Project Report on
Sentiment-Instigate Pair Extraction (SIPE) : A Smart Task for Emotion Analysis in Text.
University of Regina.

Group Name: VICTORY

Krishna Patel - 200441475 - kap788@uregina.ca

Sushitha Rajeeva - 200425621 - shj986@uregina.ca

15 April 2021

Contents

1 Team Members with Role of each student

Name : Krishna Patel
Role : Sorting out best papers from research, going through it and processing and cleaning up on datasets also working on project related to coding of web application, drafting project.
UofR mail id: kap788@uregina.ca

Name: Sushitha Hanumanthappa Rajeeva
Role: Paper Research work to get different ideas for the group assignments, Project Proposal and for the main Project. Taking equal responsibility and contribution in coding part and final report making.
UofR mail id: shj986@uregina.ca

2 Introduction

In recent years, emotional linguistic analysis has become one of the most productive and bright areas of computational linguistics. Initially, predictions about semantic polarity (positive or negative) attracted attention, but at the same time, research activity moved to more accurate sentiment modeling[3][4]. Also, everything depends on feedback from small things to large items also in different generating styles typically as audio, video and texts so to understand expressed emotion and analyzing sentiment of an individual has become most important priority in each business. Sentiment Instigate Pair Extraction (SIPE) focuses at identifying the potential causes leading from text to emotional expression. In this work, we propose a new technique: sentiment instigate pair extraction(SIPE), which focuses at finding the potential cause for human to express his feelings whether it's from positive or negative impact. SIPE extracts the cause clause for an emotion to incur. For instance, consider the sentence of a document, "Yesterday morning, a policeman visited the old man with the lost money, and told him that the thief was caught. The old man was very happy, and deposited the money in the bank"[3]. There are two clauses which made an old man happy and they are "a policeman visited the old man with the lost money" and "told him that the thief was caught". These are the cause for an old man to be happy and SIPE aims at finding the clauses which cause the emotion to incur in an individual[3][4]. Semantic polarity was highlighted in early days but later than it was lengthen into more polarity classes.

Figure 1: Overall Project

3 Problem Statement

In day-to-days life there are many sentences which could cause an individual to get emotions. Sometimes it is essential to know what caused that emotion to incur in humans. In every statement it contains hidden feeling where speaker doesn't express their feeling so it's hard to detect that feelings. To detect the emotions there were two different things to cover where one side to analyze the writers and readers point of emotions and on other side achieving the performance of to different formats[1]. Whereas in other cases due to huge amount of data model usually not processed with the next sentence to predict the current utterance[2]. For transformer block and multi head attention tried other hyper parameters and worked well in linguistic annotation but not in Visual and Acoustic. In this paper, we are going to talk about the problem statement i.e.
1) In real work scenario, the emotions must be explained before cause

extraction in SIPE is limited.
2) The first task is to explain what cause the emotion to occur in an individual and then extract the cause. Later we are going to show emotion extracted as the result.
In this proposed work, we mainly concentrate on causes on emotion and extracting the causes using SIPE technique to give judgement about a person's emotion and decide whether he/she/they are happy, sad, anger, trust, joy, confident, etc[1][3][4].

4 Discussion of Related Work

We have referred two papers [3][4] for the related work. In both of the papers the main intention to find the emotion and what cause the person/individual to get the related emotion. In the Figure 1, we can see few sentences which is a short paragraph taken from [3]. From these sentences, it's quite easy to say which emotion does it contain i.e. The old man was happy and he is also worries but it does not tell us what caused the old man happy and worried.

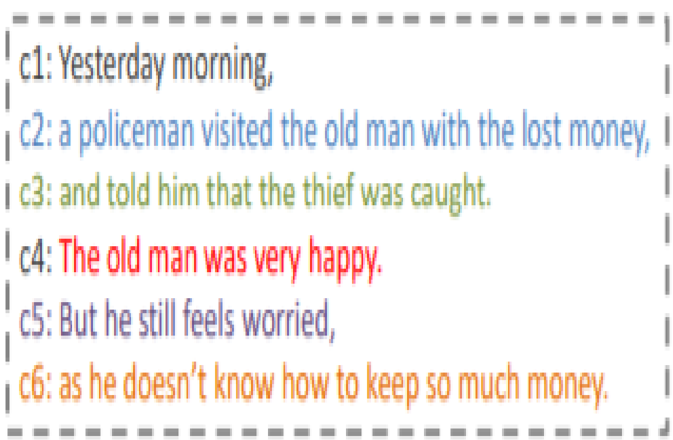

Figure 2: Few sentences from document

In Figure 1, In the sentence "The old man was very happy", shows the emotion happiness "But he still feels worries" shows that he was worried too.

It forms the emotion set and the cause set.
Emotion set =c4, c5
Cause set =c2, c3, c6
They have used a short paragraph and separated it into each line when they encounter a comma or a full stop. They have given a name to each sentence and they are trying to find the cause of the emotion happy. In this instance, c2 and c3 caused the old man to be happy meanwhile c6 made the old man worried. These are the strategies they are using in the paper [3][4].

5 Approach

Below mentioned steps are elaborate while implementing of our project.

