Group F

Project Part 5

EECS 4404 Introduction to Machine Learning and Pattern Recognition

Course Instructor: Meiying Qin

Keren Guo, Arda Temel, Jeff Wang

2023-04-08

Abstract

This project focuses on developing a machine learning model for multi-label sentiment analysis, specifically targeting emotions such as fear, disgust, happiness, and sadness. The primary objective of the project is to build a classifier that can accurately predict the emotions associated with a given tweet. The dataset used for training and testing the model consists of tweets from Twitter, which were annotated with one or more of the aforementioned emotions. The final model achieved high accuracy and F1-score on the test data, indicating its potential for practical use in predicting sentiment in social media. The model can be utilized for a variety of applications such as sentiment analysis for marketing purposes or monitoring public sentiment during crisis events.

Introduction

About our application

Our application is a multi-class classification machine learning program that predicts human emotions from tweets. The program uses naive Bayes as the main methodology, and neural network as a comparative methodology. The input will be tweets with context related to the six basic human emotions defined by Paul Ekman: sadness, happiness, fear, anger, surprise, and disgust.

Assumptions/scope of our project

The scope of our project focused on the website Twitter and it's comments (Tweets) and will have some training from prior research done on Reddit comments. We assumed that the tweets we will be analyzing will contain emotional context related to the six basic human emotions listed above and that the tweets are written in the English language.

Importance of our project

Our application is important because it can help us better understand how people express their emotions on social media or review platforms, specifically on Twitter.

Possible scenarios for when this application can be used would be:

- When a business wants to analyze their customer sentiment, and improve their services or products based on the results.
- Can help identify people who may be suffering from mental health issues or are under emotional distress, which can lead to earlier treatment or emotional support.

Similar applications

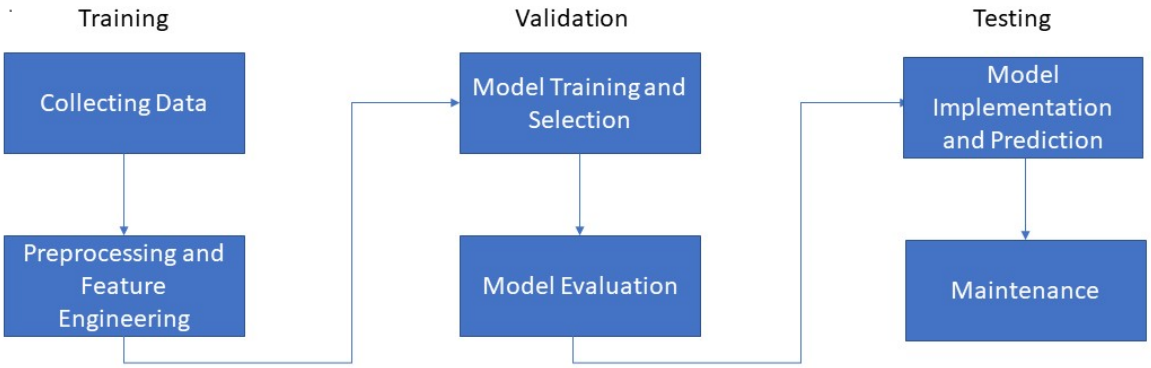
While there are similar applications to ours, such as the GoEmotions dataset and model developed by Google Research, our project differs in a few ways. Firstly, we used Naive Bayes as our main methodology while GoEmotions used a neural-network based approach. Secondly, our scope focuses on Twitter, while GoEmotions focused on comments on Reddit.

Adjustments to part 1

Initially, our proposal was to build a multi-class classification machine learning program that could predict human emotions from tweets using Naive Bayes as the main methodology. However, due to Twitter's API developer access restrictions, we had to use GoEmotion's dataset to train parts of our model as we did not have a feasible way to obtain a dataset large enough from Twitter. We manually collected tweets, but the dataset size was much smaller. Therefore, we added neural network as a comparison to Naive Bayes to provide a more comprehensive evaluation of our application's performance.

Methodology

Design/Pipeline



The design/pipeline of our project was splitted into three parts, training, validation and testing. The design is a workflow that links our data collection, model training and prediction together, and we will talk about each area in the following part.

Dataset and Preprocessing

During the training phase, we collected our target data and performed preprocessing on the raw data. Ideally, the best raw dataset would have been tweets with emotional hashtags, but due to Twitter's recent restriction of API developer accounts, we had to adopt the data used in the GoEmotions project[1]. This dataset consists of over 250,000 Reddit posts, and professionals have labeled each post into one or more emotions. We selected the posts related to the six emotions we were studying, which resulted in approximately 36,000 posts[2]. The dataset had already been partially processed, with special characters and URLs removed, and texts converted to lowercase. And after preprocessing, these posts will be our input for the project.

