

Capstone Project

Machine Learning Engineer Nanodegree

Husnuye Yasar

February 22th, 2019

A comparative study of Sentiment Analysis using various ML algorithms

Domain Background

With the democratization of the Internet every individual now has an opportunity to express his or her opinion on any topic, event, company, indeed anything at all. They have an opportunity to express themselves through pictures, words, tweets, posts and numerous other ways.

The challenge then lies in assessing the emotions, feelings, passion, sentiments being conveyed by these individuals or the population as a whole. Given the large volumes of data, the only possible way to really assess these sentiments is computationally. Sentiment analysis is this process of examining the data and drawing conclusions of what the emotions are of the population on any given topic.

This project is going to analyze the sentiments of movie reviews from a Rotten Tomatoes dataset using various Machine Learning and Natural Language Processing algorithms.

Problem Statement

The goal of this project is to implement and train various ML and NLP algorithms to predict the sentiment on a Rotten Tomatoes movie reviews dataset provided by Kaggle [1] and compare them against a preset metrics to figure out which provides the best accuracy. For each algorithm the data would have to be preprocessed,

split into training and validation sets, the classifier would then be trained on the training set and the model would then be tested on the validation set.

This effort aspires to defining a model that is most optimal for datasets and problem statements that this project tackles. The hope is that future users will have to contend with fewer choices and possibly improve upon the work done here.

Datasets and Inputs

The dataset for this project is a Rotten Tomatoes movie review dataset provided by Kaggle. It is a corpus of movie reviews used for sentiment analysis, and is in a tab separated file 'train.tsv'. 3

A sample data snapshot looks like –

	Phraseld	Sentenceld	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage ...	1
1	2	1	A series of escapades demonstrating the adage ...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2

There are four columns in the data –

1. Phraseld – The id of each Phrase. This acts like the table index.
2. Sentenceld – The id of each Sentence. Each Sentence has been parsed into many phrases. Phrases that are repeated (such as short/common words) are only included once in the data.
3. Phrase – Collection of words that expresses the Sentiment. They all come from the Rotten Tomatoes dataset.
4. Sentiment – The actual sentiment value, the label or target variable.

The phrases are labelled on a scale of five values: negative, somewhat negative, neutral, somewhat positive and positive.

Each phrase has a sentiment label associated with it as follows:

0 - negative

1 - somewhat negative

2 - neutral

3 - somewhat positive

4 – positive

Total number of Phrases – 156060

Total number of Sentences – 85294

Phrases Count by label:

Phrase	Count
Negative	7072
Somewhat Negative	27273
Neutral	79582
Somewhat Positive	32927
Positive	9206

For the project the data will be split into training and validation set by an 75-25 ratio.

Benchmark Model

The Benchmark model will be generated using CountVectorizer from sklearn package to create a Bag of Words model which will then be trained using a Naive Bayes classifier on the given training data. The predictions will then be made on the validation data and results tested using the accuracy score and the F1 score

Evaluation Metrics

A couple of Metrics will be used for Evaluation -

The goal is to reach a high *accuracy score* in predicting the correct sentiment of a movie review that was written on Rotten Tomatoes. The simplest definition of an accuracy score is the proportion of correctly classified objects. Since the higher the number of correctly classified objects, the better the model, accuracy score is the metrics that will be used to find the most suitable algorithm for the project.

Since this is an unbalanced multi-class classification problem, it is better to have more than one metric for evaluation and the project will also use *F1 score* to compare the various models. For neural networks, Keras will be used and it no longer support F1 score in its metrics.

Project Design

The following steps will be followed in the project –

1. Data Analysis and Cleansing: In the first step the data and its various features will be thoroughly examined. The sentiment spread will be studied and analyzed. The different phrases will be cleaned by tokenizing, removing the stop words, punctuation and uppercase letters and then Vectorizing the results. Experimenting will be done with CountVectorizer, TfidfVectorizer, and features needed for Word2Vec vectors. The data will be split into train and test set.
2. Train the Benchmark Naïve Bayes classifier and produce the benchmark results.
3. Train an SVM classifier and a RandomForest classifier and test the performance.
4. Set up a Convolutional Neural Network and train the model. Try fine tuning the CNN by adding, dropping layers to get the best possible accuracy results.
5. Setup an RNN using an LSTM layer and train the model. Try fine tuning the model to get the best possible accuracy.
6. Setup a Word2Vec model and calculate the accuracy.
7. Compare and contrast the accuracy and F1 score of all the models to draw conclusions on which model is the best suited for Sentiment Analysis.

Resources

- [1] <https://www.kaggle.com/c/movie-review-sentiment-analysis-kernels-only/data>
- [2] https://en.wikipedia.org/wiki/Bag-of-words_model
- [3] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
- [4] https://en.wikipedia.org/wiki/Logistic_regression
- [5] https://en.wikipedia.org/wiki/Support_vector_machine
- [6] <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>
- [7] <https://fasttext.cc/docs/en/english-vectors.html>
- [8] <http://konukoi.com/blog/2018/02/19/twitter-sentiment-analysis-using-combined-lstm-cnn-models/>