Sentiment Analysis with Perceptron algorithm

Manuel Cecere Palazzo

2020

Contents

1 Introduction

In this project it will be used an implementation of the Perceptron algorithm to operate sentiment analysis, the process of classification of opinions expressed in a text in order to identify whether the attitude of the author is positive, negative or neutral. The aim is to replicate the results of Gräßer et al. 2018, the analysis of drugs reviews to determine the patient overall satisfaction for the in-domain case. The dataset is available here.

The Perceptron's implementation used is available from scikit-learn, an open-source Python library.

2 Approaches

To use the scikit library at its best, we used pandas, an open-source library that provides data structures and data analysis tools. We've first used it to convert the dataset to a csv format, easier to work with, and than to load the data in the train set and the test set.

Following the Gräßer et al. 2018's example we derived the three level polarity labels for overall patient satisfaction using the same criteria from the article (tab 1) .

In order to perform machine learning on the reviews, the text content was turned into numerical feature vectors using a bag-of-words representation. To

count the occurences of words we used the CountVectorizer class. The Tfidf-Transformer divides the number of occurrences of each word in a document by the total number of words in it, avoiding the issue of longer reviews having higher average count values.

The task of searching the best hyper-parameters for these classes to improve the accuracy score was done using the gridSearchCV class, that exhaustively considers all parameter combinations given, using 5-fold cross validation to avoid overfitting.

Rating	Label
$rating \leq 4$	-1
$4 < rating < 7$	0
$rating \geq 7$	1

Table 1: Deriving the three level polarity lables

3 Experiments and Results

Not one of the possible combinations of the Perceptron parameters stood out for significant gains in the accuray scores. However, that changes when we try altering CountVectorizer's parameters. The inclusion of word bigram and trigram features shows big improvements, as seen in tab 2 .

Ngram features	Accuracy
ngram range(1,1)	76.7
ngram range(1,2)	86.8
ngram range(1,3)	92.3

Table 2: Accuracy scores of each combination

4 Conclusions

The goal of the project was to replicate the results of the Gräßer et al. 2018's experiment. That goal was achieved. Though the algorithm used was different, since in the article it was used Logistic Regression, the accuracy scores were almost the same.

Looking at the scikit tutorial page about choosing the right estimator, it's interesting to observe that the best estimator suggested for the task is the SGD classifier that can be implemented using either the Perceptron or Logistic Regression, confirming both of them as good performers of the task.

5 References

Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning. Gräßer et al. 2018

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.