Final NLU project - Sentiment Analysis

Davide Michelon (229313)

University of Trento

davide.michelon-1@studenti.unitn.it

Abstract

This document represents the report of the project developed for the MS.C. course Natural Language Understanding at the University of Trento. The main goal was to develop a Sentiment Analysis system that consists in two sub-task: (1) Subjectivity Detection and (2) Polarity Classification.

1. Introduction

Sentiment analysis aims to identify the point of view or emotion behind a situation. It basically means to extract, quantify and study effective states and subjective information in order to find the emotion or intent behind a piece of text. Sentiment analysis has several different uses. Most notably, with the rise of social media sites and the rise in review, rating, and recommendation sites, companies are becoming increasingly interested in this topic. Adopting an algorithm of sentiment analysis allows companies to filter out information and extract the most relevant metrics that can be used to achieve their specific goals. Moreover, Sentiment analysis can be applied to varying levels such as document, paragraph, sentence and sub-sentence levels. Sentiment analysis is done using a variety of different methods, including NLP, statistics, and machine learning methods. This report shows some solutions using these techniques. In particular, the sections of the paper are structured as follows. Firstly, in the Task Formalisation, the goal of the project and the procedure used will be explained and formalized. Secondly, in the Data Description Analysis will be presented the composition and some statistics of the two datasets. Finally, there are the Model and Evaluation sections. In the former will be explained the baseline, experiments, and the final solution whereas in the latter the metrics used in the models and the comparison among them.

2. Task Formalisation

The project consists in implementing and solving two sub-tasks:

- Subjectivity Detection: classifying a sentence or a fragment of text into one of two categories (subjective, objective)
- Polarity Classification: Extraction of the polarity of a text (positive, negative)

The first problem approached is related to the subjectivity detection. After finding a reliable model, it is then used to filter out the movie reviews in order to feed the polarity model with reviews composed only of subjective sentences. In fact, objective sentences are considered neutral and consequentially useless for the polarity analysis. Moreover, since there are less sentences the classification will be faster and it will refer only to two possible output categories: positive or negative [1].

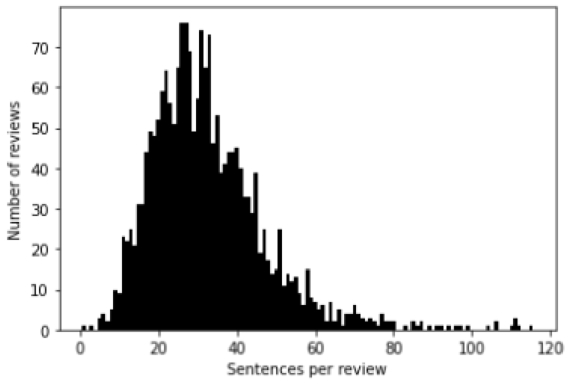

Figure 1: *Distribution of number of sentences per review*

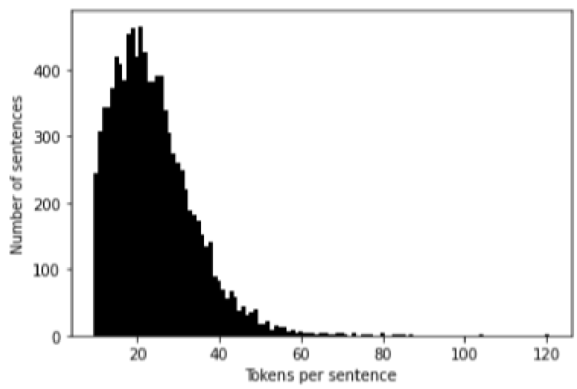

Figure 2: *Distribution of number of tokens per review*

Subjectivity Dataset (NLTK)	
Total sentences:	10.000
Sentences for each label:	5.000
Labels:	['obj', 'subj']
Average tokens for sentence:	24

Table 1: *Features of Subjectivity Dataset*

Movie Reviews (NLTK)	
Total reviews:	2.000
Sentences for each label:	1.000
Labels:	['pos', 'neg']
Average sentences for review:	33
Average tokens for review:	791

