Finding Your Personality Palette using Different RNN Models

Chieh-Yu Chuang
National YangMing ChiaoTung University
Hsinchu, Taiwan
chiehyu.cs09@nycu.edu.tw

Chun-Pei Chen
National YangMing ChiaoTung University
Hsinchu, Taiwan
chunpei19.cs09@nycu.edu.tw

Abstract

The main purpose of our project is to find the tailored palette based on people's emotions. We utilized different machine learning algorithms, including the Multinomial Logistic Regression model, GRU, LSTM and Bi-LSTM, to do sentiment analysis from a given dialog and derived a transition function to generate the customized palette. For more implementation details and results, you can refer to our Github Repository.

1. Introduction

The topic idea comes from our enthusiasm towards color science and psychological tests. We found out that the psychology tests nowadays restrict their participants to answer particular questions to get an customized output. However, since people's emotions and personality are quite complicated, we think it would be better to accept a wider range of inputs to analyze the results better. Also, we believe that dialogues may not contain only one type of emotions, it could be complicated combinations. Therefore, we chose to show our results via colors.

This report elaborates the details of how we apply different algorithms, including Multinomial Logistic Regression (MLR) as our baseline, and other deep learning approaches such as Gate Recurrent Unit (GRU), Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (Bi-LSTM), to do emotion detection. We will further compare their performance by evaluating the test data accuracy and their running time. But since we generate colors from the probability output from those algorithms so there might be limitations that the results may not reflect the true emotion combinations.

2. Related Work

In this project, we chose the supervised learning algorithm, Multinomial Logistic Regression. [6].

In addition, we applied three kinds of RNN models to perform sentiment analysis as mentioned above. We decided to use RNN since it takes historical data into account and have proved its superiority in processing sequential data [1]. The algorithm is crucial for Natural Language Processing because our input sentences are a kind of time series data. We will put our main focus on Bidirectional Long Short-Term Memory for this project, proposed by Zhou et al. [9] which had achieved impressive accuracy of 52.4% and 89.5% on SST-1 and SST-2 respectively. Comparisons will be done towards Bi-LSTM, LSTM and GRU, which is also proposed and evaluated by many scholars [1–4].

To get the perfect mixture of colors from the probability outputs, we referred to several research papers to figure out the association among emotion and colors [5, 7] and derive the color transition function by handling rgb values.

3. Methodology

3.1. Baseline : Multinomial Logistic Regression

Logistic Regression is a kind of supervised machine learning algorithm which can find a straight line to separate two categories. It transforms its output using the sigmoid function to return a probability of a certain category.

Multinomial Logistic Regression is an extension of Logistic Regression which can classify multiple categories. Since we got five emotions to classify, we chose to use this technique for our project. The implementation details will be demonstrated below.

In the beginning, we remove NA values and stopwords from input. 'Stopwords' represents frequently used words, for example, 'the', 'and', 'I', etc.) They don't add much meaning to a sentence so we remove it to reduce the amount of training data.

Then, we split the dataset into train data and test data.

Lastly, using Pipeline from sklearn, we do binary classifications sequentially to get multiple classification.

3.2. Main Approach : GRU, LSTM, Bi-LSTM

In this section, we will go through the details of how we implement the deep learning algorithms mentioned above.

We divide the whole process into two stages.

STAGE 1: Handling the Input

We remove special symbols, NA values, and stopwords from input. Then, we transfer the words to vectors using one hot encoding and split the dataset into train data and test data.

STAGE 2.0: Starting to Train Model

We use Sequential Model from Keras. To prevent overfitting, we run dropout() between every layer to randomly remove some nodes.

STAGE 2.1: Embedding Layer

For the first layer, we use Embeddding() to do dimension reduction. It transfers one hot encoding to distributed representation using unsupervised learning. Original one hot encoding can't express the relations of words, but distributed representation can do that. For example, machine can learn that happy and glad are synonymous. After using Embedding(), we can reduce the amount of training data.

