Emotions Based Text Classification

Mayur G. Warialani, Ritesh Kumar, Abhishek Varma and Ravi Ranjan

November 23, 2018

Abstract

Emotions are an essential part of our everyday life. Emotions have the ability to influence many of our decisions. How emotions vary in our daily chat conversations is an interesting problem. One approach can be to break chats into individual sentences and analyze emotional content. Also, an interesting adjacent application can be let's say how emotions vary in fictional Novels Stories. Here again, we can break down things into sentences as a valid approach. We will be looking to explore both these areas as a part of the project.

1 Problem Statement

Given a set of tweets classify the sentiment of each tweet among a set of predetermined 13 sentiment classes i.e,love,anger,sadness etc.

2 Dataset Used

The dataset is available here https://www.figure-eight.com/data-for-everyone/ under the topic "Sentiment Analysis: Emotion in Text". The dataset is about 40000 in size and has 13 Labels.

3 Tools Used

- Keras
- scikit-learn
- GloVe - Word Embeddings for twitter(50 dimensional vectors).
- WordCloud

4 Implementation Details

- In Multinomial Naive Bayes,Linear Support Vector Classifier , Logistic Regression we have sentences in to vectors using TF-IDF technique.
- In RNN,LSTM models the input is transformed using GloVe twitter 50 dimensions Vector.
- Training and Test data is split into 75:25 ratio
- Other Preprocessing includes removal of 'English' stop words.
- Removed irrelevant stuff in the dataset of tweets like user , url references.

5 Analysis

Table 1: Accuracy

Classifier	Training Accuracy	Testing Accuracy
Mutinomial Naive Bayes	43.86	31.25
Linear Support Vector Classifier	62.11	30.60
Logistic regression	47.91	33.97
LSTM	40.51	37.16

Table 2: F1-score

Classifier	Training F1-score	Testing F1-score
Mutinomial Naive Bayes	0.18	0.11
Linear Support Vector Classifier	0.60	0.17
Logistic regression	0.26	0.17
LSTM	0.22	0.19

Table 3: Precision

Classifier	Training Precision	Testing Precision
Mutinomial Naive Bayes	0.50	0.20
Linear Support Vector Classifier	0.76	0.19
Logistic regression	0.47	0.27
LSTM	0.30	0.247

Table 4: Recall

Classifier	Training Recall	Testing Recall
Mutinomial Naive Bayes	0.19	0.13
Linear Support Vector Classifier	0.53	0.17
Logistic regression	0.25	0.17
LSTM	0.22	0.193

6 Confusion Matrics on Test data:

Confusion Matrix for Support Vector Classifier:

```
[[  4   9   0  50  10   2   1   2   3   4   1   0   0]
 [  1 115   5 111 176  12  22   8  22  33   1   4   0]
 [  1   8   1  22  21   2   6   0   0  13   0   1   0]
 [  5  80   0 409 194  14  44  18   7 104   0  11   0]
 [  4 129   2 195 332  17  44  15  22  61   2  12   0]
 [  2  22   0  54  52  13  19   7   5  32   0   1   0]
 [  1  19   0  76  57   5 145   7   4  74   0   5   0]
 [  0  10   2  47  32   3  17  10   5  55   0   5   0]
 [  0  23   0  25  39   4   3   2  31  12   2   0   0]
 [  2  24   5 119  78  16  75  14   3 158   0  13   1]
 [  0   3   0   6   9   0   0   0   0   2   0   0   0]
 [  0  16   0  48  26   5  12   2   2  22   0   6   0]
 [  0   0   0   7   4   1   1   0   1   0   0   0   0]]
```

Confusion Matrix for Naive Baye's Classifier:

```
[[  0   3   0  52  28   0   0   0   0   3   0   0   0]
 [  0  39   0 160 294   0   4   0   0  13   0   0   0]
 [  0   2   0  38  25   0   2   0   0   8   0   0   0]
 [  0  10   0 513 295   0  19   0   0  49   0   0   0]
 [  0  37   0 255 504   0  14   0   0  25   0   0   0]
 [  0   6   0  86  92   0   3   0   0  20   0   0   0]
 [  0   5   0 136 107   0  80   0   0  65   0   0   0]
 [  0   4   0  76  61   0   6   0   0  39   0   0   0]
 [  0  11   0  37  90   0   1   0   1   1   0   0   0]
 [  0   5   0 217 134   0  39   0   0 113   0   0   0]
 [  0   2   0   6  12   0   0   0   0   0   0   0   0]
 [  0   5   0  71  46   0   3   0   0  14   0   0   0]
 [  0   0   0   9   4   0   1   0   0   0   0   0   0]]
```

Confusion Matrix for Logistic Regression classifier:

```
[[  1   6   0  54  20   0   0   0   1   4   0   0   0]
 [  0 115   0 147 189   6  10   0   9  34   0   0   0]
 [  0   4   0  38  18   1   5   0   0   8   0   1   0]
 [  1  51   0 503 206   5  25   3   2  87   0   3   0]
 [  0 120   0 239 395   1  22   3   3  51   0   1   0]
 [  0  20   0  79  55   7  13   1   2  30   0   0   0]
 [  0  13   0  88  64   3 133   1   3  87   0   1   0]
 [  0   8   0  67  41   1  10   4   1  54   0   0   0]
 [  0  29   0  42  48   0   1   0  17   4   0   0   0]
 [  0  20   0 164  75   4  58   5   0 178   0   4   0]
 [  0   2   0   8  10   0   0   0   0   0   0   0   0]
 [  0   8   0  56  32   0   6   1   1  29   0   6   0]
 [  0   0   0   9   4   0   1   0   0   0   0   0   0]]
```

Confusion_Matrix for LSTM :

```
[[  0   0   0  25  14   0   1   0   0   3   0   0   0]
 [  0  64   0  83 213   3   7   1  17  33   2   1   0]
 [  0   2   0  20  21   0   2   0   0  13   0   1   0]
 [  0  23   0 352 175   0  26   2   8  73   1   1   0]
 [  0  39   0 131 352   0  11   3  21  65   0   1   0]
 [  0  11   0  48  42   4  10   2   0  33   0   1   0]
 [  0   8   0  36  31   0 109   3   3  94   0   0   0]
 [  0   2   0  28  32   1   7   5   1  53   0   0   0]
 [  0   6   0  12  49   0   2   1  31   5   0   0   0]
 [  0  10   0  85  66   1  30   9   1 194   0   5   0]
 [  0   0   0   5   6   0   1   0   1   1   0   0   0]
 [  0   2   0  28  30   1   5   0   1  27   0   4   0]
 [  0   0   0   3   2   0   0   0   1   1   0   0   0]]
```

7 Conclusions

1. LSTMs work best on the given data.
2. Given the number of words in human vocabulary describing human emotions the dataset only captures a small of essence of it, thus it might give poor results on genral datasets.

8 References

https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-und
https://www.analyticsvidhya.com/blog/2017/01/ultimate-guide-to-understand
https://datascienceplus.com/multi-class-text-classification-with-scikit-l
https://medium.com/@sabber/classifying-yelp-review-comments-using-lstm-an
https://scikit-learn.org/stable/ https://keras.io/