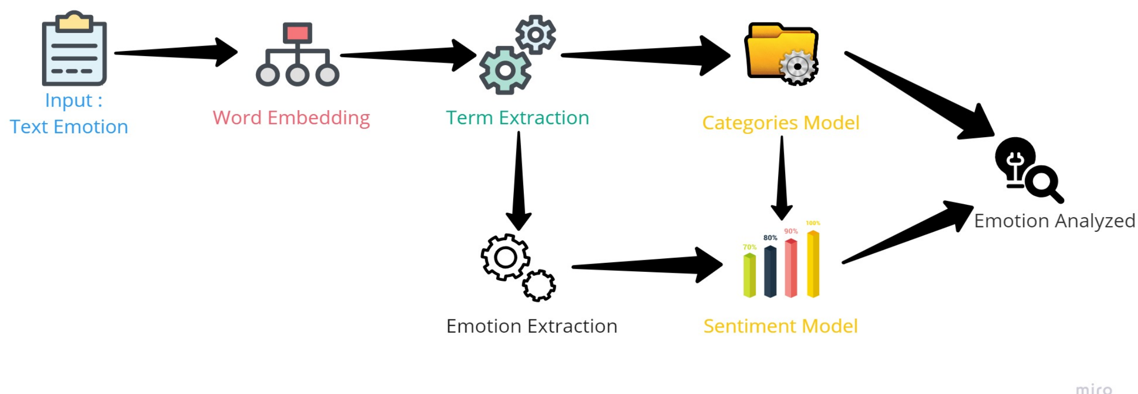

Figure 3: High Level Diagram of Data science Life Cycle

5.1 Data Selection

Initially, We have taken dataset from two papers, i.e, NLP Emotion Text Master and Emotion Cause Pair Extraction. We have trained our model using two dataset but we have got more accuracy in NLP EMotion Text Master while it shows emotion and text of reviews that contains more than 100K records in it. The data contains two features in it - Emotion and Text, Whereas in emotion there were five Various emotions mentioned in it such as Joy, Anger, Fear, Sadness, Neutral. The dataset

```
size of training set: 7934
size of validation set: 3393
joy         2326
sadness     2317
anger       2259
neutral     2254
fear        2171
Name: Emotion, dtype: int64
```

	Emotion	Text
0	neutral	There are tons of other paintings that I thin...
1	sadness	Yet the dog had grown old and less capable , a...
2	fear	When I get into the tube or the train without ...
3	fear	This last may be a source of considerable disq...
4	anger	She disliked the intimacy he showed towards so...
5	sadness	When my family heard that my Mother's cousin w...
6	joy	Finding out I am chosen to collect norms for C...
7	anger	A spokesperson said : ` Glen is furious that t...
8	neutral	Yes .
9	sadness	When I see people with burns I feel sad, actua...

Figure 4: Sample Dataset

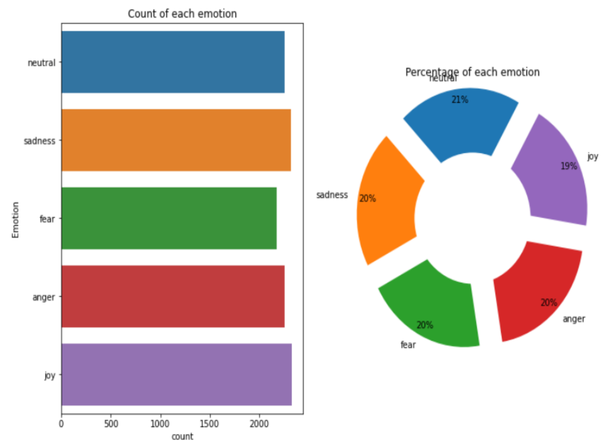

Figure 5: Count of Emotion

are in CSV format and the file size is around 1MB. Figure 1 shows the sample dataset having two columns Emotion and text while in Figure 2 it shows the count of emotions which are overall covered in dataset. Tools such as Jupyter Notebook, Google Colab are used to for preprocessing, transform the data and train different models on it.

5.2 Data Analysis

The data that is shown in Figure 1 is the raw data. Initially we need to perform preprocessing on the data and then step forward with training the various models on it. Before preprocessing the data, it is important to understand the important characteristics of the raw data. This is done in the Exploratory Data Analysis (EDA) step. Figure 3 shows few aspects of the data. There are 11327 data objects with two different attributes of which there are 1008 duplicate data objects and 0 missing values.

5.3 Data Cleaning

In this step, the data that belong to any of the below mentioned points are eliminated.

- Data that are duplicate
- Data that contain missing values
- Data whose review is only numeric

```
Total number of data objects: 11327
Total number of attributes: 2
Total number of values: 22654
Total number of duplicates data objects: 1008
Total number of missing value: 0
```

	Emotion	Text
count	11327	11327
unique	5	10272
top	joy	Yes .
freq	2326	76

Figure 6: Data Analyzed

- Data that are incorrect or improperly formatted
- Unwanted Outliers

5.4 Data Preprocessing

Once all the required irrelevant data objects are removed in the above mentioned step, the data is now ready to step forward for preprocessing In this project, the below data preprocessing tasks are then performed.

5.4.1 Elimination of noisy data

In this task, all the punctuation marks, special characters are considered as noisy data and are eliminated from each review as these characters do not depict any information for sentiment classification.

5.4.2 Elimination of Non-English reviews

Once the noisy data is eliminated, all the data objects whose reviews are not in the English language are discarded to keep the consistency among the reviews. When the punctuation marks and the special characters are eliminated in the above task, there are chances of reviews being blank. Such reviews are also eliminated in this task.

