLea	reeeeeellllyyyyyyy need tell something guess what phone fucked up #gotohellmexicanphone
20	sadness	Oh , that's	Oh thats bad doctor
21	joy	When I fel	fell love \X\ Overnight felt confidence selfesteem responsible worthwhile
22	fear	is a primiti	primitive #Instinct thats friendIt warns pay attention ur dangerit tells act save
23	anger	I have to t	talk
24	fear	I was ridin	riding friend car speed 120 km/h snowcovered motorway liked out
25	sadness	you're so v	welcome glad night love you leave :(im gonna miss much
26	surprise	another er	ending pooped pants end weeks crazy ending #dEcodEd
27	joy	One could	terribly ecstatic dangerous thing

Fig 2: Dataset

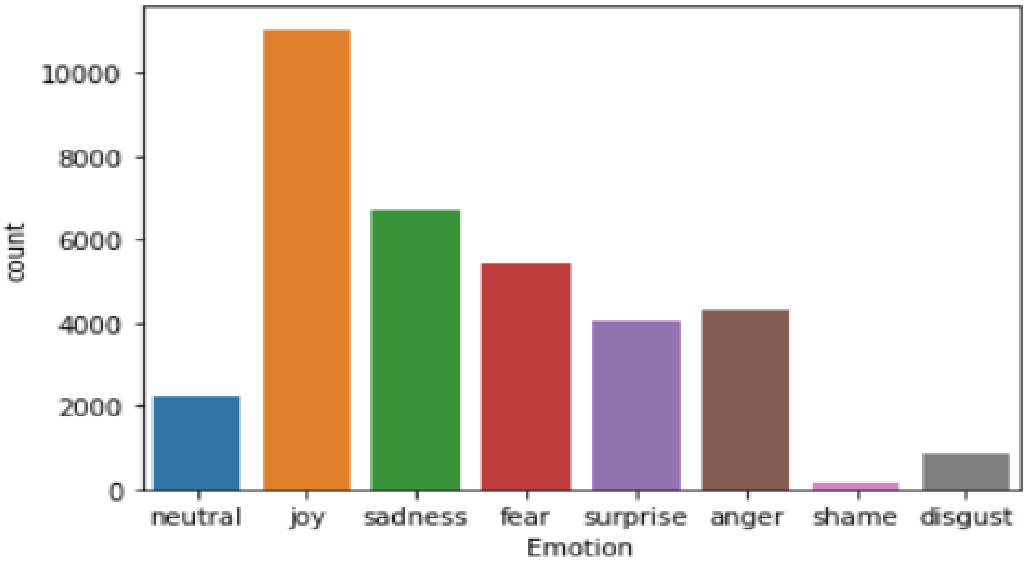


Fig 3: Bar graph of classes of dataset

4.DATA PREPROCESSING:

We have used BI-LSTM models to complete this process, where the main goal is to convert the data into vector, perform embedding on it and tokenize it. These are the steps taken for data Pre-processing.

Text Hammer

In any machine learning task, cleaning or pre-processing the data is as important as model building if not more and when it comes to unstructured data like text, this process is even more important. Objective of this kernel is to understand the various text pre-processing steps with code examples. Some of the common text pre-processing / cleaning steps are:

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

Code:

```
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)
```

Fig 4: Data Pre-processing

METHODOLOGY:

This section talks about the Glove and BILSTM models used for the project. 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 embeddings 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 columns 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.

3.2 Model Architecture

```
[ ] # Creating model
    model=Sequential()
    model.add(Embedding(voc_size, embedding_dim, weights=[embedding_matrix_vocab]))
    model.add(Dropout(0.25))
    model.add(Bidirectional(LSTM(64)))
    model.add(Dropout(0.25))
    model.add(Dense(64, activation='relu',kernel_regularizer=tf.keras.regularizers.l1(0.01))) #L1 regularization
    model.add(Dropout(0.25))
    model.add(Dense(8,activation='softmax'))
    model.compile(loss='sparse_categorical_crossentropy',optimizer= tf.keras.optimizers.Adam(learning_rate=0.001),metrics=['accuracy'])
    model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 100)         2858600

 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, 8)                 520

=================================================================
Total params: 2,951,856
Trainable params: 2,951,856
Non-trainable params: 0
_________________________________________________________________
```