For preprocessing, we utilized text vectorization by using the "TfidfVectorizer" function since the raw data was partially processed. This transformed the data into numerical

values and helped to remove stop words that do not carry much meaning in our study, such as "the," "is," and "and." The data was then split into an 80/20 ratio, where 80% was used for training, and 20% was used for testing. We increased our training set from the normal 70% to 80% since this dataset was well-processed, and our final goal was to apply the model on Twitter's tweets, which required more training data to achieve better results.

Model Training

In the validation phase, we applied different machine learning techniques, tuned their hyperparameters, and compared their performance. Unlike our original plan, we applied two different techniques, naive Bayes and neural network, to compare their performances. Naive Bayes is a relatively simple statistical model, while the neural network technique has various hidden layers and nodes, making it a relatively complex one. We chose these two methods to compare their performance on our text data. We added neural networks as our second option because their performance in most visual-based studies was excellent according to the project part2's sample articles. However, our group was interested in whether neural networks would deliver similarly good results in a text-based study. Both techniques were trained using the 36,000 labeled Reddit posts. To optimize our models' hyperparameters, we focused on tuning the alpha in the Naive Bayes model and the hidden layers in the Neural Network model to determine the best parameters. In terms of coding, our team adopted the Scikit-Learn library as our main tool to code our machine learning models[3].

Prediction

The final part involved testing, where we applied the trained model to our collected Twitter data. As mentioned earlier, Twitter restricted its API access, and we were unable to collect raw data by scraping. Therefore, we manually collected 20 tweets for each emotion from Twitter, and the tweets were labeled based on their hashtags. Before we put these tweets to our prediction dataset, our team members will also evaluate whether the hashtags emotion matches the tweets' content. Our team members then cleaned the text data to match the Reddit posts dataset's format. Finally, we compared the tweet prediction results to the Reddit posts testing set result to evaluate the performance of our model.

Result

Evaluation of machine learning techniques

Before we compare the two machine learning techniques, we ensured that our models were neither overfitting nor underfitting. In the Naive Bayes model, the accuracy rate for the dev set and testing set were 0.61 and 0.60, respectively, which was relatively close and indicated no overfitting or underfitting problem. For the neural network model, the accuracy rate was 0.57 and 0.58, which was also quite similar.

To evaluate which technique was better suited for our text-based study, we used several parameters to assess their performance, including accuracy rate, precision, recall, and F-score. Accuracy is the ratio of correctly classified instances to the total number of instances in the dataset, and since the accuracy of prediction is crucial to our project goal, it was our main reference for comparing performance. In addition to accuracy, we included other parameters to evaluate our model's performance. Precision measures the proportion of positive predictions that are actually positive, with a high precision indicating a low false positive rate. Recall measures the model's ability to identify all the positive instances, with a high recall indicating a low false negative rate. The F1-score is the harmonic mean of precision and recall and is a balanced measure that takes both false positives and false negatives into account.

	Dev set	Testing set
Naïve Bayes	Accuracy: 0.61 Classification report:	Accuracy: 0.60 Classification report:

	precision	recall	f1-score
angry	0.52	0.76	0.63
disgust	0.56	0.29	0.38
fear	0.73	0.23	0.35
happy	0.62	0.81	0.66
sad	0.59	0.63	0.61
surprise	0.69	0.48	0.55
accuracy			0.61
macro avg	0.64	0.54	0.54
weighted avg	0.61	0.60	0.57

	precision	recall	f1-score
angry	0.54	0.74	0.63
disgust	0.56	0.29	0.38
fear	0.77	0.23	0.35
happy	0.62	0.84	0.71
sad	0.59	0.63	0.61
surprise	0.69	0.48	0.57
accuracy			0.60
macro avg	0.63	0.54	0.54
weighted avg	0.61	0.60	0.57

Neural Network	Accuracy: 0.57 Classification report:	Accuracy: 0.58 Classification report:

	precision	recall	f1-score
angry	0.55	0.59	0.57
disgust	0.38	0.34	0.36
fear	0.53	0.47	0.50
happy	0.73	0.68	0.71
sad	0.58	0.60	0.59
surprise	0.56	0.62	0.59
accuracy			0.57
macro avg	0.55	0.55	0.55
weighted avg	0.57	0.57	0.57

	precision	recall	f1-score
angry	0.56	0.59	0.58
disgust	0.41	0.34	0.37
fear	0.51	0.49	0.50
happy	0.74	0.70	0.72
sad	0.59	0.61	0.60
surprise	0.56	0.63	0.59
accuracy			0.58
macro avg	0.56	0.56	0.56
weighted avg	0.58	0.58	0.58

The summary chart above shows our results on the Reddit posts dataset. We observed that the accuracy of Naive Bayes and Neural Network for their testing sets were very close to each other, with both close to 0.6. However, the accuracy rate of Naive Bayes was slightly better than that of the Neural Network.