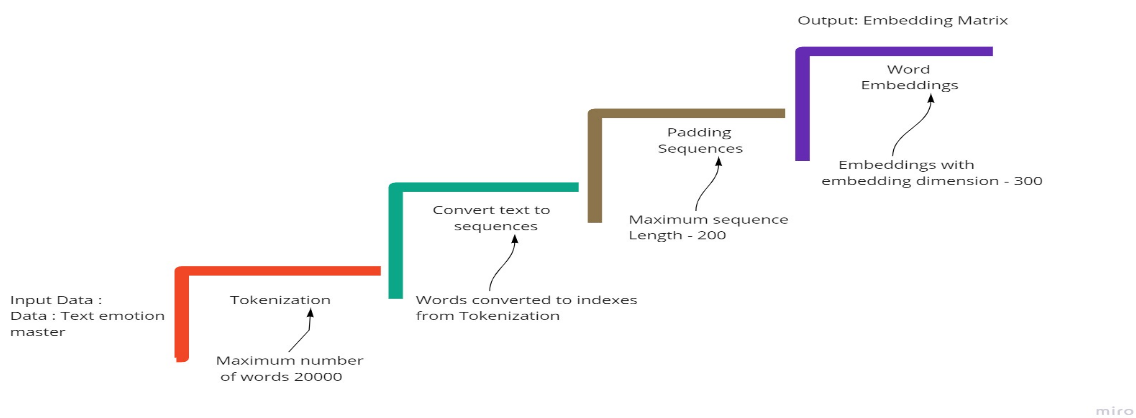

Figure 7: Steps of Data Preprocessing

5.4.3 Character case conversion

Once the data objects with non-english reviews are removed, the character case conversion is done where all the uppercase characters in all the reviews are converted to lowercase.

5.4.4 Tokenization

In this task, all the reviews are divided into small chunks known as tokens. After tokenization, each word in the review is considered as a token.

5.4.5 Elimination of stop words

Stop words are the most common words in the English language. The, Are, With, By, etc. are few of the stop words. The stop words are also eliminated as these words are not useful in classifying the sentiment of a review.

5.4.6 Stemming

Once the stop words are removed, all the remaining words are stemmed, where the words are converted to their root-form. Once all the words for a given review are stemmed, they are joined back to form the review.

5.5 Data Transformation

Once the data preprocessing is done, we are now left out with the reviews in which all the stop words are removed and the remaining words are stemmed. The reviews are in the form of text. In this project, the BERT Model , Bi-LSTM Model , Naive Bayes Classification Algorithm, Linear Support Vector , Logistic Regression and Random Forest Algorithm are used. Although we planned to work on the more Regression and classification Models, the higher training time for the algorithm made us choose a different algorithm and worked on the various datasets and eliminated most of them due to accuracy and precision level of each datasets which we have chosen related to the models.

5.5.1 BERT MODEL

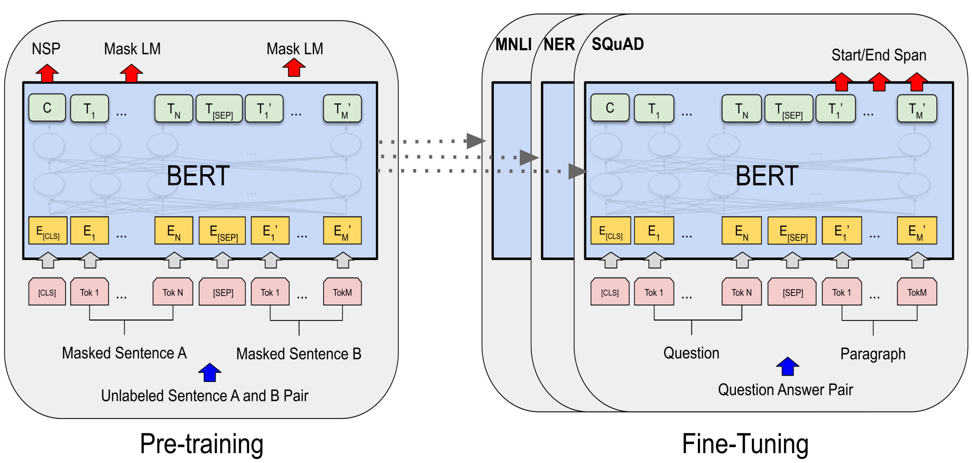

Figure 8: Pre-trained BERT model[5]

We have used a pre-trained BERT model for our project. Applying dataset of multi-class text classification on BERT model. Our dataset consists of written dialogs, messages, and short stories. Every dialog consists of an utterance/message which is labeled with one of the five emotion categories: joy, anger, sadness, fear, neutral. However, multi-class classification is done with BERT and ktrain. While preprocessing dataset in a specific way for use with BERT. This is accomplished by setting preprocess mode to 'BERT'

5.5.2 Bi-LSTM MODEL

In LSTM, the text is processed with Embedding where the representation of text takes place through words that have the similar meaning have a similar representation. We will use 300-dimensional word vectors pre-trained on Wikipedia articles. However, our dataset is quite small, and trained word vectors might not be as good as using pre-trained w2v in it. Whereas, Deep network takes the sequences of vectors as input and converted them to compressed representation. The deep network part is usually an RNN or LSTM/GRU. Then we categorized labels of emotions which are joy, fear, anger, sadness, neutral to labels 0,1,2,3,4 respectively. Stepping forward with the creation of the model pipeline the input is the first N words of each text with proper padding sequences of the text of reviews.