Fig 5: Model parameters

6.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:  0.7370445515259109
     Testing accuracy:  0.5625808305791062
     Validation accuracy:  0.5520958083832336
```

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 75 epochs to train the data. We found the accuracy of our proposed model is around 60%.

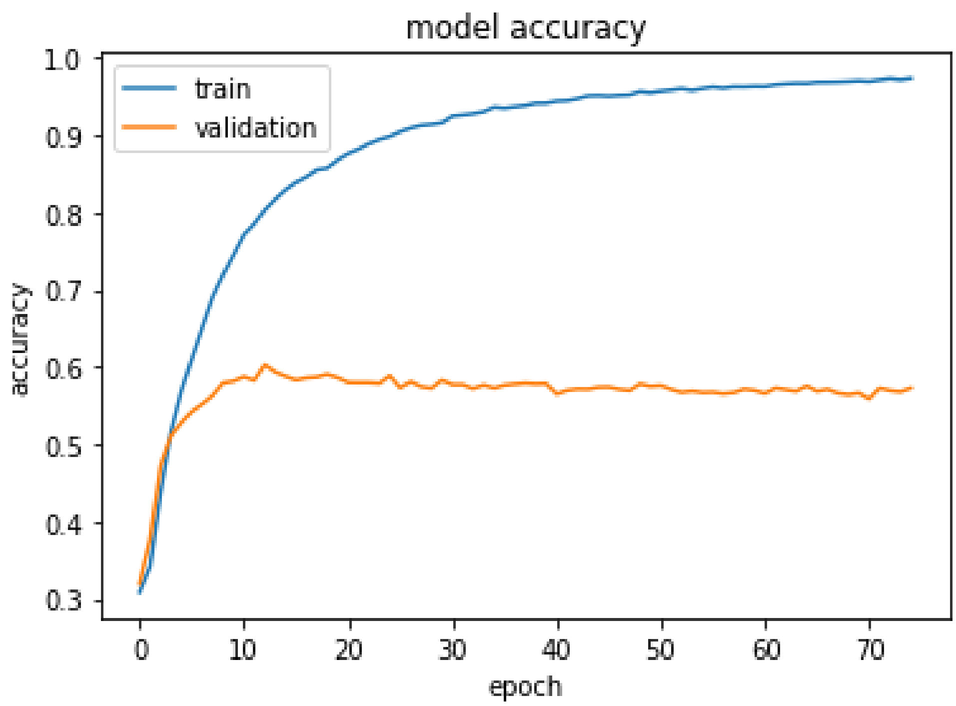


Fig 6: Model accuracy train and validation vs epochs

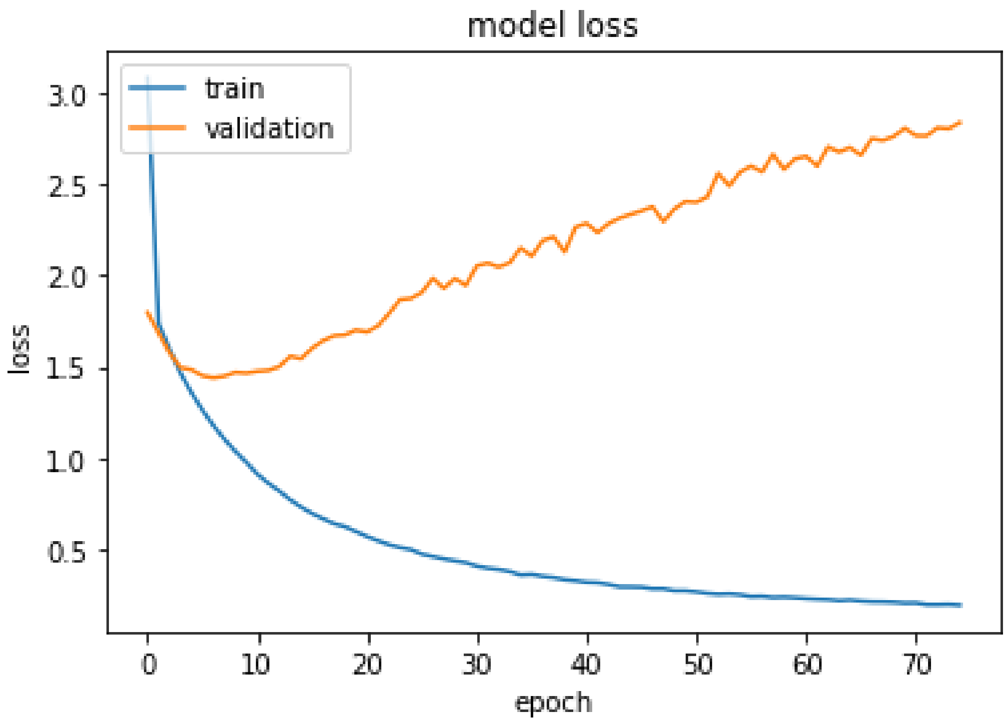


Fig 7: .Model loss train and validation vs epochs

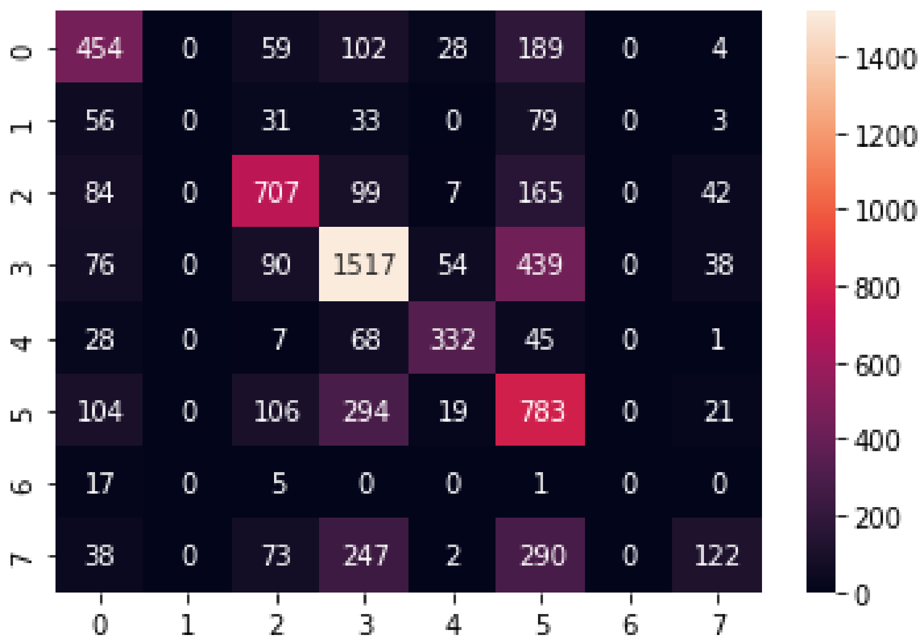


Fig 8: Confusion matrix

Comparative result analysis:

