Multiclass Emotion Analysis Using NLP

Rohan Madhani[1], Sagar Makwana[2], Viral Lakhani[3], Alabh Mehta[4], Sindhu Nair[5]

[1,2,3,4] Student, Computer Department, D. J. Sanghvi College of Engineering, Vile Parle, Mumbai, India

[5]Assistant Professor, Computer Department, D. J. Sanghvi College of Engineering, Vile Parle, Mumbai, India

Abstract: ***In the present scenario, sentiment analysis has become a popular topic in the field of Machine Learning (ML) and Natural Language Processing (NLP). Sentiment analysis is the systematic process of determining the sentimental tone in a array of words. It helps to understand the emotion, attitudes, and opinions expressed in the sentence. Machine learning techniques are widely used in determining the emotions from texts due to their precise prediction. Various classifiers can be used for performing sentiment analysis which may provide different accuracy. This paper documents a comparative study of three machine learning classifiers namely, Support Vector Machine (SVM), Recurrent Neural Network (RNN)- Long Short Term Memory (LSTM) and Naive Bayes for performing seven class sentiment analysis. The emotions which were considered for this study were: 'joy', 'sadness', 'anger', 'shame', 'guilt', 'disgust' and 'fear'. For analyzing the performance of these models precision, recall and F1 score were used. From the result, we come to know that the performance of a Recurrent Neural Network was much better than other classifiers***.

Keywords: Sentiment Analysis, Machine Learning, Support Vector Machine, Recurrent Neural Network, Naive Bayes

1. Introduction

1.1 Introduction to Sentiment Analysis

Sentiment analysis, also known as Opinion Mining, has attracted great attention. It refers to the process of systematically identifying the sentiment expressed in the sentence. Generally speaking, sentiment analysis aims to determine the attitude of a author with respect to some topic or the overall contextual polarity or emotional reaction to a document.

1.2 Sentiment analysis using machine learning

Sentiment analysis would heavily rely on techniques of "Natural Language Processing" in extracting significant patterns and features from the large data set of text and on "Machine Learning" techniques for accurately classifying individual unlabelled data samples. Sentiment analysis requires variety of angles that needs to be taken into account to get a correct result. For this study, we performed the sentiment analysis on human text. Specifically, we wanted to classify text into seven emotions, building a classifier that would output the emotion that best describes the author's mindset in writing the text: 'joy', 'sadness', 'anger', 'shame', 'guilt', 'disgust' and 'fear. For this task, we built and optimized the following models: Support Vector Machines, Recurrent Neural Network-Long Short Term Memory (LSTM) and Naive Bayes. The aim of this model is to accurately and automatically classify sentiment of an unknown text stream. Our selection of models was guided by the literature review. Sentiment analysis has many potential applications; for example, it could be used to track psychological health and general well-being of the individual, or even detect the risk of suicide. It can also be used by firms to find out the response of their product in market. Also, it can be used for predicting political elections and predicting socioeconomic phenomenon like stock exchange. Emotional analysis model could also be applied to short form text, like Flipkart or Amazon reviews, which could provide specific feedback and insights to merchants.

2. Literature Review

Sentiment analysis methods are classified into two types: Lexicon based and machine learning based.

a) Lexicon based methods:
This method involves calculating the sentiment from the semantic orientation of word or phrases that occur in a text. With this approach, a dictionary of positive and negative words is required, with a positive or negative sentiment value assigned to each of the words. The steps in sentiment analysis are:

1) Preprocess the text and remove redundant information.
2) Initialize the overall sentiment score (s) to 0.
3) Tokenize the text. Go through each token in the sentence and match it with the entries in the dictionary. If a match is found:
   a) If the associated sentiment is positive, increment s.
   b) Else if it is negative, decrements.
4) After the entire statement has been parsed. Compare the value of s to a predefined threshold value (t). If s is greater than t, overall sentiment is positive. If it is less than t, the overall sentiment is negative.

b) Learning based methods:
Researchers have developed various machine learning models. The complexity of models and methods used for sentiment analysis have significantly changed over the past few years. The earlier works used linear models like Linear Regression and Support Vector Machines while the latest works are using non-linear vector space models like neural networks. Support Vector Machine (SVM), Recurrent Neural Network (RNN) and Naive Bayes are widely used for performing sentiment analysis. Each classifier provides different accuracy based on the parameters used to classify. Most machine learning methods use Supervised learning to classify the text and training, as well as a test set, are needed.

