Final NLU project: Sentiment Analysis

Kevin Depedri (mat. 229358)

University of Trento

kevin.depedri@studenti.unitn.it

1. Introduction

Sentiment analysis (SA), also known as opinion mining, is a branch of Natural Language Processing (NLP) which addresses the identification, extraction and study of subjective information encompassed in documents, texts and reviews. This type of application is becoming increasingly popular, since it represents a powerful tool that both, individuals and companies, can use to get insights about the text under analysis.

A really simple but effective example could be done taking into consideration a manufacturing company. Assuming that this company is interested in checking the popularity of one of its products, it will need to gather information about whether the product is liked or not by its consumers.
This can be done in different ways, here sentiment analysis clearly offers one way of accomplishing the job in a quick and cost constrained manner, with respect to hiring human employees to do the job. All this is possible thanks to the advent of social networks, review websites and generally of all the digital communication systems, which give a huge quantity of information that can be used to train and test sentiment analysis algorithms.

Performing sentiment analysis is a complex task and it is composed of different sub-tasks, which can be mainly summed up as:

- ***analysis*** *of the input text*
- ***objectivity identification*** *and removal*
- ***pre-processing*** *of the text*
- ***training and fine-tuning*** *the chosen model for the* ***polarity classification***

The purpose of this project is to perform sentiment analysis over the Movie Reviews data-set which is composed of 2000 movie reviews. To accomplish this task all the above listed steps have been performed, and, two different models have been trained, fine-tuned and tested over different possible pre-processing configurations. A detailed explanation of the performed operations follows in the ensuing sections.

2. Task Formalisation

The procedure for Sentiment Analysis adopted in this project can be described through the following list of steps:

1. **Analysis of the data-set**, compute general and specific statistics regarding the used data-set;
2. **First classification phase**, compute two baselines (see section 4) to understand the level of accuracy that can be achieved without applying any change to the data-set;
3. **Objectivity removal**, train a model to discern objective and subjective sentences, then apply this model to remove all the objective sentences from the used data-set;
4. **Second classification phase**, compute a new baseline to understand how the level of accuracy has changed removing the objective sentences, but without performing any pre-processing;
5. **Applying pre-processing**, use different functions to process the data-set. This procedure should allow to increase the sentiment classification accuracy;
6. **Third classification phase**, compute the final accuracy obtained after objectivity removal and pre-processing.

Two different types of ML models have been used in the previously described procedure, in particular:

- ***Multinomial Naive Bayes (MNB)****, used to perform 1st classification phase, objectivity removal, 2nd and 3rd classification phase;*
- ***Bi-Directional LSTM Recurrent Neural Network (BI-LSTM)****, used to perform the 3rd classification phase with a deep model, both with and without self-attention.*

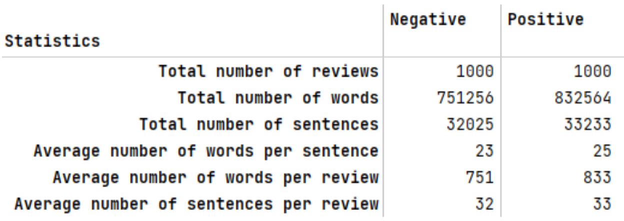
3. Data Description and Analysis

The training and the evaluation of the models used in this project have been performed relying on two different data-sets. The data-sets have been imported from the Natural Language Toolkit (NLTK) library and they are composed as follows:

- ***Movie Reviews data-set****, it is composed of 2000 movie reviews, the first 1000 are negative and are identified with the label: 'neg', while the other 1000 are positive and are identified with the label: 'pos'. This is our major data-set and it has been used in all the training and evaluation phases, except for the one regarding objectivity detection and removal;*
- ***Subjectivity data-set****, it is composed of 10000 sentences, of these, 5000 are subjective and are identified with the label: 'subj', while the other 5000 are objective and are identified with the label: 'obj'. This data-set has been used only for the objectivity detection and removal step.*

The previous data-sets have been created by Bo Pang and Lillian Lee [1] and are redistributed with NLTK with permission from the authors.