STAGE 2.2: RNN layer

The second layer is crucial. It uses special RNN neurons, and it shows the differences between three methods.

LSTM: It has four inputs (input gate, output gate, forget gate, and result from previous part) and one output. Gates can optionally let information go through. The input gate and the output gate decide whether the input or output is valid or not. The forget gate decides when to forget the previous memory. Unless the forget gate closes, the memory will remember all previous data. In that case, LSTM cell has better performance than simple RNN, which only remembers last input.

Bi-LSTM: It shares the same attributes, and it phrases sentences from two directions.

GRU: It is the simplified version of LSTM. It combines forget gate and input gate to update gate. This method reduces training time and also gets good performance.

STAGE 2.3&4: Two Normal Neuron Network Layers

In the end of the model, we use two normal neuron network layers, resulting in five output nodes correspond five emotions. In general case, the maximum value is considered as the emotion of the sentence. However, we directly transfer the probabilities to colors.

3.3. Color Transition Function

After generating the probability output from different models, we first derive five rgb combinations to represent five different emotions. Next, we multiply the different rgb values with the proportion obtained from the previous stage and add them up to get the new color mixture. Throughout the process, we added the parameter alpha to give different weights to the emotions that account for the highest, second highest and lowest percentage in order to get a more obvious result.

4. Experiments and Results

4.1. Baseline and Main Approaches Comparison

RNN models achieves a higher accuracy than MLR (baseline) after processing the same iterations. However, our baseline spends much less running time. We can think of Logistic Regression as a neuron network which has only one layer and cannot remember previous input, so it has worse performance but less running time.

4.2. Different RNN Models Comparison

The comparison is done by evaluating the test data accuracy and running time :

Test Data Accuracy

Before training the model, we thought that Bi-LSTM is the best and GRU is the worst. However, it turns out that three models get similar results and GRU performs better than Bi-LSTM and LSTM after processing the same iterations and learning rate.

After researching some papers [8], we found that GRU can be better in some situations especially in small datasets. Comparing with the essay using millions of data , our dataset turns out to be pretty small (about twenty thousand pieces of data), which may be the reason of the result we get. In addition, Bi-LSTM is only slightly better than LSTM, which may be another consequence resulted from the small and simple dataset.

Running Time

As for the running time, Bi-LSTM takes the longest, while GRU takes the shortest. This is because GRU requires fewer gates by connecting the input gate and forget gate.

Optimal Learning Rate

On top of the comparison, we tried to adjust the learning rate of the models and see if we can optimize the results. In consequence, e found out that the optimal value of learning rate shall be approximately 0.01, which performs the best for our dataset.

4.3. Conclusion

To conclude, GRU turns out to be the most efficient model under our dataset due to high accuracy and low cost.

For future directions, we plan to collect a larger amount of data and more complicated sentences, aiming to get more obvious differences between different RNN models. Also, we could train a multiple output model containing the emotion proportions to get a more accurate prediction of the emotion combination. The representative colors of emotions could also be trained by deep learning algorithms to increase its credibility.

References

[1] Kiran Baktha and BK Tripathy. Investigation of recurrent neural networks in the field of sentiment analysis. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 2047–2050. IEEE, 2017. 1

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 1

[3] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017. 1

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[5] Naz Kaya and Helen H Epps. Relationship between color and emotion: A study of college students. *College student journal*, 38(3):396–405, 2004. 1

[6] WP Ramadhan, STMT Astri Novianty, and STMT Casi Setianingsih. Sentiment analysis using multinomial logistic regression. In *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, pages 46–49. IEEE, 2017. 1

[7] Lisa Wilms and Daniel Oberfeld. Color and emotion: effects of hue, saturation, and brightness. *Psychological research*, 82(5):896–914, 2018. 1

[8] Shudong Yang, Xueying Yu, and Ying Zhou. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, pages 98–101. IEEE, 2020. 2

[9] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*, 2016. 1