50.038 Computational Data Science Fall Term Project

Text Based Emotion Recognition

Caryl Beatrice Aragon Peneyra[1], Lee Jet Xuen[2], and Tay Sze Chang[3]

[1]1004618
[2]1004365
[3]1004301

December 3, 2021

Contents

1 Introduction 1

2 Problem & Related Work 1

3 Data 2
3.1 Methodology 2
3.2 Data Pre-Processing 2
3.2.1 Data Cleaning 2
3.2.2 Count Vectorization . . . 3

4 Classification Models 3
4.1 Naive Bayes Classifier 3
4.2 Logistic Regression 3
4.3 Bi-LSTM with pre-trained embedding layer 3
4.3.1 Word Embedding Layer . 3
4.3.2 Stacked Bi-LSTM Layer . 3

5 Evaluation Methodology 4
5.1 Defining confusion elements . . . 4
5.2 Metrics of interest 4

6 Result & Discussion 4
6.1 Examining evaluation metrics . . 4
6.2 Impact of dataset on results . . . 5

1 Introduction

In the era of big data, machine learning is a popular research area. Machine learning allows researchers to be able to fully utilise massive data by classifying and predicting content of the data. One of the prominent field is called sentiment analysis, which is a branch of natural language processing(NLP).

This report aims to take one step further to classify emotion from text based data. Emotion recognition can help service providers provide tailored services to their customers. The objectives of our project is to predict one of the six emotions(anger, fear, joy, love, sadness, surprise) given the text based data.

The rest of this report is structured as follows. Section 2 will be illustrating the problem that this report aim to address as well as some of the related and past researches that have been done. Section 3 gives an overview of the data set that this report utilises and explains the data cleaning and preprocessing methods that are applied. As a remedy, section 4 introduces the classification models such as Naives Bayes Classifier, Logistic Regression and Bi-LSTM with a pre-trained GloVe embedding layer, which the result is then evaluated in section 5. Section 6 presents the result and discussion based on the result while last section concludes.

2 Problem & Related Work

Despite the problem not being a completely new subject, many group of researchers have been working on emotion recognition in conversation or certain text based data, such as movie reviews. Those researches aim to draw some insightful results and overcome certain language subtleties as text can have a mixture of conflicting feelings that may pose great challenges to classify texts into one definite category.

For example, Vijayaragavan et al.[1] developed a Support Vector Machine(SVM) based classification for sentimental analysis of online product reviews. The research aims to determine the possibility of a customer to purchase a product by applying different algorithms to the model.

Bernhard's work has direct implications for management, academia and finance, etc. As such, various application areas of decision support – such as customer support, marketing, or recommendation systems – can be improved considerably through the use of effective computing. The research has found that effective computing allows one to infer individual and collective emotional states from textual data and thus offers an anthropomorphic path for the provision of decision support.[2]

However, this report aims to address the difference between traditional machine learning algorithms, namely Naives Bayes Classifier, Logistic Regression and a modern architecture, Bi-LSTM with a pre-trained GloVe embedding layer.

3 Data

This report uses the data set from an online source, Kaggle[3]. There are a variety of sources online for text based emotion detection, however, the one that is retrieved is one the most user friendly source.

This data set consists of a total of 20,000 lines of data and contains two columns: text, which contains a line of text, and emotion, which contains a single emotion tagged to the text. The text in each row is labelled with one of six emotions - anger, fear, joy, love, sadness, and surprise. The data set is split into 3 portions namely, training, validation and test data set as shown in figure 1-3 with the number of data for each emotion in each data set. Each partition has the sizes as shown in Table 1.

