Sentiment Analysis on Movie Reviews

Matteo Farina (Mat. Number 221252)
University of Trento
Via Sommarive, 9, 38123 Povo,Trento TN
matteo.farina-1@studenti.unitn.it

Abstract

The work focuses on the sentiment analysis task, performed with a Two Stage Classifier that combines Naive Bayes classification and Support Vector classification. The work also focuses on the subjectivity detection task, chaining it to the former to improve the overall performances. An exhaustive experimental section can be found at the end of this document, where many different document representations are evaluated.

1 Introduction and Archive Content

Sentiment Analysis is a very important task under the umbrella of Natural Language Understanding, with relevant impact over recent years with the increased amount of text data shared over the Internet and, above all, social media. This work focuses on the document-level Sentiment Analysis task on the MovieReviews Dataset available within NLTK. The provided archive contains: *(i)* a **src** directory with all the source code, a *(ii)* **models** directory, with dumped models generated while training for different tasks with different configurations, and *(iii)* a **data** directory, where external resources are put. The src folder is organized as follows:

- the **subjectivity_detection** folder contains Python scripts to train subjectivity detectors both at the token-level and the sentence-level;
- the **feature_extraction** folder contains the implementation of a particular document representation named *DiffPosNeg*;
- the **utils** folder contains miscellaneous utilities as well as common preprocessing functions;
- finally, the **main.py** script runs the exhaustive evaluation whose results are listed in the experimental section of the document.

Code with installation and execution instructions is fully available here.

2 Problem Statement

The *Subjectivity Detection* task can be expressed as a binary classification task, where the goal is to classify the input to be either *subjective* or *objective*. This translates into understanding which documents (or sentences) contain opinions and which, on the other hand, contain factual statements.
Sentiment Analysis is the task of classifying the *sentiment polarity* of a document (or sentence). The *sentiment polarity* can either be *positive* or *negative*, with inputs carrying positive or negative opinions, respectively. Since a document may contain factual statements, thus carrying no positive/negative opinion, it is clear that linking the two tasks may be beneficial.

3 Data

This section briefly illustrates the resources used during the project by individual task.

3.1 Subjectivity Detection

The data used for the sentence-level *subjectivity detection* task is the subjectivity dataset avilable at cs.cornell.edu. The dataset contains 5000 subjective as well as 5000 objective sentences, and has been manually preprocessed by removing spanish and portuguese sentences.

3.2 Sentiment Analysis

For what concerns the document-level sentiment analysis task, the Movie Reviews dataset by Pang and Lee (2004) has been used. The dataset contains 1000 positive and 1000 negative documents, each tokenized. Additionally, in the implementation of the *DiffPosNeg* feature as proposed by Nguyen et al. (2013), the SentiWordNet resources by Esuli and Sebastiani (2006) available within NLTK have been used. The SentiWordNet data contain a *positive score*, a *negative score* and an *objective score* for each set of synonyms that share a common meaning.

4 Subjectivity Detection

The *subjectivity detection* task has been tackled in two different ways, and comparisons of the performances of the different methods have been made. The goal of both methods is to provide classification output (*subjective* or *objective*) to a **sentence**. The first method extracts token-level features in order to perform subjectivity detection at the token level, then aggregates the information from the individual tokens to produce the result at the sentence level. The second method performs supervised classification directly on sentences after a sentence vectorization process.

4.1 Token-Level

In this implementation, the following features are extracted for each token of each sentence:

1. its *TF-IDF* value;
2. its *positional encoding* within the sentence, being *(i)* 0 if the token appears at the beginning of the sentence, *(ii)* 1 if the token appears in the middle of the sentence, *(iii)* 2 if the token appears at the end of the sentence;
3. its *part-of-speech* feature, relative to the Universal Tagset by Petrov et al. (2011);
4. its *negation* feature, representing whether the token belongs to a negated (portion of the) sentence.

These features are a subset of the ones proposed for the subjectivity detection task in Kamal (2013). Labels are produced by marking each token belonging to a subjective sentence as *subjective* and *objective* otherwise. A Multinomial Naive Bayes classifier is trained at the token level; thus, an empirical threshold is needed to determine which is the minimum percentage of subjective tokens that causes a sentence to be classified as subjective. The authors in 2013 propose that a single subjective token is enough to classify a sentence as subjective. However, empirical results with this decision criterion showed it caused 100% of sentences to be classified as subjective. After some tests, an empirical threshold of 25% subjective tokens within a sentence is chosen to determine the subjectivity of a sentence. 10-Fold Cross Validation on this method produced an average F1 Score of 54%.

