2-Step Sentiment Analysis for Movie Reviews

Samuel Bortolin (221245)
University of Trento
Via Sommarive, 9, 38123 Povo, Trento TN
samuel.bortolin@studenti.unitn.it

Abstract

Sentiment analysis is an important topic in the Natural Language Understanding field of research. This work will deal with sentiment-polarity classification, a well-known task that tries to judge if a human-written text is saying something positive or negative about a certain topic. This will be done also dealing with a related task, subjectivity/objectivity detection. The target text of the work will be the movie reviews collected from Internet Movie Database (Maas et al., 2011). This report explains the realization of some neural networks able to perform subjectivity/objectivity detection and sentiment-polarity classification tasks. In the end, it compares also the results obtained with some baselines and shows some improvements over them.

1 Introduction

The goal of the work is to perform sentiment analysis on movie reviews with the objective of classify them into positive and negative. The identifier of the project is SA2 and consists in implementing a 2-step classification:

- sentence-level subjectivity detection;
- aggregate into document-level sentiment-polarity.

This will be done using the well-known movie reviews dataset (Maas et al., 2011) and using as evaluation measure the F1 score. The report is structured as follows: the next section defines the problem statement and how the two steps are connected in order to get the final classification. Then, the following sections analyze the two datasets and describe the models used. Ultimately, the last two sections discuss the results and the conclusions of the work.

2 Problem Statement

Given a review, the objective is to detect if this review has a positive or negative sentiment. This is done with a huge preprocessing of the reviews that aims to have a simpler text with only subjective sentences, from which is easier to predict the correct sentiment-polarity since there is less text and only the relevant part for this task. This requires an objectivity classifier and subsequently a sentiment-polarity classifier.

3 Data Analysis

Two datasets were used, both on movie reviews. The first one is the Rotten_IMDB subjectivity dataset (Pang & Lee, 2004), it contains two files:

- plot.tok.gt9.5000 that contains 5000 objective sentences from plot summaries available from the Internet Movie Database;
- quote.tok.gt9.5000 that contains 5000 subjective sentences that are movie review snippets taken from Rotten Tomatoes.

I used this data to train and test the performance of an objectivity classifier.

The other dataset is a large dataset of informal movie reviews from the IMDB (Maas et al., 2011) that contains two folders: train and test. Each folder is structured with two subfolders:

- neg that contains 12500 negative reviews;
- pos that contains 12500 positive reviews.

This huge dataset for a total of 50000 reviews is used to train and test the performance of a sentiment-polarity classifier.

4 Models and Results

In this section, it is explained how some simple baselines using some methods available in Python libraries are developed. Then it is illustrated the logic behind the developed models and which are the results obtained with these approaches.

4.1 Baseline using a Count Vectorizer and a Support Vector Classifier for document-level sentiment-polarity classification

I tried as first baseline using a Count Vectorizer to transform the reviews from the IMDB dataset in vectors and a Support Vector Classifier to obtain the labels. I train it using the train data directly without any kind of pre-processing and I test it with the test data obtaining a good **0.862** of F1 score.

4.2 Baseline using VADER for document-level sentiment-polarity classification

I build a baseline using the VADER module of NLTK, in particular I used the Sentiment Intensity Analyzer for document-level sentiment-polarity classification. I passed directly to it the entire review and I assigned the label according to the highest returned polarity score. This simple approach without training allow to get an overall F1 score of **0.684** (0.630 for the negative class and 0.738 for the positive one).

4.3 Baseline using VADER for sentence-level sentiment-polarity classification and using objectivity remotion

Starting from the previous baseline, I decided to apply the same approach but at sentence-level. I tried to sum the labels for each sentence and take the popular label among the sentences of a review, but the results were poor with an overall F1 score of **0.550** (0.395 for the negative class and 0.704 for the positive one). I tried to sum up the positive and negative contributions and assign the final label to the greatest one getting an overall F1 score of **0.700** (0.651 for the negative class and 0.749 for the positive one). I tried also to sum up the compound score and assign the final label based on the final sign getting an overall F1 score of **0.697** (0.649 for the negative class and 0.745 for the positive one).