Table 2: *Features of Movie Reviews Dataset*

3. Data Description Analysis

Firstly, the Dataset used for the subjective detection is the subjectivity dataset provided by NLTK. As it is shown in table 1 it is a labeled dataset and contains 5000 subjective and 5000 objective processed sentences, from figure 2 it is possible to notice that most sentences have between 15 and 30 tokens. Secondly, the dataset of movie reviews is shown in table 2. It contains 2.000 reviews and each review has on average 33 sentences and moreover in figure 1 is shown that most documents have between 10 and 50 sentences. When the dataset has to be splitted in train and test set, the samples are splitted using a *scikit-learn* method which splits arrays or matrices into random train and test subsets. Each sentence/review needs some pre-processing steps. Firstly, each string is converted to ASCII and normalized. Then, it has to be converted in a vector. We adopted three methods:

- Count Vectorizer: Convert a collection of text documents to a matrix of token counts. The representation produced is parse.
- Word embedding: Using an English pipeline optimized for CPU (en_core_web_lg), it converts each word to a vector of 300 dimensions.
- BertTokenizer: Pre-trained tokenizer from the BERT base model.

One of the first attempts was to adopt a 1-hot vector approach considering all words in the reviews. Unfortunately, the dictionary contained more than 20,000 words which led to unsustainable computational costs.

4. Model

4.1. Baseline for Polarity Classification

The baseline for the Subjectivity Detection step is a SVM classifier and has been found as surprisingly reliable. Steps: (1) Data splitting in training and testing, (2) Data Normalization, (3) transformation of words into vectors using CountVectorizer, (4) fit the model according to the given training data, (5) perform classification and at the end (6) compare predicted labels with the correct ones.

4.2. Baselines for Sentiment Analysis

The models are:

- Support Vector Machine proved to be a fairly efficient method, indeed it is memory efficient (uses a subset of training points), versatile and effective in high dimensional spaces. In this case, the classifier is fed with the whole review, both subjective and objective sentences were considered. The steps of the implementation are almost identical to those shown in the aforementioned SVM.
- Support Vector Machine using only subjective sentences. Each sentence is evaluated by feeding it to the baseline model for the Subjectivity Detection. If the sentence is objective it is dropped, otherwise it remains part of the review.
- Sentiment Intensity Analyzer of Vader module for NLTK [2]. The reviews are directly passed through the analyser and the category with the highest value is assigned.
- Another version of Sentiment Intensity Analyzer of Vader module. The analysis is made for each sentence of the review and the contribution is added up for each one.

4.3. Model for Subjectivity detection

RNN architecture remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data. To run these networks, firstly it is needed to do some pre-processing steps on the input. These steps consist in removing stop words and punctuation and convert the text to a vector. Afterwards, the sequence length is padded to a fixed size and the input fed to the models. The adopted model is a LSTM (Long short-term memory) which is an RNN extensions designed to learn sequential data and their long-term connections more precisely than standard RNNs. Indeed, the architecture has been chosen to overcome the vanishing gradient problem. After various tests and attempts it was decided to use a Bidirectional LSTM, where the input flows in both directions, and it is capable of utilizing information from both sides. The architecture has also two recurrent layers, with the second LSTM taking in outputs of the first LSTM and computing the final result. After the LSTM layers, had been also added:

- In order to encourage regularization, prevent the co-adaptation of neurons and avoid overfitting, it was decided to apply the Dropout technique. In fact, it randomly zeroes some of the elements of the input tensor with probability p=0.3.
- LeakyReLU with negative slope=0.1.
- Final linear layer with output dimension=2(positive and negative).

The number of features in the hidden state was chosen by testing the model with 256, 512 and 1024, although no great differences were noticed, 256 was chosen as a compromise between accuracy and time.

4.4. Model for Polarity Classification

The initial solution was to adapt the previous LSTM for the Sentiment Analysis task, unfortunately the results were not satisfactory and therefore it was decided to change strategy. Thanks to [3] and [4], it was possible to create a reliable model based on BERT (Bidirectional Encoder Representation for Transformer). Transformer architecture has encoder and decoder stack whereas BERT is just an encoder stack of transformer architecture. The greatest advantage in comparison to LSTM is that the model is non-directional, so the attention mechanism allows for learning contextual relations between words.

