projectReport

February 28, 2023

1 Project Title: "Classifying Emotion in Text-Based Content"

Auther: Hicran Arnold

1.1 Abstract

Sentiment analysis is a popular area of natural language processing (NLP) that aims to classify text into different categories based on the underlying emotions and opinions expressed within the text. In this project, we explore the use of deep learning models for multi-class sentiment analysis on a large dataset of English twitters. We compare multiple different machine learning models and propose an optimized model that achieves high accuracy on the task

1.2 Introduction

The exponential growth of digital communication has resulted in a massive amount of text-based content that expresses a wide range of emotions. Understanding the emotions convey in this content is essential for businesses, organizations, and individuals to understand their audiences better and gauge public sentiment about their products and services. However, manual analysis of these texts is time-consuming and impractical, making it necessary to develop automated methods for emotion classification.

This project focused to develop a machine learning model for emotion classification in text-based content, which will provide a more efficient and accurate method for analyzing the emotions conveyed in digital communication.

Data Preparation ***Data Collection***: The data set is English Twitter messages with six basic emotions: anger, fear, joy, love, sadness, and surprise. The data includes training, test , and validation text files. The training data set has 16000, and validation and test txt files have 2000 lines of text each. The entire data set is 738 kB. The data set is publicly available and can be found at PRAVEEN-Kaggle and dair-ai/emotion_dataset.(1)

For that prep processing I used below techniques:

1. Formatting the data : The data was combination of twit and emotion (sad, angry,etc). So I converted data to a data frame, first column is the twits and the other part is the emotion. Then I created another column in data frame using pyspark StringIndexer method to convert to label to a numeric column Our emotion labels are : joy, anger, sadness, fear, love, surprize
2. Data Cleaning Process: The process that I follow is removing the tags, none letters, remove words that less than 1 or characters

3. Lemmatize_text: This process expensive to compute since we have to check each word and stem but I wanted to follow this process to see how it effects the accuracy of the models.
4. Removing Stop Words: In this step we want to get rid of the words that does not effect our prediction
5. Tokenizing : We need the number version of the words in the twit
6. Countvectorizer : Counting each words to see how many times it repeated in the twit
7. Idf: This shows us how important a word for that twit
8. ngram: I want to check to see if the context matter, I used ngram=2
9. hashingTF:This is very similar countvectorizer but computationally more effective
10. ChiSqSelector: I want to check to see if less amount of words we still be able to get high accuracy

1.3 Methodology

I used three machine learning models , support vector machine, logistic regression , and naive bayes algorithm and I created four data models to use and analyze accuracy in these three machine learning model. First data model is on the cleaned that I simply used countvectorizer plus idf, the second model is the same pipeline but I added chisquare selector and third one unclean data so it means that no removing tags, small words extra , I used unclean data , countvectorizer and idd. The last one is ngram, hashingtf and idf. I cleaned the data and selected the features and later I created 12 machine learning models with these four data model.

1.4 Results

1. We received a high accuracy result. The final model f1 accuracy is 85% with 86% precision and recall.

2. A comparison of the performance of different text preprocessing techniques on the accuracy of sentiment analysis models, with findings that demonstrate the effectiveness of certain techniques in improving accuracy. Different text preprocessing techniques yield different result. Surprisingly cleaning the data did not specifically improved the dta instead we received better accuracy on the not cleaned text data.
3. In this research we saw that feature selection with chisquare we received better performance than the data model where we did not limit the feature size. This showss that selecting important features improve accuracy in sentiment analysis models. Our experimental results showed that the SVM model achieved the best performance on the task of multi-class sentiment analysis. The SVM model achieved an F1 score of 85% and 86, recall of 87.5%, and and accuracy of 86.2% on the test dataset.

1.5 Conclusion

In this project, we explored the effect of the different feature selection and cleaning for multi-class sentiment analysis. We compared the performance of several machine learning models and we tuned our final model for best accuracy. Our experimental results showed that the SVM model did better comparing the other models on the task of multi-class sentiment analysis. Our proposed model

can be used for sentiment analysis in various applications, such as customer feedback analysis and social media monitoring.