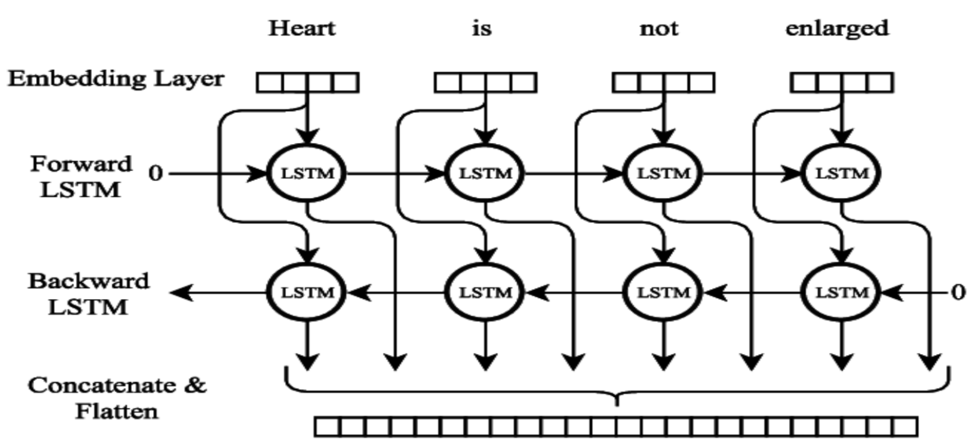

Figure 9: Bi-LSTM Model[6]

5.5.3 Other MODELS

We have also used more models such as Naive Bayes, Random Forest, Logistic Regression, and Linear support Vector we classified short messages into five emotion categories (joy, fear, anger, sadness, neutral). After importing datasets we will split the dataset into 70 percent of training and 30 percent of testing part. We will prepare our dataset NLTK and regular expressions and then vectorize words using TF-IDF (Term Frequency-Inverse Document Frequency) metric. Later we will use classifiers provided by sci-kit-learn and classify sentences into five emotion categories: joy, sadness, anger, fear, and neutral. However, most of the preprocessing steps did not improve our classification results. Since our data was mostly taken from written dialogs it was almost ready to use now.

5.6 Model Training

Once the transformation data is done, it is given as input to the model/algorithm. The transformed data is split into training and test data (in some cases, training, test, and validation data) with 70 percent of training data and 30 percent of test data. The model is trained on the training data. Once the model is trained, the parameters are used to improve the performance of the model.

5.6.1 BERT MODEL

We started with loading the pre-trained BERT Model, the BERT model and vocabulary will be automatically downloaded. BERT can handle a maximum length of 512, but in our project, we used less to reduce memory and improve speed. Then we have wrapped up in a learner object While beginning training using an one cycle policy with a maximum length of 3 epochs.

5.6.2 Bi-LSTM MODEL

Since Model Pipeline will get an LSTM/GRU layer which will receive word embeddings for each token as inputs, the LSTM layer is generating

```
begin training using onecycle policy with max lr of 2e-05...
Train on 7934 samples, validate on 3393 samples
Epoch 1/3
7934/7934 [==============================] - 475s 60ms/sample - loss: 0.9311 - acc: 0.6364 - val_loss: 0.5669 - val_acc: 0.8034
Epoch 2/3
7934/7934 [==============================] - 466s 59ms/sample - loss: 0.4569 - acc: 0.8470 - val_loss: 0.5211 - val_acc: 0.8232
Epoch 3/3
7934/7934 [==============================] - 466s 59ms/sample - loss: 0.1911 - acc: 0.9411 - val_loss: 0.5589 - val_acc: 0.8320
<tensorflow.python.keras.callbacks.History at 0x7ffa776ace10>
```

Figure 10: Training of BERT Model

a new encoding for the original input. The output level has several neurons equal to the classes of the problem and a "softmax" activation function. To check how well our model has been trained we adjusted parameters and added dropout layers. We have trained our model for 15 epochs to make our model more accurate.

```
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be rem
Instructions for updating:
Use tf.cast instead.
WARNING:tensorflow:From /Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/tensorflow/python/ops/math_grad.py:102: div (from tensorflow.python.ops.math_ops) is deprecated and will be removed
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
Train on 7934 samples, validate on 3393 samples
Epoch 1/6
7934/7934 [==============================] - 65s 8ms/step - loss: 1.3612 - acc: 0.4674 - val_loss: 1.0939 - val_acc: 0.6434
Epoch 2/6
7934/7934 [==============================] - 76s 10ms/step - loss: 0.8290 - acc: 0.7221 - val_loss: 0.7868 - val_acc: 0.7109
Epoch 3/6
7934/7934 [==============================] - 84s 11ms/step - loss: 0.6074 - acc: 0.7893 - val_loss: 0.7440 - val_acc: 0.7327
Epoch 4/6
7934/7934 [==============================] - 84s 11ms/step - loss: 0.4874 - acc: 0.8394 - val_loss: 0.7168 - val_acc: 0.7468
Epoch 5/6
7934/7934 [==============================] - 76s 10ms/step - loss: 0.3833 - acc: 0.8890 - val_loss: 0.7067 - val_acc: 0.7548
Epoch 6/6
7934/7934 [==============================] - 60s 8ms/step - loss: 0.2975 - acc: 0.9220 - val_loss: 0.7042 - val_acc: 0.7566
```

Figure 11: Training of LSTM Model

5.6.3 OTHER MODELS

After transformation of data, vectorizing text using Term Frequency technique (Term Frequency(TF) — Inverse Dense Frequency(IDF)). After our preprocess and tokenization, we have to find its TF and IDF using TF = (Number of repetitions of the word in a document) / (Number of words in a document)
IDF = log(Number of documents / Number of documents containing the word)

The multinomial Naive Bayes classifier is trained on the training data. Various types of matrices such as the word count matrix, TF-IDF matrix, binary matrix, and frequency matrix are considered for training and at last, the word count matrix is chosen. Figure 8 shows the performance metrics by matrix type where the accuracy, precision, recall, and f1score for different matrices are compared.

5.7 Model Evaluation

Once the models are trained using the training data, they are evaluated on the test data. The models are evaluated based on the performance metrics. The performance metrics that are suitable for the classification problem are Accuracy, Precision, Recall, and F1score.

5.7.1 BERT MODEL

Using the predictor model on the ktrain to get the emotion classes from the data which are joy, sadness, fear, anger, neutral. Below is the figure of the evaluated model.