3. Proposed Model

3.1 Architecture

There are several classifications techniques which can be used for performing sentiment analysis. We aim at using three machine learning classification techniques, namely, SVM, RNN-LSTM and Naive Bayes in order to predict the precise emotion.

1. Support Vector Machine:
Support vector machine analyses the data, define the decision boundaries and uses the kernels for computation which are performed in input space. The input data are two sets of vectors of size each. Then every data represented as a vector is classified in a particular class. Now the next step is to find a hyperplane that divides the document as per the sentiment, and margin between these classes should be as high as possible. The objective is to find the hypothesis h for which the error is lowest. If we symbolize the hyperplane as h and text as t, and represent the classes into which the text has to be classified as gr {1, -1}corresponding to the sentiment of the text, the solution can be written as:

$$H= \sum aiCiti$$

The texts that have a>0 are the ones which contribute in finding the hyperplane. As long as the text classification is linearly separable, SVM does not assume a feature to be irrelevant, which sometimes leads to a loss of information. SVMs are inherently two-class classifiers. The traditional way to do multiclass classification with SVMs is to use one vs all approach. The most basic and widely used technique in practice is that of building of one-versus-rest-classifiers, commonly referred to as one-versus-all or OVA classification, and selecting a class which classifies the test datum with the highest margin.

2. Recurrent Neural Network- Long Short Term Memory:
As a recurrent neural network model, we employ a one hidden-layer bi-directional LSTM (biLSTM), trained on seven-class sentiment prediction of phrases and sentences. This model takes as input a sequence of words x1, x2, ..., xT (as well as this sequence in reversed order), where each word is represented by a word embedding of some appropriate dimension. In our experiments, we use as input the tokenized sentences, pre-processing them by lowercasing. On seven-class sentiment prediction of full sentences the model achieves a bit less accuracy as compared to binary classification (positive vs. negative, ignoring neutral sentences). Given a trained neural network that models a scalar-valued prediction function fc (also referred to as a prediction score) for each class c of a classification problem, and given an input vector x, we are interested in computing for each input dimension d of x a relevance score Rd quantifying the relevance of xd w.r.t to a considered target class of interest c. In other words, we want to analyze which features of x are important for the classifier's decision toward or against a class c. For estimating the pertinence of a pool of input space dimension or the variables (e.g. in NLP, where we use distributed word embeddings as input, it is essential to compute the relevance of word and not only the single vector dimensions), we simply sum up the relevance scores Rd of its constituting dimensions 'd'.

3. Naive Bayes' Classification:
One of the most popular supervised classification paradigm are the Bayesian network classifiers. The Naive Bayes' classifier a renowned network classifier which forms the basis on the Bayes' theorem is a probabilistic classifier which also considers the aspects of Naive independence assumptions. It remains as one of the most popular method of the text categorizing, the one that involves problem of judging documents as belonging to one category or the other also involving the word frequencies as an added feature. The most important aspect of the Naive Bayes' is that it requires petty amount of data required to train the parameters essential for the sentiments classification. Naive Bayes' considered to be a conditional probability model, though simple but with strong assumptions, it has satisfactorily provided results and been successful in large number of domains. In Naive Bayes' the observed data and prior data can be combined and it also bolsters by providing practical learning algorithms. The base of the Naive Bayes' technique is to find the intended probabilities of the categories wherein the text document is provided and the method to do so is by that of the joint probabilities of words and categories. Word independence forms the basis of the assumption of Naive Bayes' technique. The initiating point is the Bayes' theorem for conditional probability, stating that, for a given data point x and class C:

$$P(C/x)=P(x/C)/P(x) \quad \text{....................}(1)$$

Furthermore, by making the assumption that for a data point x = {x1,x2,...xj}, the probability of each of its attributes occurring in a given class is independent of one another, the probability of x can be estimated as below:

$$P(C/x)=P(C).\prod P(xi/C) \quad \text{..................} (2)$$

3.2 Performance Evaluation Criteria

The evaluation of the performance of the various models is done using 3 different measures namely Precision, recall and F1 Score.

