Sentiment Analysis

on IMdb dataset

Name: Minjie Xu
Student Number: c1898688
Data: 2020.1.13

Content

Introduction

The sentiment analysis dataset provided including positive and negative movie reviews, including training development and test splits is provided to be analyzed by steps of preprocessing the data, selecting a feature and training a machine learning model. Firstly, these methods used in my project are introduced in detail. In the last part of this report, the table of performance is provided to compare different features and models on aspects of efficiency and accuracy and find the best solution.

Steps of the process

Preprocessing

Remove stop words

Useless words used commonly are called stop words, which should be filtered out to avoid them being construed as a signal for prediction. The parameter “stop_words” is used to make sure that the stop word list has had the same preprocessing and tokenization applied as the one used in the vectorizer (Scikit-learn.org, 2020).

Standardization

Standardization of a dataset is a common requirement for many machine

learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data. Fig.2 shows the accuracy is improved after standardization, but the efficiency of the training model is reduced.

```
start fit!
fit time: 12072.455430269241
start predict!
Fold completed.
0.849
```

(a)

```
start fit!
fit time: 328.6024308204651
start predict!
Fold completed.
0.725
```

(b)

Fig.2 (a) The accuracy and time used after standardization (b) The accuracy and time used

Sentences tokenization

A stream of text is broken into sentences tokenized. Using the string split() method directly is more efficient.

Choice of features and Vectorization

Unigram

Each token occurrence frequency is treated as a feature, then a collection of text documents is transformed into numerical feature vectors as input

data. This method is also called the Bag of Words based on word frequency while ignoring the relative position information.

N-grams

Bag of words model does not capture phrases and multi-word expressions, effectively disregarding any word order dependence. Thus, N-grams as features are considered. Bigrams, 1-to-2-gram,1-to-3 gram are chosen as features and compared with each other. Table 1 shows that the accuracy of 1-to-2-gram is similar to that of 1-to-3 grams. Also, Bigrams has the lowest accuracy. Thus, 1-to-3 gram and Bigram are abandoned in the feature selection experiment to save time.

TF-IDF

TF-IDF weighting scheme (term-frequency times inverse document-frequency) is a standard statistic method that re-weight the count features into floating-point values to filter out common words and retain important words.

Two methods are used to vectorize the dataset:

1.Use CountVectorizer() combined with TfidfVectorizer()

2.Use TfidfVectorizer() directly.

The performances of these two methods are the same. The latter is recommended due to simplicity(Cnblogs.com, 2019)

Punctuation frequency and Sentence length

Punctuation frequency and sentence length are also considered as features. However, these two features are abandoned because of low accuracy (around 0.5).

Feature selection

Feature selection is the process of finding a better subset of features to get the highest accuracy and efficiency for the training model. This method benefits the performance of the model, reducing underfitting and overfitting. Chi-square, which is based on statistical tests, is carried out to select features. The developing dataset is used to find the best number of features for training.

Training and test

The training dataset is used as input data to train the model. Many different models are used to be trained and result in different performances. These models are as follows:

- SVC() with linear, RBF and polynomial kernels
- LinearRegression()
- RandomForestClassifier()
- VotingClassifier()

They are compared in another assay then used to predict the results of the test dataset.

Justification

Improvement performance with data:

The development split is used to test if “StandardScaler()” can improve the accuracy of models or the efficiency of training models. “StandardScaler()” used for the low number of features can improve the efficiency of training models, while it does not work well for the high number of features.

The development split is also used in chi-square to select the different number of features, controlling overfitting. Table 1 shows the accuracy of SVC models with a different number of features. Finally, the effect is optimal when the feature dimension is 1000~1500.

Improve performance with ensembles:

The development split is used to train models by mixed feature vectors and mixed models (Brownlee,2016). Fig.1 shows the results for mixing feature vectors of “TF-IDF" and feature vectors of "Unigram". Obviously, it can improve accuracy but reduce efficiency.

```
(0.8662534986005598, 0.8664836765417986, 0.8662585131894485, 0.8662336047285681)
('time is :', 498.5199270248413)
```

Fig.1. The accuracy of the SVC on mixed features and the time used.

Voting Classifier combines multiple different models into a single model. Combinations of Random forest classifier and different kinds of SVC models are carried out. As table 1 shows, this model performance well by unigrams as features. Obviously, it has low efficiency.

Performance

Evaluation of performance considers these aspects: the scores of precision, recall, f-measure and accuracy. Also, efficiency should be considered. Table 1, as follows, gives data on the performance of the SVC(kernel="linear") model with different features:

Table 1. Accuracy and time used records of SVC (linear)

Model/Feature (accuracy/time)	TF-IDF	Bigram	Unigram	1-to-2 grams	1-to-3 grams
500 features					
SVC(linear)	0.845/109.7s	0.727/283.54s	0.837/5063.7s	0.837/298.4s	0.837/296.5s
VotingClassifier	0.806/363.05s				
1000 features					
SVC(linear)	0.852/187.23s	0.764/288.3s	0.865/349.3s	0.863/379.05s	0.86/384.6s
VotingClassifier	0.815/671s	0.751/769s	0.854/634s		
5000 features					
SVC(linear)	0.876/1004.4s		0.84/908.3s	0.846/969.5s	
VotingClassifier	0.806/6644s		0.850/2794s		

Improvement

1.TF-IDF just take term frequency into account, ignoring the position of words. Adding weights artificially on the words in the first and last paragraphs can be considered as a possible way to improve accuracy.

2. In this project, models use default parameters to fit datasets. Grid search and Learning Curve are two ways to find the best parameter for each model to improve accuracy.

3. In this project, features are based on English words. However, emoticons that appear in sentences posted communicate the emotions of the writer more efficiently. Plutchik's emotion model can transform these emoticons into the emotional vector as a vital part of feature vectors to train the model. (Sho Aoki and Osamu Uchida, 2011)

Conclusion

In conclusion, Bigrams has the lowest accuracy as features, while the accuracy of 1-to-2 grams, unigrams are similar to that to 1-to-3 grams. In the aspect of feature dimension, the SVC model with TF-IDF features can get higher accuracy as more features.

Reference

Scikit-learn.org. (2020). 6.2. Feature extraction — scikit-learn 0.22.1 documentation. [online] Available at: https://scikit-learn.org/stable/modules/feature_extraction.html#stop-words [Accessed

7 Jan. 2020].
Cnblogs.com. (2019). TF-IDF formula and TfidfVectorizer in sklearn-Clownszz-Blog Park. [online] Available at: https://www.cnblogs.com/Rvin/p/10695477.html [Accessed 7 Jan. 2020].
Brownlee, J. (2016). How To Get Better Machine Learning Performance. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/machine-learning-performance-improvement-cheat-sheet/ [Accessed 10 Jan. 2020].
Sho Aoki and Osamu Uchida. (2011). A method for automatically generating the emotional vectors of emoticons using weblog articles. ACACOS'11, pages 132--136, Stevens Point, Wisconsin, USA