Partition	Lines of data
Training	16000
Validation	2000
Test	2000
Total	20000

Table 1: Partition sizes

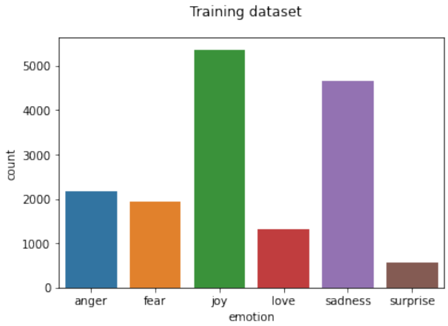

Figure 1: Counts of data of training data set on each emotion

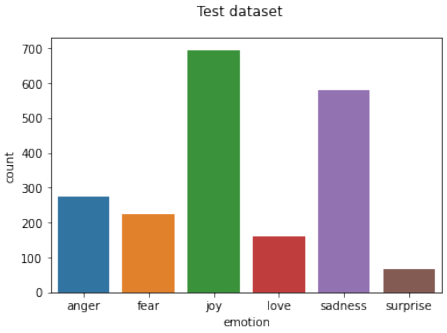

Figure 2: Counts of data of test data set on each emotion

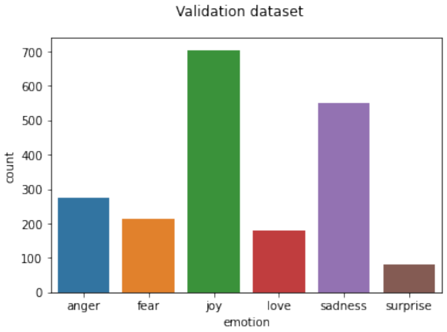

Figure 3: Counts of data of validation data set on each emotion

3.1 Methodology

First, this report feeds the data into the data cleaning and preprocessing. Then, the stop words and some punctuation is removed from the original data. Next, this report applies some vectorization techniques to convert text into a feature matrix. Lastly, this report applies a few different algorithms, namely Naives Bayes Classifier, Logistic Regression and a Recurrent Neural Network (RNN) with a Bidirectional Long Short-Term Memory (BiLSTM) layer to train and test the feature matrix.

3.2 Data Pre-Processing

3.2.1 Data Cleaning

In order to facilitate the data prediction in the later work, raw texts obtained from the online sources are preprocessed. Elements such as stop words, punctuation, numbers such as "a", "the", "of", etc are removed since they have little significance on the information about the user's emotion. Then all words are converted to lower case, and words that are infrequently occurring are removed.

3.2.2 Count Vectorization

Vectorization is the process of converting text data to numerical form and express it in a matrix representation. In the count vectorization, a document term matrix is generated where each cell is the term frequency, which is the count corresponding to the data indicating the number of times a word appears in a document.

Each column is dedicated to each word in the corpus. The count is directly proportionate to the correlation of the category of the text data. In other words, if a word appears frequently in the data that is labelled joy, then this particular word has a high predictive power of determining if a new data containing this would be joy.

4 Classification Models

4.1 Naive Bayes Classifier

The Multinomial Naive Bayes Classifier is used to classify text into multiple emotions due to its simplicity and the ease of implementation. It is based on the assumption that each feature $x_{(j)}$ is conditionally independent given the class y^i. The probabilities are given as the following:

$$P(x_j = 1|y = 1) = \frac{\sum_{i=1}^{n} \{x_j^{(i)}=1 \cap y^{(i)}=1\}}{\sum_{i=1}^{n} \{y^{(i)}=1\}}$$

$$P(x_j = 1|y = 0) = \frac{\sum_{i=1}^{n} \{x_j^{(i)}=1 \cap y^{(i)}=0\}}{\sum_{i=1}^{n} \{y^{(i)}=0\}}$$

$$P(y = 1) = \frac{\sum_{i=1}^{n} \{y^{(i)}=1\}}{n}$$

The probability of the text being classified as the i-th emotion is given as $P(y_i = 1|x)$:

$$\frac{\prod_{j=1}^{d} P(x_j=1|y=1)P(y=1)}{\prod_{j=1}^{d} P(x_j=1|y=0)P(y=0)+\prod_{j=1}^{d} P(x_j=1|y=1)P(y=1)}$$

4.2 Logistic Regression

Logistic regression is used to classify by using a sigmoid function that gives a probabilistic output. The output is given by:

$$h(x) = \frac{exp(\theta \cdot x+\theta_0)}{1+exp(\theta \cdot x+\theta_0)}$$

The probabilities of y_i being the class given an observation x is given by:

$$P(y_i|x) = \begin{cases} h(x) & \text{for } y_i = +1 \\ 1 - h(x) & \text{for } y_i = -1 \end{cases}$$

To maximize the probabilistic output, the maximum likelihood is taken. The following function is maximized:

$$E(\theta, \theta_0) = -\frac{1}{n} \sum_{i=1}^{n} log \left(\frac{1}{P(y^i|x^i)} \right)$$
$$= \frac{1}{n} \sum_{i=1}^{n} log \left(1 + exp(-y^i(\theta \cdot x^i + \theta_0))\right)$$

θ is then updated using stochastic gradient descent. The objective function for each t^{th} instance is given by:

$$e^{(t)}(\theta) = log \left(1 + exp(-y^{(t)}(\theta \cdot x^{(t)} + \theta_0))\right)$$

The gradient is given by:

$$\nabla e^{(t)}(\theta) = \frac{-y^{(t)} x^{(t)}}{1+exp(y^{(t)}(\theta \cdot x^{(t)}+\theta_0))}$$

The weight update is:

$$\theta \leftarrow \theta - \eta \nabla e^{(t)}(\theta)$$

4.3 Bi-LSTM with pre-trained embedding layer

The third model used is an RNN with the following architecture:

- Word embedding layer
- BiLSTM with 4 hidden states and 2 stacked layers
- Fully connected linear layer with 64 hidden states
- ReLU activation function
- Dropout layer with dropout rate 0.2

Early stop is implemented during model training to minimize loss. The Adam algorithm is also used for weight optimization.