Regarding precision, recall, and F1-score, both fear and disgust had a relatively lower rate under the Naive Bayes model. However, the neural network model had better performance in predicting fear, as its F1-score was significantly higher than Naive Bayes'. Although the neural network model performed better in a specific emotion category, the Naive Bayes model still had overall better performance.

Results of prediction on tweets

In this case, we decided to use the Naive Bayes model to predict our tweets dataset and verify if the model performed as intended. Below are the results of the prediction on our manually collected tweets data:

```
Accuracy: 0.49
Classification report:
              precision    recall  f1-score

       angry       0.32      0.50      0.39
     disgust       0.89      0.40      0.55
        fear       1.00      0.05      0.09
       happy       0.42      0.90      0.57
         sad       0.56      0.70      0.62
    surprise       0.67      0.40      0.50

    accuracy                           0.49
   macro avg       0.64      0.49      0.45
weighted avg       0.65      0.49      0.45
```

As we applied the model trained by the data from another platform (Reddit) to a new platform (Twitter), we anticipated that the accuracy rate would not be as high as the training set. However, the decrease in the accuracy rate was not substantial either.

Discussion

Implications

Based on our results, it appears that Naive Bayes outperformed the neural network in both evaluations in terms of accuracy: reddit (0.6 vs 0.58) and tweets (0.49 vs 0.48). However, the lower accuracy scores for our tweets dataset indicate that there is room for improvement. One limitation of our study was the size of our manually collected tweet dataset, which may have affected the accuracy of our model. In the future, we could aim to collect a larger dataset of labeled tweets, either by gaining developer access to Twitter's API or by collecting tweets manually. Additionally, analyzing the misclassifications made by our models could help us identify patterns and areas for improvement.

Strengths

One of the main strengths of our design was including multiple models (Naive Bayes and neural network) to compare their performance and evaluating these techniques. Another strength would be our adaptability in addressing our limitations, such as manually collecting tweets for our dataset when Twitter restricted developer access to their API. Another strength in our design was that we used stratified sampling for each emotion category by equally representing them in our dataset. This helped mitigate risk of bias in the model.

Limitations

As stated above, one of our limitations include the size of our manually collected tweet dataset leading to some inaccuracy in our model caused by Twitter restricting their developer access to their API. Another limitation could be the amount of emotions we could have tried to classify, as related research of emotion classification of posts/comments such as GoEmotions had 27 emotion categories. It is also worth noting that emotion classification is a subjective task, as different annotators may label the same text differently, so the model may struggle with sarcasm and irony.

Future Directions

Moving forward, if we were to continue working on this project, we would aim to collect a larger dataset of labeled tweets either by hopefully getting developer access to Twitter’s API or manually. As stated before, we could probably incorporate more emotion categories as well. We could also explore other machine learning models and techniques and to test if their performance is better. We should also expand our prediction set to improve the accuracy of our model.

Additional Questions

Useful feedback from the peer evaluation

We found the feedback from Group 1 and Group 4 regarding the future directions of our project to be useful. They suggested that we should expand our prediction set to improve the accuracy of our model. Group 2 provided valuable feedback on our evaluation process, suggesting that we should include more details and explanations, which we have taken into consideration.

Changes made based on the feedback

According to the Group 2 review, the reviewer believes that it would be helpful to add more information in model prediction and training. So we have added the evaluation of overfitting and underfitting and talked about what hyperparameters we are using to tune the model.

Reference

[1] Demszky, Dora, Yulia Movshovitz-Attias, Anh Tran Koenecke, Raksha Trivedi, Kshitijh Talamadupula, and Srinivasan Ravi. "GoEmotions: A Dataset of Fine-Grained Emotions." arXiv preprint arXiv:1910.04612 (2019).

[2] "GoEmotions Full Dataset." GitHub repository, https://github.com/google-research/google-research/tree/master/goemotions/data/full_dataset (accessed April 8, 2023)

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.