The majority of the work has been done on the Movie Review data-set. For this reason, a comprehensive analysis has been carried out for this data-set. To begin with, some general statistics have been computed over the two polarities of the data-set. The results are shown in the underlying Figure.1.
From this data it is possible to notice how, generally, the positive reviews have a higher number of sentences and words, leading this type of review to be longer than the negative ones. This is also confirmed graphically by the plot in Figure.2. Furthermore, to see more in details the previous values, some specific



Statistics	Negative	Positive
Total number of reviews	1000	1000
Total number of words	751256	832564
Total number of sentences	32025	33233
Average number of words per sentence	23	25
Average number of words per review	751	833
Average number of sentences per review	32	33

Figure 1: *General statistics for Movie Review data-set.*

statistics have been computed for each review. The results can be consulted in the project .ipynb notebook file.

To further investigate what emerged from the previous analysis, the number of words distribution has been plotted both for positive and negative reviews, it is shown in the underlying Figure.2.

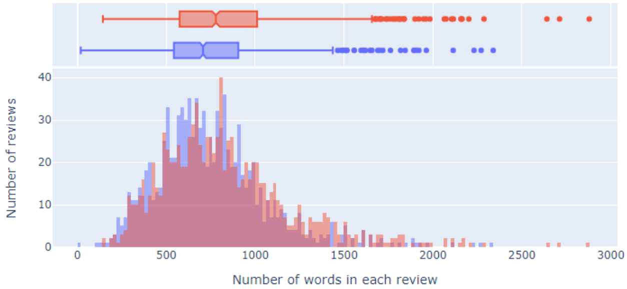


Figure 2: *Number of words distribution. Positive reviews in orange, negative reviews in blue, overlap in red.*

The plot clearly shows that the positive reviews are on average longer than the negative ones. This can be verified very quickly from the marginal box distribution plotted above the histogram. Having positive reviews longer than negative reviews could depend on the fact that, when people are happy about something, they are more prone to spend time giving it a good review. On the other side, people that are not happy could think that they have already wasted enough time. Unfortunately, there is just a slight difference in length between the two polarities. Therefore, a feature based on the length of the review may result ineffective.

As last step of the data-set analysis, the 25 most frequent words, both for negative and positive reviews, have been plotted. The results are shown in the underlying Figure.3. Notice that in the plots, both punctuation and stop-words have been removed. In this way we are able to visualize a more realistic histogram about the important words in our data-set. To see the plots with stop-words please consult in the project .ipynb notebook file. From the plots we can see how, for both the polarities, we have a few words with a very high frequency, while all the other words in the top 25 most frequent words have approximately the same frequency. This suggests us that, in the pre-processing step, it will be fundamental to impose a cut-off threshold that will remove the words which are too frequent, since they will act as stop-words without bringing any relevant meaning for the classification task.

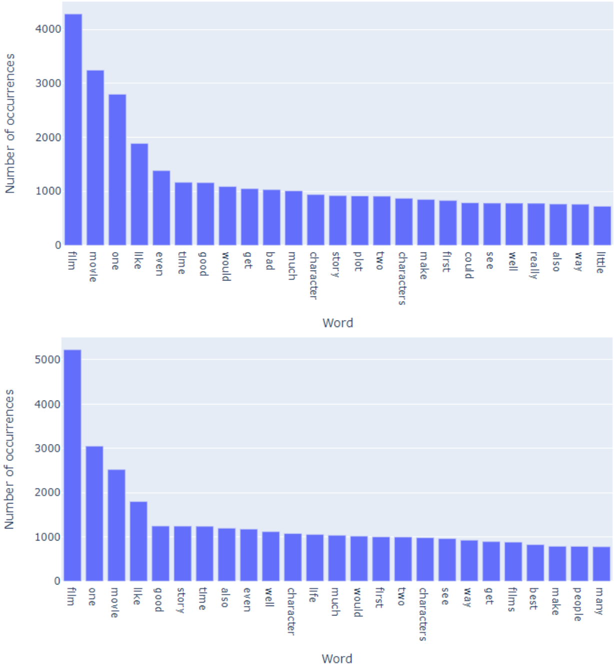


Figure 3: *Top 25 most frequent words, for negative reviews above and for positive reviews below.*