```
[ ]  predict_emotion('I am very happy and joyful today')

     1/1 [==============================] - 0s 25ms/step
     'joy'
```

```
[ ]  predict_emotion('today is beautiful')

     1/1 [==============================] - 0s 23ms/step
     'joy'
```

```
[ ]  predict_emotion('once in a while')

     1/1 [==============================] - 0s 26ms/step
     'neutral'
```

```
[ ]  predict_emotion('i almost foget hair loored looked mirror')

     1/1 [==============================] - 0s 26ms/step
     'surprise'
```

```
[ ]  predict_emotion('ON THE WAY TO MY HOMEGIRL BABY FUNERAL!!! MAN I HATE FUNERALS THIS REALLY SHOWS ME HOW BLESSED I AM  ')

     1/1 [==============================] - 0s 26ms/step
     'sadness'
```

8. CONCLUSION:

Sentiment analysis is a field of study that analyzes people’s sentiments, attitudes, or emotions towards certain entities. It tackles a fundamental problem of sentiment analysis, sentiment polarity categorization. A sentiment polarity categorization process has been proposed along with detailed descriptions of each step. Experiments for both sentence-level categorization and review-level categorization have been performed. 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 text 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.

9. REFERENCES:

[1] Laksono, R. A., Sungkono, K. R., Sarno, R., & Wahyuni, C. S. (2019). Sentiment Analyzis of Restaurant Customer Reviews on TripAdvisor using Naïve Bayes. 2019 12th International Conference on Information & Communication Technology and System (ICTS). doi:10.1109/icts.2019.8850982

[2] A. Hassan, A. Abbasi, and D. Zeng, "Twitter Sentiment Analysis: A Bootstrap Ensemble Framework," in 2013 International Conference on Social Computing, 2013, pp. 357–364.

[3] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," Decis.Support Syst., vol. 57, pp. 77–93, Jan. 2014.

[4] E. Fersini, E. Messina, and F. A. Pozzi, "Sentiment analysis: Bayesian Ensemble Learning," Decis. Support Syst., vol. 68, pp. 26–38, Dec. 2014.

[5] T. Chalothom and J. Ellman, "Simple Approaches of Sentiment Analysis via Ensemble Learning," Springer, Berlin, Heidelberg, 2015, pp. 631–639.

[6] L. Naji, "Twitter Sentiment Analysis Training Corpus (Dataset)," 22 september 2012.

[7] S. Hridoy, M. Ekram, M. Islam and R. M. Rahman, "Localized twitter opinion mining using sentiment analysis," Decision Analytics, vol. 2, no. 1, 22 october 2015.

[8] Q. Ye, Z. Zhang, and R. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches," Expert Systems with Applications, vol. 36, no. 3, pp. 6527–6535, 2009.

[9] A. S. Abrahams, J. Jiao, G. A. Wang, and W. Fan, "Vehicle defect discovery from social media", Decision Support Systems, vol. 54, pp. 87-97, 2012.

[10] B. H. Kasthuriarachchy, K. De Zoysa and H. L. Premaratne, "A review of domain adaptation for opinion detection and sentiment classification", in Advances in ICT for Emerging Regions (ICTer), 2012 International Conference on, 2012, pp. 209-213.