After these trials, I decided to take also into account the objectivity information and tried to remove the objective sentences, the ones with a neutral score higher than both positive and negative scores. Results using the three same techniques as before were even **worse**, with an average of **0.503** F1 score on the three approaches. Also trying to unify back the subjective sentences and apply again the analysis on that I got an overall F1 score of **0.505** (0.321 for the negative class and 0.689 for the positive one).

4.4 Recurrent Neural Network for sentence-level objectivity classification

The power of RNNs is well-known in the NLP community and this kind of networks allows to process a huge amount of data and learn deep patterns in data allowing to obtain higher performances in a lot of tasks. To prepare the data, I embedded the reviews from Rotten_IMDB using for each word the vectors provided by the *en_core_web_lg* spaCy English model. After that I padded them to a fixed length of 128. I tried to use different kinds of RNNs and with a lot of different parameters, I found out that the best one is with a multi-layer GRU structure followed by fully connected layers both for objectivity detection and sentiment-polarity classification. I also used dropout and leaky ReLU as activation function in order to have better performances at test time. After training the network reach a high **0.937** F1 score in objectivity classification.

4.5 Recurrent Neural Network for document-level sentiment-polarity classification

I used the same network for sentiment-polarity classification, I trained it on IMDB using as part of the data preparation the remotion of the sentences classified as objective by the network trained before. If all the sentences of a review are classified as objective, I decided to maintain all of them and let the network decide the output on the whole review. The preparation after the objectivity remotion is done as before, but this time I padded all the reviews to the maximum length of 384 (the higher this value, the fewer reviews are cut, but the drawback is that with higher values the training requires much higher computation times). After training, the network reaches a high **0.886** F1 score in sentiment-polarity classification.

4.6 Convolutional Neural Network for document-level sentiment-polarity classification

CNNs compared to RNNs should be more powerful since they are able to learn hierarchical features from data and have a better coverage of the sentence. They can learn good local features and aggregate them to get document-level features from which is possible to obtain good performances when classifying the whole review. I tried to use some convolutional-pooling layer structures considering different filter sizes followed by fully connected layers. After a training phase like the one of the RNN using the same data preparation with objectivity remotion, the network reaches a good **0.879** F1 score in sentiment-polarity classification.

I tried to use the same network also for objectivity classification, but for this task it overfits a lot the training set in few epochs and though I was able to get **0.993** of F1 score I decided to use to use the RNN version as part of the preprocessing for the second step.

5 Discussions

Baselines results are not exceptional, the one that is good is the first one based on a support vector classifier, because compared to the other methods it has a training phase and this allows to achieve good performances without any kind of preprocessing (I tried also to retrain it with the reviews without the objective sentences, but I got no improvements in performance). Thanks to its properties on bounding the generalization error, this approach is capable to reach **0.862** of F1 score, much better of the others that without training are not able to get an F1 score over **0.700**.

NNs results were quite good in both tasks, with an **0.937** F1 score in objectivity detection and up to **0.886** in sentiment-polarity classification. A weakness of this approach is that the process of training of the networks is quite slow. The objectivity network one is faster to train thanks to smaller dataset, but the other networks on the IMDB dataset requires about 15/20 minutes for epoch for the training of the networks for sentiment-polarity classification. Another additional time-consuming thing is the process of objectivity remotion that requires about 2 hours. All this was performed on Google Colab with the provided environment and default GPU.

6 Conclusions

Removing objective sentences allows to have less text to analyze when it is time to classify the sentiment-polarity of the reviews. This should lead to better performances comparing to analyze the whole review and allows to speed up the processing and the classification time. The obtained results are good in both the tasks but not perfect. The **0.937** F1 score on the objectivity detection task could cause a reduction in performances on the sentiment-polarity classification. This because if the remotion of the objective sentences is not perfectly done, this can cause the remotion of some important subjective sentences and this can badly affect the network and lead to wrong predictions. With a perfect objectivity detection, I think that the performances could be much higher, but the achieved **0.886** of F1 score is anyway a good result if compared to the other presented methods.

References

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Bo Pang, & Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity. In *Proceedings of ACL* (pp. 271–278).