Finding The Perfect Playlist with Text Emotion Classification

Yong-Ching Liang
Department of Computer Science
National Yang Ming Chiao Tung University
carrieliang.cs09@nycu.edu.tw

Yu-Rou Lin
Department of Computer Science
National Yang Ming Chiao Tung University
wn4doe.cs09@nycu.edu.tw

Abstract

Music gives great companionship to most people along with any kind of situation, but most of the time we are not able to successfully identify which emotion we are going through at present. In this project, we classify people's mood by approaches including Decision Tree, Logistic Regression, and BERT Fine-tuning. Regarding the validation accuracy, we found that the BERT Fine-tuning model has the best learning performance, so we further implemented this model by using the determined result to find the corresponding song.

1. Introduction

People nowadays tend to listen to music that matches their mood at the moment, unfortunately, we can't always correctly classify the mood that we are experiencing because it might be a situation or some mixed feelings with different ratios of each of them, not to mention finding the corresponding songs. By using three distinct approaches, we decided to assort our users' instant emotion with a model trained for Natural Language Processing (NLP), compare to figure out which gives the most accurate prediction with a validation data set, and find the perfect song recommendation using the best model on a worldwide streaming platform Spotify.

2. Related Work

2.1. BERT Model

There are several existing studies dealing with NLP, after we decided to use BERT Fine-Tuning as our main approach, we referred to the paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understand" [1], which gives a clear explanation on Fine-tuning handling single sentence classification and several recommended parameter values.

2.2. Data Set

The dataset we were using in this project from kaggle : "Emotions dataset for NLP classification tasks" [2] , which includes sentences with corresponding emotion labels : sadness, joy, anger, fear, love and surprise. There are 20000 sentences in the dataset in total, 6761 of joy, 5797 of sadness, 1641 of love, 2373 of fear, 719 of surprise and 2709 of anger. We combined the train, test, and validation file that kaggle provides, then separated into 18000 train data and 2000 test data.

3. Methodology

3.1. Decision Tree

The target of this project is to classify sentences based on features, therefore we choose the Decision Tree as our baseline method. Decision Tree is a supervised learning technique that consists of branches and nodes. The nodes represent a feature, the branches represent different values of the nodes, and the leaves represent the decision results. Decision Tree uses greedy algorithm to make the optimal choice to split into branches at each node. However, there are some concerns about the Decision Tree. First, it's easily overfitted because the model only stops while all the information is in a single class, and there may be inappropriate splitting using noise data. Also, the greedy algorithm to split the nodes makes choices locally, which may not be the optimal global choice of the Decision Tree. The Advantages of Decision Tree are that it's simply understandable, and efficient while processing data.

3.2. Logistic Regression

Logistic Regression model finds out logistic functions that can separate and classify the data. The logistic functions are a type of sigmoid function, which is an S shaped graph and maps any real numbers to probability between 0 and 1. One will be classified in the class if its probability precedes 50%, else it will be classified in the opposite class. The concern of Logistic Regression is that if the features are more than the observed data, the model may be overfitted.Also, Logistic Regression can't solve nonlinear problems because it has to make linear decisions. Advantages of Logistic are simple and efficient to use, and output as a probability format which may support further use .

3.3. BERT Fine-Tuning

By using the embeddings of a pre-trained BERT (Bidirectional Encoder Representation from Transformers) model, Fine-tuning means to use minimal task-specific parameters and train on the downstream tasks. In this project, we use BERT to train a text classifier by adding an untrained layer of neurons on the end. We chose this approach instead of other deep learning models because the pre-trained model takes much less time to train, and requires less data compared to building a model from scratch.

3.4. Integration with Spotify

In this project, we use Spotipy, which is a python library for the Spotify Web API, to get access to the music data that Spotify provides. We chose existing playlists that correspond to our emotion labels from Spotify beforehand. After recognizing the user's input, we obtain an emotion label of the sentence. We use the label to get a random song from the matching playlist, then play the song with Spotify on the web browser.

4. Experiments

We trained all three models with the emotional text dataset mentioned above, and computed the accuracy using the validation data.

4.1. Accuracy - Decision Tree

While giving a maximum depth of 100 to the tree, we get an accuracy of 85.2%.

4.2. Accuracy - Logistic Regression

While giving a maximum depth of 100 to the tree, we get an accuracy of 88.7%.

4.3. Accuracy - BERT Fine-Tuning

We trained the neural network layer of our fine-tuning model for three epochs, we get a final validation accuracy of 93.5%, and the least average training loss of 7%.

4.4. Song Recommendation

Because the fine-tuning model gives the best accuracy, we choose it to implement the song recommendation, we classify the input sentence into a emotion label and match it with a song took randomly from a given playlist.

5. GitHub

https://github.com/yrlin411/2022_AI-FinalProject

6. Contribution

6.1. Yong-Ching Liang 109550134 (50%)

Implementation of Decision Tree, Logistic Regression, Spotify API utilizing, slide for video

6.2. Yu-Rou Lin 109550069 (50%)

Implementation of BERT Fine-Tuning Model, 2-page report with script

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova(2019) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* *1*

[2] *Emotions dataset for NLP classification tasks:* https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp *1*

[3] https://www.educative.io/answers/what-is-sigmoid-and-its-role-in-logistic-regression

[4] https://mccormickml.com/2019/07/22/BERT-fine-tuning/advantages-of-fine-tuning