4. Model

The model that has been built is divided in different sections. The purpose of each section is to show how distinct processing of the data-set affect the final classification result. For this reason, more than one baseline and accuracy value have been computed in the model. The availability of different accuracy values will be crucial moving on with the optimization, since it allows to compare the effectiveness of the processing steps, guiding us toward the most optimal processing setup. The numerical results relative to the steps discussed in this section are visible in Figure.7 of section 5.

The **First classification phase** has the goal to show the polarity classification performance over the raw-text of the Movie Review data-set. To assess that, we have employed:

- ***CountVectorizer****, vectorizer used to convert words in numbers, essential since ML algorithms are able to deal only with numbers. Imported with default settings;*
- ***MultinomialNB****, Naive Bayes classifier used to carry out the supervised classification task. Lowering option enabled to lower the input text and avoid caps problem;*
- ***TrainTestSplit****, splitting tool used to divide the data-set in train-set (80%) and test-set (20%). Shuffle and stratify options have been enabled to ensure that train and test sets are both chosen randomly but in a balanced way;*
- ***IterativeLoop****, loop of 20 iterations where in each one split, train and test are performed. It allows to compute a more stable and reliable average accuracy.*

In this phase, two baselines have been computed to show one possible error that can be performed. The first one has been computed fitting the vocabulary of the vectorizer over the entire dataset. This is a wrong approach in the real world, since, when we build a model, we can fit it only on the available data. Indeed, the testing data will come at inference time. The second one has been computed fitting the vocabulary of the vectorizer over the train-set of each iteration, this is the correct approach. The two baselines show us how this error can affect our final accuracy. Their values are reported under the labels ”RawText - FullDict” and ”RawText - 80Dict” in Figure.7. From now moving on, instances of vectorizer will be fitted with the correct approach.

The **Objectivity removal** follows the previous phase and its goal is to train a classifier which can then be used to identify and remove objective sentences from the raw-text of the Movie Review data-set. Here we have used the same classifier and the same procedure as described above, applying it to the Subjectivity data-set. Once that the classifier has been trained, it has been deployed to remove all the objective sentences from the Movie Review data-set. This allows to work on subjective sentences, from which it is possible to extract the sentiment. Results are reported under the label ”Object Removal (OR)” in Figure.7

The **Second classification phase** is directly linked with the previously performed objectivity removal. Here we have trained and tested a classifier as described previously in the beginning of this section, but this time input data has been the Movie Review data-set cleaned up by the objective sentences. The result is a more reliable polarity classification since the classifier is dealing with subjective and relevant information. Results are reported in Section.5 (not in the table in Figure.7).

The **Pre-processing** follows the previous phase and its goal is to perform different operations over the subjective data to make them even more clean and relevant for the classifier. These operations should enable us to remove noise from the data and should lead to a more meaningful input for our classifier. The operations considered in this project are:

- ***Tokenization and its correction***, *it is the procedure of breaking down each sentence in words (also called tokens). After tokenization we apply a correction since this procedure leads to some wrongly tokenized terms such as ”don’t” which is broken in ” don’ ” and ” t ” ;*
- ***Removal of punctuation, stop-words and numbers***, *all these are frequent elements in the reviews. They do not bring any evident semantic meaning and do not increase our capability to distinguish a positive polarity from a negative one. Furthermore, in the case of numbers they can cause problems since they could be referred to dates, points of a list, or to marks given to the films;*
- ***Correction of elongated words***, *used to correct words which have been written in strange ways (e.g.: ”cooool” become ”cool”), this should allow to increase the generalization capabilities of the model;*
- ***POS analysis and Lemmatization***, *POS tags are used in the lemmatization procedure to group together the inflected forms of a word and trace them back to a single item called lemma (e.g.: ”am”, ”are” become ”be”);*
- ***Words frequency cut-off***, *used to remove words which are too frequent (that act as stop-words) and that are too rare (slow training and do not allow to generalize well);*
- ***Negations handling***, *used to manage negations and recurrent negations adding a _NEG tag to the words that come after a negative word.*