- Precision stands for the ability of the classifier not to label a negative sample as positive. It can be thought as measure of classifiers exactness.
- Recall stands for the ability of theclassifier to find all the positive samples. It can be thought of as a measure of classifiers completeness
- The harmonic mean of the precision and recall gives the F1 Score. It conveys the balance between the precision and recall

The formula for F1 score is:

$$F1 = 2 * \frac{Precision * Recall}{(Precision + Recall)}$$

4. Performance Analysis

The dataset that we used to test our models was taken from ISEAR(International Survey on Emotion Antecedents and Reactions). It contained a table having two columns: sentence and its emotion. The dataset contained around 7500 rows.

We ran our three classification algorithms on the dataset and calculated the precision, recall and F1 score on the different emotions. The results from the models are given below. A good classifier must have a high precision and high recall altogether.

1. **RNN-LSTM**

Table 4.1

Emotion	Precision	Recall	F1Score
Anger	0.54	0.54	0.54
Disgust	0.71	0.62	0.66
Fear	0.64	0.75	0.69
Guilt	0.53	0.58	0.56
Joy	0.78	0.85	0.82
Sadness	0.66	0.71	0.68
Shame	0.66	0.48	0.56
Avg/Total	**0.65**	**0.65**	**0.64**

2. **Naïve Bayes**

Table 4.2

Emotion	Precision	Recall	F1Score
Anger	0.45	0.48	0.46
Disgust	0.66	0.54	0.59
Fear	0.62	0.67	0.64
Guilt	0.47	0.48	0.48
Joy	0.66	0.67	0.67
Sadness	0.60	0.56	0.58
Shame	0.46	0.48	0.47
Avg/Total	**0.56**	**0.56**	**0.56**

3. **SVM (kernel=linear)**

Table 4.3

Emotion	Precision	Recall	F1 Score
Anger	0.43	0.54	0.48
Disgust	0.61	0.56	0.59
Fear	0.66	0.64	0.65
Guilt	0.48	0.50	0.49
Joy	0.71	0.67	0.69
Sadness	0.62	0.52	0.57
Shame	0.44	0.45	0.44
Avg/Total	**0.57**	**0.56**	**0.56**

On checking the performance for each algorithm carefully, the RNN-LSTM Model has much better performance as compared to other models

5. Conclusion

While it's difficult to speculate how a relatively immature system discussed in the paper might evolve in the future, there is a general assumption that sentiment analysis needs to move beyond a one-dimensional positive to negative scale. This sentiment analysis can be extended to a multi faceted approach to improve accuracy. Here the main aim was also to improve and also evaluate the performance for the classification in terms of precision, recall and F1 score. Here we compared three supervised learning techniques such as Support Vector Machine (SVM), Naive Bayes and the Recurrent Neural Network (RNN). The experimental results depicted that the classifiers yielded better results for Recurrent Neural Network (RNN) approach giving much higher accuracies than the other two approaches. Thus we can say that we can use Recurrent Neural Network (RNN) to successfully predict the emotions of the text.

References

[1] Dave, Kushal, Steve Lawrence, and David M. Pennock. ”Mining the peanut gallery: Opinion extraction and semantic classification of product reviews.” Proceedings of the 12th international conference on World Wide Web. ACM, 2003.

[2] Mcauliffe, Jon D., and David M. Blei. ”Supervised topic models.” Advances in neural information processing systems. 2008.

[3] Pang, Bo, and Lillian Lee. ”Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales.” Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005.

[4] Socher, Richard, et al. ”Recursive deep models for semantic compositionality over a sentiment treebank.” Proceedings of the conference on empirical methods in natural language processing (EMNLP). Vol. 1631.