5.7.2 Bi-LSTM MODEL

By use of prediction model of padding sequence of the train data, we get the evaluation of the data.

```
              precision    recall  f1-score   support

         joy       0.87      0.85      0.86       707
     sadness       0.84      0.79      0.82       676
        fear       0.86      0.87      0.86       679
       anger       0.81      0.80      0.81       693
     neutral       0.78      0.85      0.81       638

    accuracy                           0.83      3393
   macro avg       0.83      0.83      0.83      3393
weighted avg       0.83      0.83      0.83      3393

array([[598,   8,  15,  13,  73],
       [ 18, 537,  37,  54,  30],
       [ 16,  20, 590,  40,  13],
       [ 19,  49,  35, 557,  33],
       [ 37,  24,  12,  24, 541]])
```

Figure 12: Accuracy of BERT Model

```
[ ] print("Accuracy: {:.2f}%".format(accuracy_score(data_test.Emotion, predictions) * 100))
    print("\nF1 Score: {:.2f}".format(f1_score(data_test.Emotion, predictions, average='micro') * 100))

    Accuracy: 75.66%

    F1 Score: 75.66
```

Figure 13: Accuracy of LSTM Model

5.7.3 OTHER MODELS

It is also important to compare the training and validation accuracy and loss for different epochs as it helps to understand how these metrics vary concerning the number of epochs.

Figure shows the comparison of accuracy and loss on epoch in the Naive Bayes Algorithm, Random Forest, Logistic Regression, and Linear Support Vector.