The previous list of operation has been applied in different configurations to be able to evaluate the effectiveness of different setups. The list is reported here below:

- ***PreProcessing1***, *Tokenization and its correction;*
- ***PreProcessing2***, *PreProcessing1 + Removal of punctuation, stop-words and numbers;*
- ***PreProcessing3***, *PreProcessing2 + Correction of elongated words + POS analysis and Lemmatization;*
- ***PreProcessing4***, *PreProcessing3 + Words frequency cut-off*
- ***PreProcessing5***, *PreProcessing4 + Negations handling*

The **Third classification phase** is finally performed after the pre-processing step. Here, we trained and tested the two classifier introduced in Section.2 under the different configurations of pre-processing listed above. In particular:

- ***Multinomial Naive Bayes*** *has been trained using n-grams in the range of 1-3 and limiting the number of features to 20000 in all the configurations. This values have been proven to be optimal through testing, leading to lowered training time and increased accuracy;*
- ***Bi-Directional LSTM*** *has been trained twice for each configuration. The first time without attention and without any limit on the length of the input sequences. The second time with self-attention and with a limit of 700 elements in the sequence, this to lower train time and to allow training over a machine with 16GB of RAM.*

Results are reported under the labels ”(OR) + PreProcessingX” in Figure.7

5. Evaluation

The evaluation of the different phases of the model has relied fully on the accuracy metric. Starting from the **First classification phase** we can see in Figure.7 how we achieved accuracy values of 82.2% and 81.4%, respectively with the wrong approach and with the correct one. As previously mentioned the difference is not substantial, but it shows how the vectorizer fitted on a smaller quantity of data leads to a lower accuracy (since it will encounter words that it does not know and that consequently will not be converted to numbers).

Moving on with the **Second classification phase** we can see from Figure.7 how this procedure lead to an increase in accuracy to 84.9%. This was expected, since, removing the objective sentence allows the classifier to focus on the subjective ones, where the polarity is expressed. Looking at Figure.4 we can notice how, at this point, the ROC curve has a great shape, with an AUC value equal to 0.920. This means that the classifier performs very well and is able to distinguish accurately between the two classes. Despite that, we still miss a 0.080 to reach the performance of an ideally perfect classifier, meaning that we have margin for further improvements. Furthermore, the accuracy distribution shows how the value of accuracy reached in the different iterations are quite widespread, between 82% and 88%, with peak concentration around 85%.

The **Third classification phase** shows even more interesting results in Figure.7. Here we have 5 values of accuracy for the 5 Pre-processing configurations which stand between 77.5% and 86.5% for the MNB classifier. Analysing the best result, plotted in Figure.5, we can see that the ROC curve has a

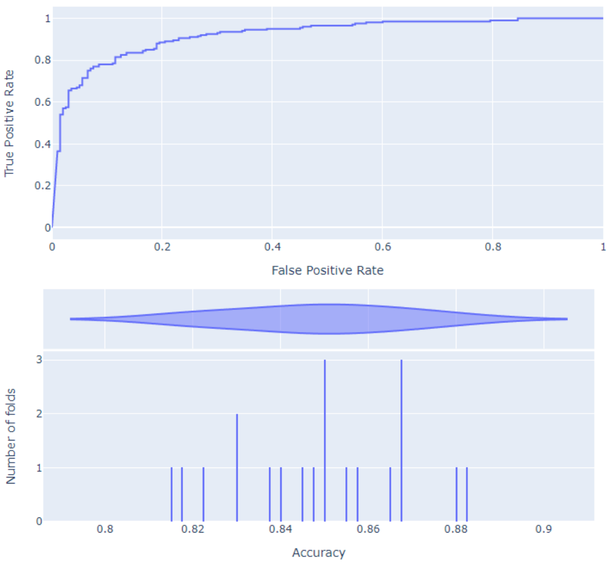


Figure 4: *ROC curve and accuracy distribution over 20 iterations for MNB classifier with Objectivity Removal(OR) applied.*