4.2 Sentence-Level

The sentence-level implementation makes use of a **BoW** (Bag-of-Words) representation for each sentence. So, each sentence is represented by a binary vector, with the element at the i-th index being 1 if the i-th word in the extracted vocabulary appears within the given sentence, 0 otherwise. A Multinomial Naive Bayes classifier is trained on this sentence representations, with labels directly available within the subjectivity dataset from Cornell, resulting in a 92% F1 Score after 10-Fold Cross Validation. The BoW representation leveraging the TF-IDF values of tokens has also been evaluated, along with a Naive Bayes classifier based on the Bernoulli probability distribution. Nevertheless, the results were almost identical and all the combinations <representation; classifier> (<*counts*; *multinomial*>, <*counts*, *bernoulli*>, <*tfidf*, *multinomial*>, <*tfidf*, *bernoulli*>) have proved to be equally effective.

Since the sentence-level method significantly outperformed the aggregate token-level approach, it has been chosen for the further Sentiment Analysis task. Thus, every subjectivity detection procedure in Section 5 refers to this approach.

5 Document-Level Sentiment Polarity

The document-level sentiment polarity classification task aims at understading whether a document (a movie review, in this case) contains a *positive* or a *negative* opinion. As introduced in Section 2, it may be the case that some documents don't carry any positive/negative opinion, thus suggesting the use of a subjectivity detector as a helper. In this section, this is explored in two ways:

1. subjectivity detection as a **filter**, rejecting objective sentences within documents before classifying the sentiment polarity;
2. subjectivity detection as a **feature extraction step**, acting on documents by making use of the aggregate subjectivity of their sentences. The subjectivity feature for a document is extracted as a majority vote based on the subjectivity labels of the document's inner sentences.

5.1 Document Representations

For the Sentiment Analysis task, different document representations are evaluated:

1. The *DiffPosNeg* representation, in which documents are represented as the absolute numerical distance between sentences with a positive orientation and negative orientation. This feature was first proposed in Nguyen et al. (2013). The implementation makes use of lexicon-based sen-

timent analysis as seen in this lab by Evgeny A. Stepanov, thus leveraging SentiWordNet and word sense disambiguation techniques to determine the sentiment polarity of tokens within sentences. The polarities are then aggregated from the token-level to the sentence-level, further computing and scaling the distance between orientations to produce the document-level feature.

2. All the BoW representations from Section 4 are also evaluated (count vectorization and TFIDF vectorization).

5.2 Two Stage Classifier

Following the workflow of 2013, a Two Stage Classifier has been implemented. The first stage consists of a Naive Bayes classifier, while the second stage consists of a Support Vector Classifier.

The classification pipeline works as follows: **simple features** are extracted from the data, then fed to the Naive Bayes classifier for sentiment polarity classification. The second stage is only invoked when the first stage is unable to classify input document with enough confidence. This *reject option* is based on two empirical thresholds, T_{neg} and T_{pos}, that determine the minimum required confidence of the Naive Bayes classifier when predicting samples of the negative and positive class, respectively. If the reject option is applied, **richer features** are extracted from the data, then used for classification with the Support Vector Classifier. This classification output is then considered as the final output, with no additional steps.

When properly tuned, this pipeline has the advantage of saving great amounts of computational resources while still maintaining high effectiveness in the classification. On the contrary, the presence of the two empirical thresholds T_{neg} and T_{pos} may lead to a troublesome tuning procedure of the system.

Sidenote: noticing the sparsity of the used BoW representations, an additional pipeline with an LDA (Linear Discriminant Analysis) classifier has been implemented to assess the effectiveness of dimensionality reduction when it comes to Support Vector classification.

6 Experimental Results

This section highlights the results of the different experiments. Each experiment is identified by a combination of four parameters and has been performed with both $T_{neg} = 0.6$ and $T_{pos} = 0.6$. The parameters defining the experiments are listed below:

- **vec1**: document vectorization method used for the first stage classifier (Naive Bayes). Its value can be one among *dpn* or *count*, referring to the aforementioned document representations *DiffPosNeg* and the BoW count vectorization, respectively;

- **vec2**: document vectorization method for the second stage. Since the main idea of the Two Stage pipeline is to provide the second stage with **richer features**, this can be *count* or *tfidf* when $vec1 =$ *dpn* and *tfidf* when $vec1 =$ *count*.