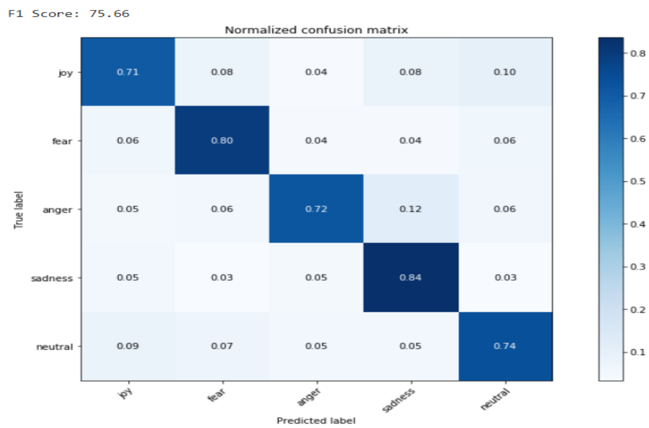

Figure 14: Confusion Matrix of LSTM Model

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 500, 300)          3626400
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 498, 256)          230656
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               65792
_________________________________________________________________
dense_2 (Dense)              (None, 5)                 1285
=================================================================
Total params: 3,924,133
Trainable params: 297,733
Non-trainable params: 3,626,400
_________________________________________________________________
```

Figure 15: Model Summary LSTM Model

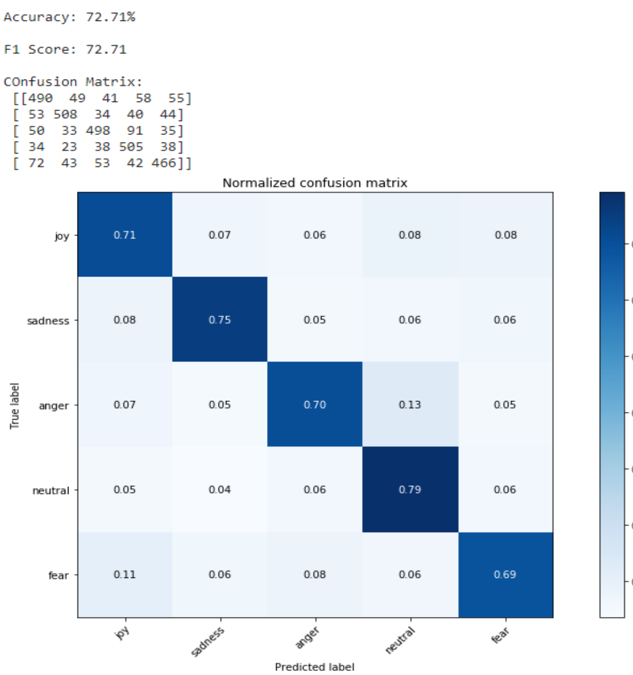

Figure 16: Accuracy and Confusion Matrix of Linear Support Vector

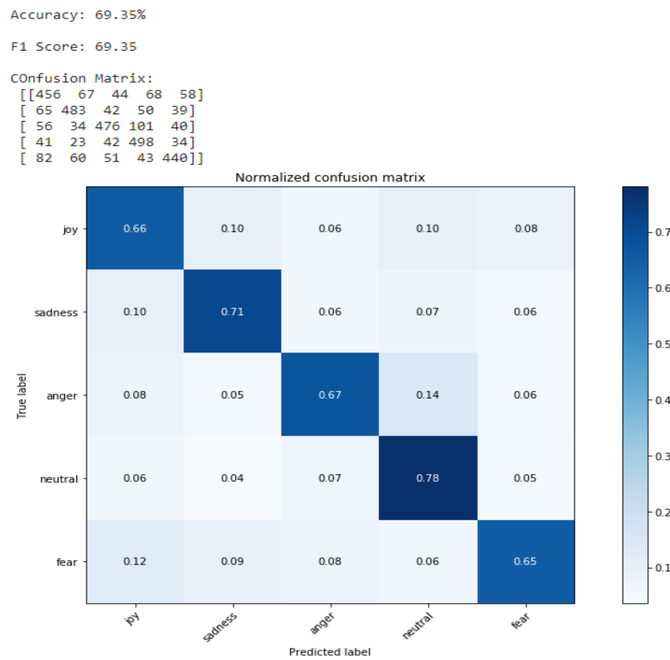

Figure 17: Accuracy and Confusion Matrix of Logistic Regression

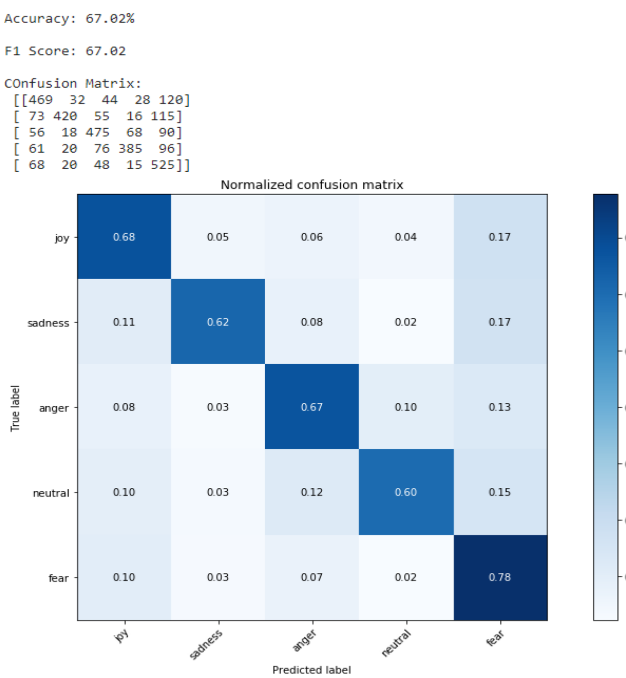

Figure 18: Accuracy and Confusion Matrix of Naive Bayes

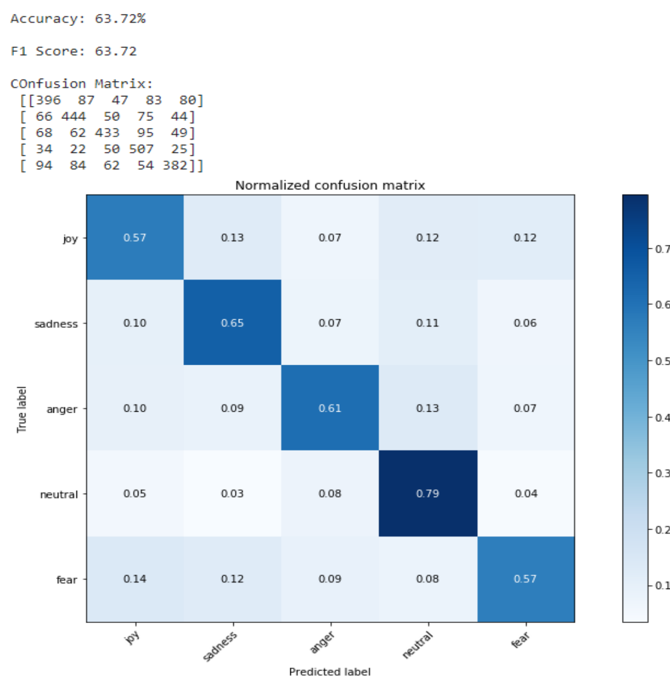

Figure 19: Accuracy and Confusion Matrix of Random Forest

5.8 Web Application

We have chosen six models i.e. BERT Model, Bi-LSTM Model, Naive Bayes Classification Algorithm, Random Forest, Logistic Regression, and Linear Support Vector in our project for selected datasets. I have compared the accuracy of all models. Among all, We found the BERT model as more accurate. We used the BERT model for making a website. The website takes a review as input, uses one of the trained models i.e. BERT on the back end to predict the probabilities of different class labels as emotions specific to review, and outputs the predicted values of emotions. A python web framework named as flask is used to create a website application. HTML and CSS are used to build the user interface for the website. Google Cloud Platform (GCP) can be used to deploy our model and can be accessed with the localhost. We can see how it works in the below figure.

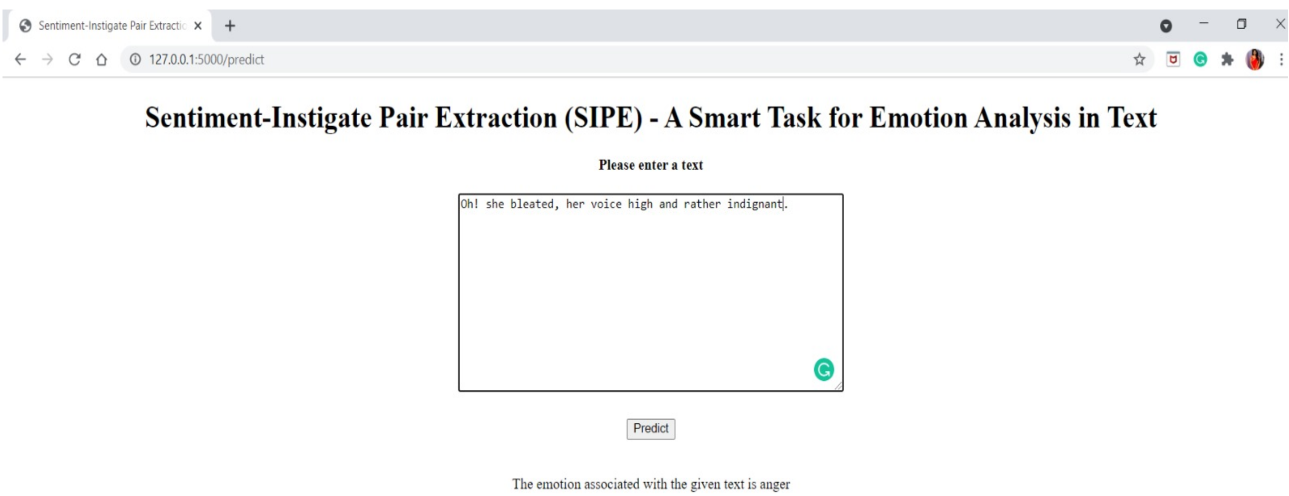

Figure 20: Example 1 : Output of Web Application using BERT MODEL

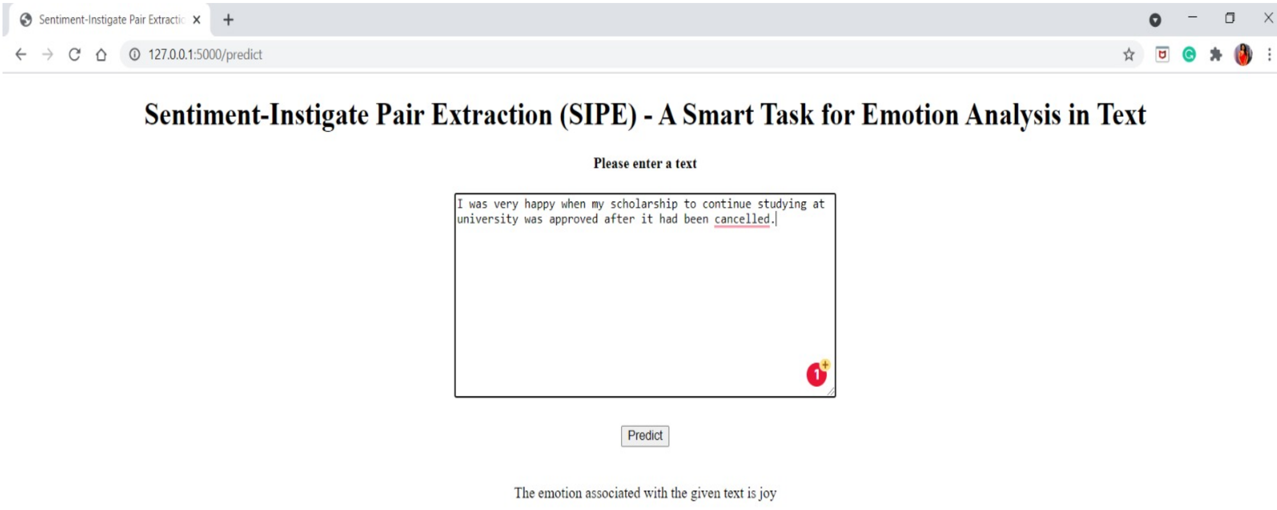

Figure 21: Example 2 : Output of Web Application using BERT MODEL

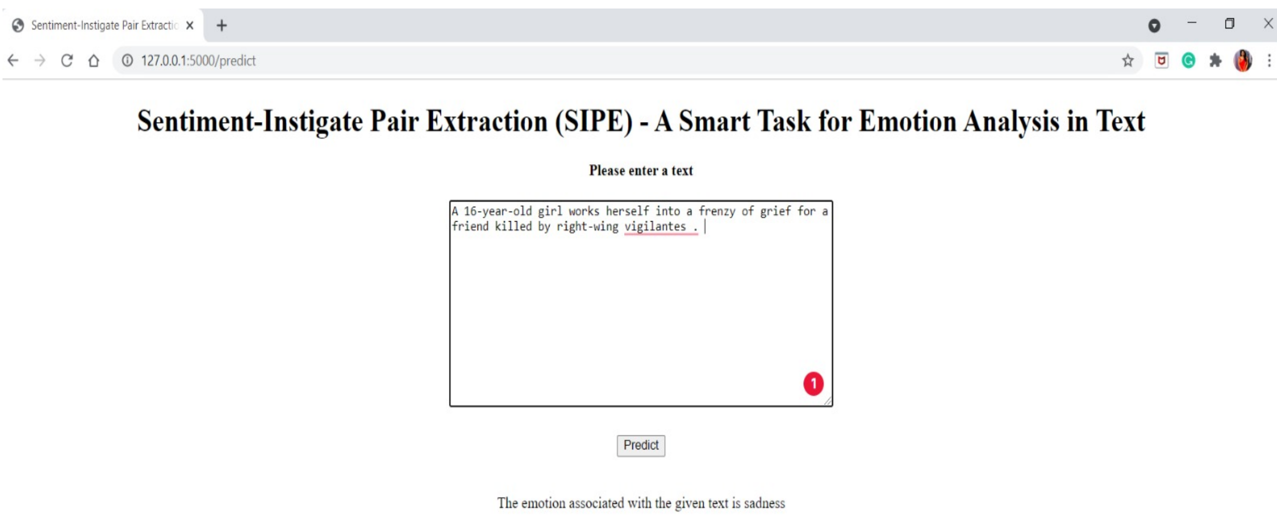

Figure 22: Example 3 : Output of Web Application using BERT MODEL

6 Data

In this project, we have used dataset from two papers[3][7]. In the first paper[3], the dataset is a large English corpus that has a short paragraph along with the emotion and cause pair. The English corpus is a huge dataset which has more than 100k line and it is in text format. We have processed the dataset and converted that text file to csv to train using our models. We can see in the below figure for the format.

