Text Emotion Classification Model

Anonymous ACL-IJCNLP submission

Abstract

This document contains Predicting future prices of Bitcoin using the Text Emotion Classification Model. The hypothesis is that if you can read the feelings written on SNS such as Twitter, you can expect the price of Bitcoin through this. Bitcoin is a virtual currency, so its value is determined by people's perception. Therefore, it is important to understand people's perception of Bitcoin.

1 Introduction

In this project we created our own model with more than one method that classify the text into 6 categories. We trained the model on an already trained data.(40K labeled data) At the end we applied the model with the Twitter data that we collected. **With the new model, you can see whether the Bitcon price will rise or fall.**

2 Collecting Data

We have created first an API twitter account where we got our consumer key, consumer secret, access token and access token secret in order to access to twitter tweets. Our search word was bitcoin and we have collected the data.

We have imported tweepy and pandas models. For the search we have used the cursor method from tweepy model. Our target tweets was just the tweets that contains the bitcoin word and that are only written in English. We have updated the data of the search everyday to avoid old tweets. At the end we have created a data frame and stored the data there. Then we have converted them to csv format, where we have labeled the tweets manually.

3 Cleaning Data

We first cleaned the tweet emotion data by removing N/A content and empty sentiment. The cleanup reduced 40,000 raw data to 3,7647.

sentiment	**frequency**
`neutral`	8638
`worry`	8459
`happiness`	5209
`sadness`	5165
`love`	3842
`surprise`	2187
`fun`	1776
`relief`	1526
`hate`	1323
`empty`	827
`enthusiasm`	759
`boredom`	179
`anger`	110

Table 1: The frequency for each sentiment.(raw data)

There were a total of 13 sentiments in the tweet emotion data(raw data) : 'empty', 'sadness', 'enthusiasm', 'neutral', 'worry', 'surprise', 'love', 'fun', 'hate', 'happiness', 'boredom', 'relief', 'anger'. The frequency for each sentiment is shown in the Table 1. **These 13 sentiments are divided back into six : 'positive', 'negative', 'worry', 'happiness', 'sadness', 'love'.** We classified 'surprise', 'fun', 'relief', and 'enthusiasm' as 'positive' and 'Hate', 'Boredom', and 'Anger' as 'negative'. We also removed 'neutral' sentiment to achieve a higher score. The frequency for each sentiment is shown in the Table 2.

Then we went through pre-processing of Twitter content through 8 Step.

Step 1. remove html and urls

Step 2. replace emojis to text

Step 3. remove non-ascii code

Step 4. remove stopwords

Step 5. remove mentions

Step 6. normalize hashtags

sentiment	frequency
`worry`	8459
`happiness`	5209
`sadness`	5165
`love`	3842
`positive`	4722
`negative`	1612

Table 2: The frequency for each sentiment.(cleaned data)

measure	score
`f1_score`	29.85207916584428
`precision_score`	31.044328567744156
`recall_score`	29.346828964538545

Table 3: Result using classifier : LinearSVC

Step 7. replace contractions

Step 8. lemmatization with spacy

We mapped the major part or the most used emojis into labels, then in the cleaning phase we replaced all the emojis in the Tweets with it's corresponding label. Putting the label between ⟨ ⟩ to distinguish it from the text, for example :-) emoji will be translated into ⟨happy⟩

Finally, the cleaned file was stored in "clean_dataset.csv"

4 Creation of 4 models

First, each sentiment was labeled with a corresponding label.

'happiness': 0, 'love': 1, 'negative': 2, 'positive': 3, 'sadness': 4, 'worry': 5

Then we split the data into train (70%) and test (30%) sets. To extract the features from the text we used several methods including tf-idf-vectorizer provided by scikit-learn, Word2Vec from spacy and also the universal-sentence-encoder-Transformer from the Tensorflow project. Then we used several neural network models to create the model including naive Bayse, LinearSVC, Multi-Layer classifier and RNN from Tensorflow. We evaluate all classifiers using the confusion matrix on the test set, getting the scores for accuracy, recall, precision and f1-score. The results are shown in the Table 3 and Table 4.

measure	score
`f1_score`	13.471369817035109
`precision_score`	17.252099402791007
`recall_score`	17.081889577594257

Table 4: Result using classifier : GaussianNB

5 Conclusion

We completed the new model, and the accuracy was approximately 70%. After using multiple models, transformation accuracy was better than the others. As a result with the new model, we can see whether the Bitcon price will rise or fall. We can analyze the sentiment of Twitter of celebrities such as Elon Musk and predict the price of Bitcoin.