4.3.1 Word Embedding Layer

The embedding layer matrix contains pretrained GloVe vectors trained on the "Wikipedia 2014 + Gigaword 5" dataset, which was trained on a corpus of 6 billion tokens and contains a vocabulary of 400 thousand tokens.[4]

4.3.2 Stacked Bi-LSTM Layer

In a bidirectional LSTM, the model consists of two hidden LSTMs: one that learns the sequence of the provided input in the forward direction, and another that learns in the backward direction. This allows the network to learn both the previous and future context of the textual input i.e. knowing the words that come directly before and after a given word in a sentence.

Given an input sequence $x = (x_1, x_2, ..., x_n)$, a BiLSTM calculates a forward hidden sequence $\overrightarrow{h_t} = (\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_n})$ and a backward hidden sequence $\overleftarrow{h_t} = (\overleftarrow{h_1}, \overleftarrow{h_2}, ..., \overleftarrow{h_n})$.

The encoded vector y_t is obtained from concatenating the final forward and backward outputs, $y_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$.

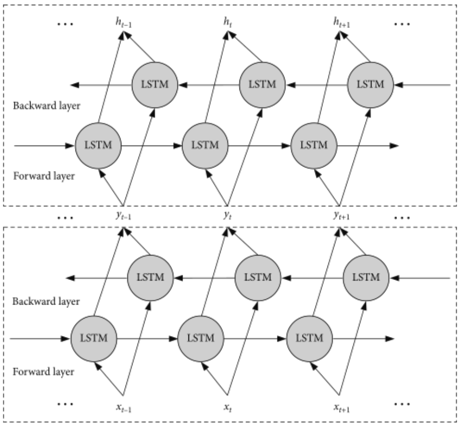

Figure 4: Architecture of a stacked BiLSTM network.[5]

$$\begin{aligned}\overrightarrow{h_t} &= \sigma(W_{\vec{h}x}x_t + W_{\vec{h}\vec{h}}\overrightarrow{h_{t-1}} + b_{\vec{h}}),\\ \overleftarrow{h_t} &= \sigma(W_{\overleftarrow{h}x}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h_{t+1}} + b_{\overleftarrow{h}}),\\ y_t &= W_{y\vec{h}}\overrightarrow{h_t} + W_{y\overleftarrow{h}}\overleftarrow{h_t} + b_y,\end{aligned}$$

Where $y = (y_1, y_2, ..., y_t, ..., y_n)$ is the output sequence of the first hidden layer, σ is the logistic sigmoid function, x_t is the t-th word vector of the input sequence, h_t is the hidden state, and W and b terms indicate the weight matrices and bias vectors respectively.

In a stacked BiLSTM network, the output y_t from the lower layer becomes the input of the upper layer:[5]

$$h_t = W_{h\vec{h}}\overrightarrow{h_t} + W_{h\overleftarrow{h}}\overleftarrow{h_t} + b_h$$

5 Evaluation Methodology

We evaluate each algorithm using the metrics from the classification report generated by the `sklearn.metrics.classification_report` function.

5.1 Defining confusion elements

A multi-class classification model with n classes generates an $n \times n$ confusion matrix of the form,

$$C = \begin{bmatrix} c_{11} & \cdots & c_{nn} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}$$

The confusion elements for each class are hence defined as such:

- True positive: $tp_i = c_{ii}$
- False positive: $fp_i = \sum_{j=1}^{n} c_{ji} - tp_i$
- False negative: $fn_i = \sum_{j=1}^{n} c_{ij} - tp_i$
- True negative: $tn_i = \sum_{j=1}^{n} \sum_{k=1}^{n} c_{jk} - tp_i - fp_i - fn_i$

5.2 Metrics of interest

This report evaluates the models used using the following metrics:

- Accuracy: Percentage of correct classifications out of all observations

$$\text{accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

- Precision: Percentage of classified items that are correct

$$\text{precision} = \frac{tp}{tp + fp}$$

- Recall: Percentage of correct items that are classified

$$\text{recall} = \frac{tp}{tp + fn}$$

- F1-score: Harmonic mean of precision and recall

$$\text{recall} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

6 Result & Discussion

6.1 Examining evaluation metrics

Tables 2, 3, 4 detail the evaluation metrics for the Logistic Regression, Naive Bayes Classifier and BiLSTM models respectively. Out of the three models used, the Logistic Regression showed the best performance.