```
29 4
 (3, 2),(3, 4),
1,null,null,Atkinson and others had read some of the papers published by Xerox PARC
2,null,null,so they knew they were not getting a full description
3,disgust,complain,Jobs phoned the head of the Xerox venture capital division to complain
4,null,null,a call immediately came back from corporate headquarters in Connecticut decreeing that Jobs and his group should be shown everything .
31 6
 (5, 3),
1,null,null,He was working so hard that one morning
2,null,null,in a daze
3,null,null,he drove his Corvette into a parked truck and nearly killed himself
4,null,null,Jobs immediately drove to the hospital to see him
5,fear,worried about,We were pretty worried about you
6,null,null,he said when Atkinson regained consciousness .
```

Figure 23: Part of first dataset in text format[3]

In the second paper, we have used the dataset from reviews from one of the websites[7], This dataset is in CSV format and it has 10k lines to train. While training our model, we found [7] as more convenient and we got more accuracy in that dataset. So, we used a dataset from this paper.

1	Emotion	Text
2	sadness	I experienced this emotion when my grandfather passed away.
3	neutral	when I first moved in , I walked everywhere . But within a week , I had my purse stolen â€" just a block away from the police station ! Now , I always take public transportation
4	anger	` Oh ! " she bleated , her voice high and rather indignant .
5	fear	However , does the right hon. Gentleman recognise that profound disquiet has been expressed about some of the proposals in the Bill , particularly the fast-track ones ?
6	sadness	My boyfriend didn't turn up after promising that he was coming.
7	neutral	It's freezing .

Figure 24: Part of second dataset in csv format[7]

7 Tools

Following is the list of tools that are used while working on this project:

7.1 Modules

Numpy, Panda, Matplotlib and Tensorflow: Numpy library is required by opencv-python as each frame or image is treated as an array. Panda has been used to create a CSV file that lists all the images and it's corresponding annotation details. This csv file is used to visualize the number of samples for each signpost available in the dataset using matplotlib library. We have used Tensorflow as we are using deep learning networks.

7.2 Platform

• Jupyter lab : Jupyter lab is a web browser based IDE for writing python scripts.
• Google Colab : Google Colab is an online VM platform provided by Google. I have used it for training deep learning model.
• Overleaf : It is an online tool for writing documents

7.3 Storage

• Google Drive : Created a zip file of the images and annotation files and stored it on Google drive

8 Results

Once all the models are trained, the performance metrics of the algorithms are compared. Although the output type of these algorithms vary(BERT Model outputs the class label whereas the other Model outputs the probability of each class label), the performance metric values of the BERT Model are slightly higher than the Bi-LSTM as BERT and BiLSTM are two main models in our project. We also compared it to the Classification and Regression models. And the output of all models can be seen in the below figures.