In this project, we used the pre-trained BERT-base-uncased model which has 12 layers in the encoder and it does not make any difference between upper and lower characters. It is needed a Tokenizer to convert the text. It has been chosen to use the *Tokenizer.encode_plus* which returns a dictionary containing the encoded sequence with some additional information. Although the model is not too complicated, the results obtained are satisfactory. Finally, besides the Bert part, a Dropout with p=0.3 and a Linear layer are added. A big issue during training was the exploding gradient. To avoid it we clipped the gradient norm using Torch.nn.utils.clip_grad_norm_ .

5. Evaluation

This section will show the results of the various baselines and the models. The main metric considered is the accuracy which is computed for each model but in some of them the data is shown using the classification_report of sklearn which is a text report showing the main classification metrics. These metrics are (in addition to accuracy): precision (percentage of correct positive predictions relative to total positive predictions), recall(percentage of correct positive predictions relative to the total actual positives) and f1-score (a weighted harmonic mean of precision and recall).

5.1. Evaluation for Subjectivity Detection

As a baseline we used the SVM which gave very interesting and satisfying results, indeed the accuracy value is **0.879**. The model used is a bidirectional-LSTM. After various tests and some adjustments in the parameters, the accuracy obtained after 10 epochs is about **0.9035**. For the training, after 5/6 epochs an almost maximum accuracy (approximately 0.98) has been already reached whereas for the testing the maximum accuracy achieved in the whole period never exceeds 0.92.

5.2. Evaluation for Polarity Classification

Four baselines were shown for this task, two of which use Sentiment Intensity Analyzer of VADER module and their results have never been good, the accuracy achieved is approximately 0.64. The other two baselines use the SVM classifier and maximum accuracy is achieved when objective sentences have been removed from reviews, the accuracy reached is approximately **0.850**. By doing various tests, the difference between the accuracy of the two SVMs is variable but in general the one without objective sentences is 2-3% higher than the one with objective sentences. The BERT model has achieved an accuracy of approximately **0.895**. In the Figure 3 is shown the trend of the accuracy for training and testing, the training reaches 100% accuracy after a few epochs while the test accuracy stabilizes around 88-90%. The model was also tested with stop words and without normalization and the accuracy is approximately 4-5% less accurate.

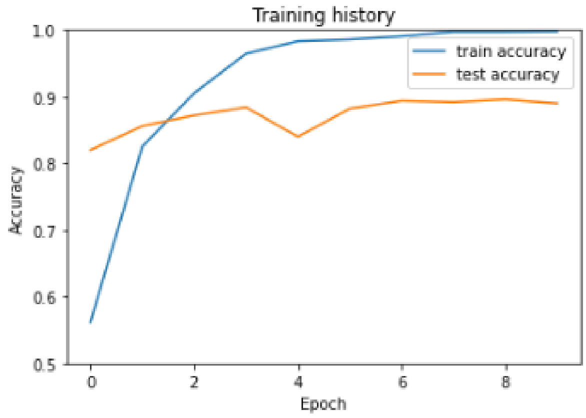

Figure 3: *Training of BERT on Sentiment Analysis*

Accuracy for Polarity Classification	
SVM with obj sentences	0.818
SVM without obj sentences	0.850
VADER on the whole review	0.620
VADER with each sentence analyzed separately	0.642
BERT	0.895

Table 3: *Accuracy for Polarity Classification*

6. Conclusion

To sum up, this work presents two different models to solve two sub-task of Sentiment Analysis: Subjectivity Detection and Polarity Classification. For the former the model used is a LSTM whereas for the latter it is a pre-trained BERT. In both cases the results obtained are acceptable and the baselines have been improved. BERT is fed with reviews that are pre-processed and indeed subjectivity sentences are removed. The removal of these sentences is based on an SVM which has an accuracy of 0.88, improving this model would also improve the polarity classification.

7. References

[1] Pang, Bo and Lee, Lillian, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts" *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 2004.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text" *Proceedings of the International AAAI Conference on Web and Social Media, 8(1), 216-225*, 2014.

[3] Ashish Vaswani,Noam Shazeer,Niki Parmar,Jakob Uszkoreit,Llion Jones,Aidan N. Gomez,Lukasz Kaiser and Illia Polosukhin "Attention Is All You Need" , 2021.

[4] Jacob Devlin and,Ming-Wei Chang and,Kenton Lee and Kristina Toutanova "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" , 2018.