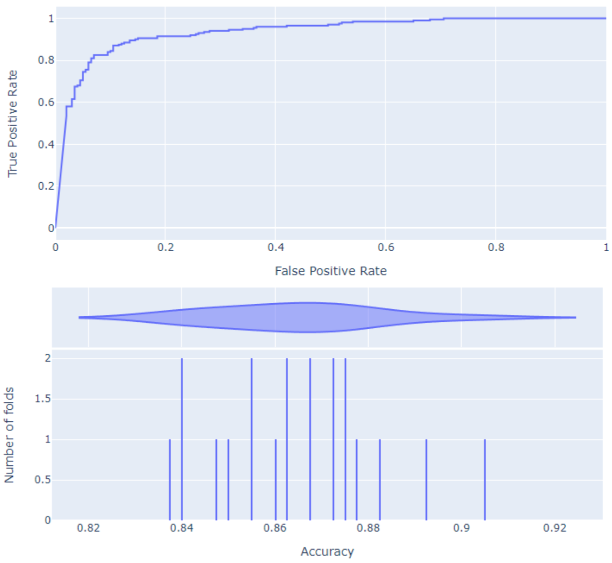


Figure 5: *ROC curve and accuracy distribution over 20 iterations for MNB classifier with OR + PreProcessing1 applied.*

slightly better shape with respect to the one shown in Figure.4. This is confirmed also by the AUC values, which rises to 0.935, bringing the classifier closer to an ideally perfect one. Also the accuracy distribution has changed, with a shift from the previous 82% - 88% to a new range between 84% and 91%, with peak concentration around 87%.

These results confirm that the pre-processing worked, allowing the MNB classifier to perform better. Although an increase in performance was foreseen, it has been quite unexpected that the best result is the one related with PreProcessing1, which, as seen before, is only composed off Tokenization and its correction. Indeed, we expected more substatial pre-processing configuration, like the number 3 or 4 to perform better, since these should be able to remove even more noise from the data-set, allowing the classifier to generalize better. Therefore, we can affirm the the results contradict our expectation, and this is probably due to the size of the Movie Review data-set. Indeed, pre-processing is performed to reduce the quantity of data that will be used to build a model, with the idea of removing the less relevant data, while keeping only the most useful for generalization. This works as expected only when the initial quantity of data is large enough. In this case, we performed the pre-processing over a data-set which was already small in the beginning (2000 reviews). Furthermore, we removed the objective sentences from the data-set, which made every review even shorter. At this point we performed pre-processing, this removed even more data, which we deemed as irrelevant, but that in reality was essential, since the removal of this data started to leave the classifier with a too small data-set to train its model.

Looking at the **Bi-Directional LSTM** results, we have values of accuracy between 56.5% and 62.7% for the implementation *without attention*. The values rise drastically between 81.0% and 88.2% for the implementation *with self-attention*. The best accuracy and loss over epochs for the Bi-Directional LSTM with self-attention are shown in Figure.6. These plots show a very high training-accuracy, near to 98.1% and a testing-accuracy which is quite close to the training one, with a value of 88.2%. We have chosen a number of epoch equal to 3 since higher values lead to over-fitting for the model *with attention*, where training-accuracy increased close to 99.9% while testing-accuracy decreased considerably. Differently, the models tested *without attention* have gone through 5 epochs, since these models have proved to be more resilient to over-fitting. For this two different architectures, the best performing pre-processing configuration have been respectively PreProcessing5 and PreProcessing3, this shows how a deep-model have different needs in term of input, to perform optimally, with respect to a non-deep model such as MNB.