	Precision	Recall	F1-score
anger	0.88	0.87	0.88
fear	0.85	0.84	0.85
joy	0.90	0.94	0.92
love	0.79	0.75	0.77
sadness	0.93	0.93	0.93
surprise	0.75	0.58	0.65
Accuracy			0.89
Macro avg.	0.85	0.82	0.83
Weighted avg.	0.89	0.89	0.89

Table 2: Classification Report for Logistic Regression model

When considering accuracy, the Logistic Regression model performs the best at a value of 89%, which is higher than 80% and 86% for the Naive Bayes Classifier and BiLSTM models respectively.

The Logistic Regression model again performs the best with values of 85% and 89% for the

	Precision	Recall	F1-score
anger	0.91	0.66	0.76
fear	0.83	0.64	0.72
joy	0.78	0.96	0.86
love	0.86	0.35	0.50
sadness	0.78	0.93	0.84
surprise	0.70	0.11	0.18
Accuracy			0.80
Macro avg.	0.81	0.61	0.65
Weighted avg.	0.80	0.80	0.78

Table 3: Classification Report for Naive Bayes Classifier model

	Precision	Recall	F1-score
anger	0.86	0.84	0.85
fear	0.77	0.85	0.81
joy	0.89	0.89	0.89
love	0.78	0.67	0.72
sadness	0.92	0.91	0.92
surprise	0.64	0.70	0.67
Accuracy			0.86
Macro avg.	0.81	0.81	0.81
Weighted avg.	0.86	0.86	0.86

Table 4: Classification Report for BiLSTM model

macro and weighted average precisions respectively.

The Logistic Regression model also attains the highest recall values of 82% macro averaged recall and 89% weighted average recall.

Correspondingly, the F1 scores for the Logistic Regression are the highest among the three models, with a macro averaged F1 score of 0.83 and a weighted average score of 0.89.

6.2 Impact of dataset on results

Table 5 shows the support values for each emotion, which indicates the number of actual occurrences of each emotion in the test dataset. The test dataset is unbalanced, with 34.75% of texts labelled with joy, 29.05% labelled with sadness, and only 3.3% labelled with surprise. The training and validation datasets are similarly unbalanced as can be seen from Figures 1 and 3.

Class	Support	Percentage (%)
anger	275	13.75
fear	224	11.2
joy	695	34.75
love	159	7.95
sadness	581	29.05
surprise	66	3.3

Table 5: Test partition support values

The classification reports show that the F1-score of a class for any model increases when the class has a larger percentage of occurrences in the dataset. From the Logistic Regression model metrics in Table 1, the larger joy and sadness classes have similar F1 scores of 0.92 and 0.93 respectively, while the smaller surprise class has a significantly lower score of 0.65.

Conclusion

This report proposes a methodology to classify emotion from text based data by predicting one of the following six emotions (anger, fear, joy, love, sadness, surprise) given the text based data. The data is first cleaned by removing stop words, converted to lower case and the most frequent words are taken. It is then followed by vectorization by converting text to a matrix representation using word counts. Finally it is then fitted into the various model including Naive Bayes Classifier, Logistic Regression and Bi-LSTM using GloVe the word embedding layer. The model that gave the highest accuracy was the logistic regression with 89% accuracy.

Future Work

Future work can include other complicated models for analysis. For example, 1D Convolutional Neural Network could be used to provide better performance as it demonstrates longer effective memory.[6] In addition, future work can take it a step further to analyze conversations rather than texts as sequences of text messages could provide more insights on the emotion of each text messages.[7]

To better improve the classification accuracy of the models, future work can include methods such as oversampling, undersampling or collecting more data to remedy an unbalanced dataset.

GitHub Repo Link

https://github.com/carrotbeetrice/cds-project

References

[1] V. Pichiyan, R. Ponnusamy, and A. Murugaiyan, "An optimal support vector machine based classification model for sentimental analysis of online product reviews," *Future Generation Computer Systems*, vol. 111, 05 2020.

[2] B. Kratzwald, S. Ilić, M. Kraus, S. Feuerriegel, and H. Prendinger, "Deep learning for affective computing: Text-based emotion recognition in decision support," *Decision Support Systems*, vol. 115, p. 24–35, Nov 2018. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2018.09.002

[3] Praveen, "Emotions dataset for nlp." [Online]. Available: https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp

[4] stanfordnlp, "Glove: Global vectors for word representation." [Online]. Available: https://github.com/stanfordnlp/GloVe

[5] L. Cai, S. Zhou, X. Yan, and R. Yuan, "A stacked bilstm neural network based on coattention mechanism for question answering," *Computational Intelligence and Neuroscience*, vol. 2019, p. 1–12, Aug 2019.

[6] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018. [Online]. Available: http://arxiv.org/abs/1803.01271

[7] R. Pappagari, P. Żelasko, J. Villalba, L. Moro-Velazquez, and N. Dehak, "Beyond isolated utterances: Conversational emotion recognition," 2021.