```
import time

message = 'I just broke up with my boyfriend'

start_time = time.time()
prediction = predictor.predict(message)

print('predicted: {} ({:.2f})'.format(prediction, (time.time() - start_time)))

predicted: sadness (0.06)
```

Figure 25: Output of Emotion working on BERT model

```
Message: ['delivery was hour late and my pizza was cold!']
predicted: anger (0.01 seconds)
```

Figure 26: Output of Emotion working on LSTM model

```
Message: My boyfriend didn't turn up after promising that he was coming.
Predicted: sadness
```

Figure 27: Output of Emotion working on LSTM model

9 Challenges

• The challenge faced in our project was the unavailability of devices that have high processing capabilities. So, we used online platforms like Google Colab and Jupyter Notebook that can train the algorithms remotely. Although these platforms have access to Graphics Processing Unit(GPU) and Tensor Processing Unit(TPU), the algorithms can be trained only for a limited time.
• We worked on many datasets for training each model but it was very hard to get the accuracy in each model. We shortlisted two datasets from the papers which we have referred to in this project[3][7]. We found more accuracy with the dataset in the second paper[7].
• Time constraint was one of the major issues while running a huge

dataset. It used to take more than two days to train the model

10 Future Work

As the systems with high processing GPU were not available, the Bi-LSTM is trained on fewer epochs. If we train the models on a more number of epochs then it will increase its performance and can make higher accurate predictions. As discussed, six different algorithms are trained in our project. The project can be extended by considering the classification algorithms or Regression Models on Various Datasets of emotion-text, or a combination of different algorithms, and the accuracy performance of algorithms can be evaluated. The sentiment analysis is done specifically to the reviews of customers in text data. This can be extended by considering the emotion and cause pair extraction as per we learned from the papers and the sentiment can be evaluated in furthermore secondary emotions specific to a data.

11 Conclusion

We have taken the reviews of Emotion text master from GitHub and trained six various algorithms whose performance is evaluated and the overall sentiment of different reviews is identified. With the knowledge that is discovered from the trained models, the operations research analyst makes informed decisions regarding the reviews that are accessible to the company for better performance. It also gives an overall idea of how well the users are satisfied with the services as well as the satisfaction of the products. A web application is created in which BERT Model is used to predict the probability of a review belonging to different class labels of emotions of datasets. This helps the companies to input the review and identify the predicted emotion of the review and use feedback from it.

12 References

1. **Full Reference:** Link to paper - Link to code - Paperswithcode
S. Buechel, U. Hahn, "EmoBank: Studying the Impact of Annotation Perspective and Representation Format on Dimensional Emotion Analysis" in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2017. pp, 578-585.

2. **Full Reference:** Link to paper - Link to code - Paperswithcode
J. Delbrouck, N. Tits, M. Brousmiche and S. Dupont, "A Transformer-based joint-encoding for Emotion Recognition and Sentiment Analysis" in Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML), 2020. pp, 1-7.

3. **Full Reference:** Link to paper - Link to code - Paperswithcode
R. Xia, Z. Ding, "Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts" in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019. pp, 1003–1012.

4. **Full Reference:** Link to paper - Link to code - Paperswithcode
Z. Ding, R. Xia, J. Yu, "ECPE-2D: Emotion-Cause Pair Extraction based on Joint Two-Dimensional Representation, Interaction and Prediction" in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. pp, 3161–3170.

5. **Full Reference:** Link to paper

6. **Full Reference:** Link to paper

7. **Full Reference:** Link to code