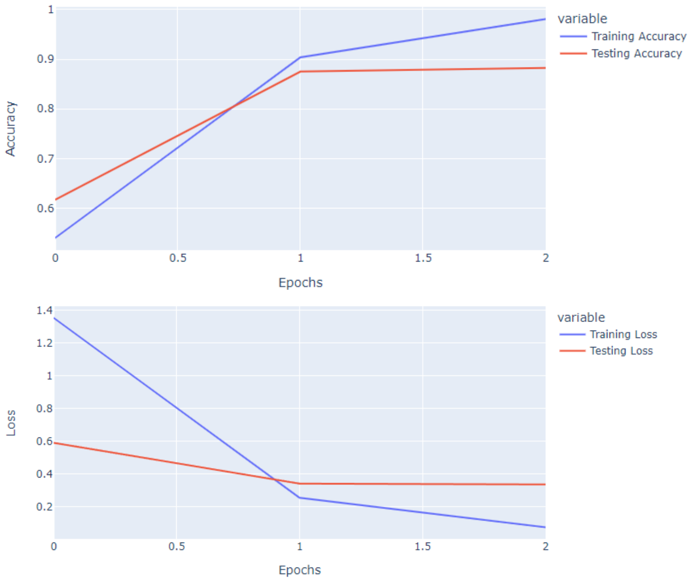
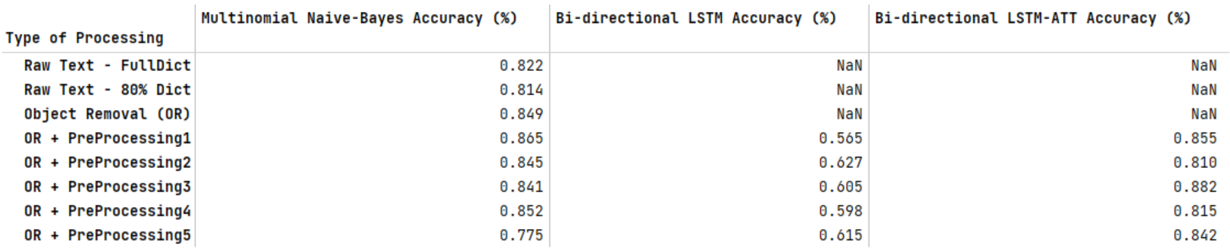


Figure 6: *Accuracy and Loss over 3 epochs for BI-LSTM self-attention model with OR + PreProcessing3.*



Type of Processing	Multinomial Naive-Bayes Accuracy (%)	Bi-directional LSTM Accuracy (%)	Bi-directional LSTM-ATT Accuracy (%)
Raw Text - FullDict	0.822	NaN	NaN
Raw Text - 80% Dict	0.814	NaN	NaN
Object Removal (OR)	0.849	NaN	NaN
OR + PreProcessing1	0.865	0.565	0.855
OR + PreProcessing2	0.845	0.627	0.810
OR + PreProcessing3	0.841	0.605	0.882
OR + PreProcessing4	0.852	0.598	0.815
OR + PreProcessing5	0.775	0.615	0.842

Figure 7: *Final results for the trained models.*

6. Conclusion

Through this work we have shown the procedure followed to build a Multinomial Naive Bayes and a Bi-Directional LSTM (with and without attention) that have been used to perform sentiment analysis. The result is that both the models have reached very high level of accuracy, around 86.5% and 88.2% respectively. The Objectivity Removal step has proven to be a straightforward and essential step to get better performances for SA tasks, while the optimal pre-processing is more difficult to choose and requires extensive testing. Finally, a quick comparison between BI-LSTM and BI-LSTM with attention has been carried out, showing how the former is more resilient to over-fitting but yields lower value of accuracy, while the latter is easier to over-fit but also much more accurate in its predictions.

7. References

[1] B. Pang and L. Lee, “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, Jul. 2004, pp. 271–278. [Online]. Available: https://aclanthology.org/P04-1035