- **subjDet**: subjectivity detection method, being either *filter* or *agg*. Following the principles of Section 5, when the value is *filter* the subjectivity detection phase rejects objective sentences within documents before proceeding to the actual sentiment polarity classification. On the contrary, when this value is set to *agg*, the subjectivity feature obtained by aggregating the subjectivity label of sentences extracted with the subjectivity detector is added as an additional component to the feature vector of each document;

- **dimRed**: whether or not dimensionality reduction with the LDA classifier was used.

For each combination of these parameters, the average F1 Score for the Two Stage Classifier, for the Naive Bayes only and for the SVC only retrieved with 5-Fold Cross Validation is provided. Furthermore, along with their F1 Score, the inference time on the whole MovieReview dataset is shown. The presented inference time spans the entire pipeline from preprocessing to the actual classification step, therefore including also the vectorization step, the subjectivity filtering/feature extraction step and the dimensionality reduction if any. Experimental results are summarized in Table 1.

Remark: in case $dimRed =$ *True* and $subjDet =$ *aggregate*, dimensionality reduction is applied before extracting the subjectivity feature at the document level. Thus, resulting in a 1D feature x for each document where $dim(x) = 2$, the components being the value along the projection direction obtained via the LDA classifier and the subjectivity label.

Table 1: Experimental results obtained by exhaustively running every possible parameter combination.

vec1	vec2	subjDet	dimRed	2Stage F1	NB F1	SVC F1	2Stage time	NB time	SVC time
dpn	count	agg	True	0.52	0.49	0.91	0m:33s	0m:22s	0m:11s
dpn	count	agg	False	0.73	0.52	0.73	2m:54s	0m:22s	2m:31s
dpn	count	filter	True	0.54	0.49	0.89	0m:19s	0m:14s	0m:05s
dpn	count	filter	False	0.75	0.51	0.75	2m:15s	0m:15s	1m:59s
dpn	tfidf	agg	True	0.49	0.52	0.90	0m:33s	0m:22s	0m:11s
dpn	tfidf	agg	False	0.83	0.52	0.83	3m:12s	0m:22s	2m:53s
dpn	tfidf	filter	True	0.49	0.49	0.81	0m:20s	0m:15s	0m:05s
dpn	tfidf	filter	False	0.85	0.55	0.86	2m:25s	0m:14s	2m:13s
count	tfidf	agg	True	0.81	0.81	0.90	0m:07s	0m:07s	0m:11s
count	tfidf	agg	False	0.81	0.81	0.83	0m:07s	0m:07s	3m:21s
count	tfidf	filter	True	0.84	0.84	0.81	0m:04s	0m:05s	0m:05s
count	tfidf	filter	False	0.84	0.84	0.86	0m:05s	0m:04s	2m:44s

7 Discussion

Examining the results listed in Table 1, the following considerations arise:

- the *DiffPosNeg* feature proposed in 2013, although being a conceptually simple feature, is proved to be less effective than the BoW count vectorization for the Multinomial Naive Bayes classifier. Likely, the higher dimensionality of the BoW representation (10^4 order of magnitude for the MovieReviews dataset) causes data points to be far away from each other in the input space, thus making it easier for the classifier to produce accurate predictions;

- the best performances with the SVC classifier are obtained when $dimRed = \mathit{True}$ and $subjDet = \mathit{agg}$, thus when the feature vector for each document fed to the SVC only has two components. This clearly shows that the quality of features has a meaningful impact, perhaps higher than the dimensionality of the input space;

- oftentimes, the F1 score of the Two Stage classifier is (almost) identical to either the F1 Score of the first stage (Naive Bayes) or the F1 Score of the second stage (Support Vector Classifier), meaning that documents at the first stage are (almost) always rejected/accepted. This information indicates how relevant the tuning phase of the fixed thresholds T_{neg} and T_{pos} is in order to best exploit the advantages of the Two Stage pipeline as a whole;

- Although good performances have been achieved with the BoW representations, it would be interesting to assess whether neural embeddings could provide even more accurate results.

As a final note, the reader is suggested not to put too much emphasis on the presented inference times when the *DiffPosNeg* feature is involved. That is because the pure pythonic implementation cannot be compared to the scikit-learn BoW implementations, although being already optimized with parallel operations on multiple cores and with a caching technique similar to an instance of dynamic programming where the sentiment orientation of tokens is stored within a hash-map.

References

Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Ahmad Kamal. 2013. Subjectivity classification using machine learning techniques for mining feature-opinion pairs from web opinion sources.

Dat Quoc Nguyen, Son Bao Pham, et al. 2013. A two-stage classifier for sentiment analysis. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 897–901.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*, pages 271–278.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset.