Sentiment analysis of custom reviews Data-Set

Davide Saitta Davidesaitta39@gmail.com
Fabio Amoroso Fabioamorosofa98@gmail.com

1. Model Description

Two main approaches have guided our work: a topic-based and a binary fine-grained one. The scope of the topic-based approach is to extract recurrent topics in the text data. The fine-grained, instead, helps us identifying the polarity of a sentence (positive or negative).

The models used for the two approaches are quite similar: in both cases the models contain three layers. The first one is an embedding layer, that receives as input the vocabulary that we have previously created converting each word of the dataset into an integer. The goal of the embedding layer is to map words to tensors of size 256 for the topic-based analysis and 128 for the fine-grained one.

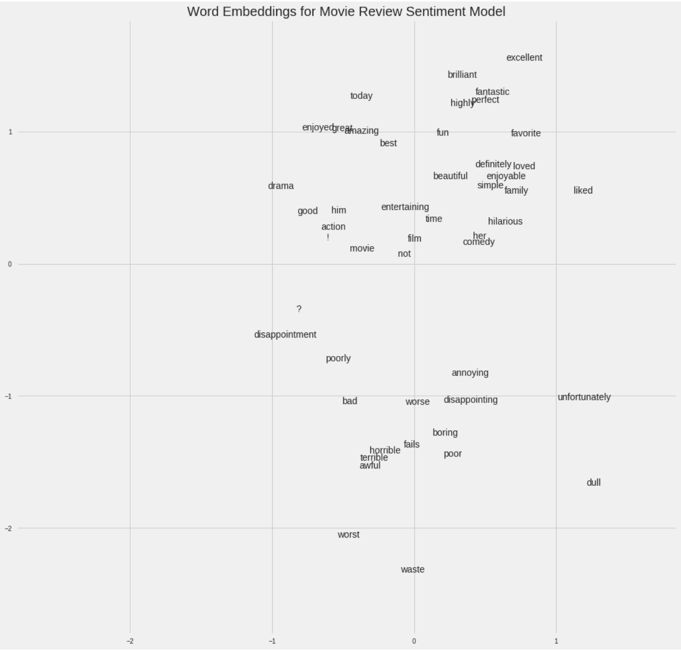

Figure 1: An example of movie's Sentiment Analysis embedding.

Then, there is an LSTM Cell that introduces the recurrency in our models. It decide, through the use of sigmoids and tanh functions, what information to throw away and what new information to store in the cell state. In this way, we can keep track of long term dependencies and it is useful in our case because we have to deal with some long sentences. The output of our LSTM Cell has size 64 for the topic recognition task and 32 for the sentiment classification. The output pass through a dense layer to return the final results.

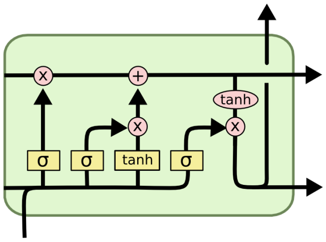

Figure 2: A simple representation of the LSTM Cell.

The first model has the scope to identify three classes. So, the output of the model is a 1x3 tensor. Instead, the fine-grained model return a tensor of size 1x2. We have tried a lot of different architectures of the models and hyperparameters, but these models are the ones that gave us the best trade-off between performance and computational time.

2. Dataset

For this project, we have created a customized data-set. The aforementioned data-set was created by taking 10,000 statistical units (5000 positively labeled and 5000 negatively labeled) from each of the following data-sets taken from Kaggle website: ”Amazon reviews”, ”Yelp Review Sentiment Data-set” and ”Large Movie Review Data-set (Maas et al., 2011)”.

The resulting data-set is a perfectly balanced data-set, consisting of 30,000 labeled reviews. To make this possible, some data preprocessing work was required. Specifically we had to:

- align the labels which, even if in all three cases, represented positive or negative, this was determined by distinct values.
- remove the column with the review’s title from the amazon review data-set.

Once the data-set was created, it was manipulated in order to work on it. First we made everything lowercase, removed all punctuation and divided each review by row, obtaining a list. After that, since the input of our model is the single word, we split the words within the sentence, obtaining a list of lists. As our RNN cannot use words directly, we converted each word in a number.

Finally, seeing that the average length of the reviews is about 142 words, and considering some reviews exceeded 2000 words, we opted for a resizing of the reviews to a length of 200 words, using padding where necessary and converting the result to tensor. Data-set's labels are also converted to tensor. Since the project consists of a double analysis, a second set of labels is created for the topic based task. The resulting data-set was divided into train, validation and test with a proportion of 80%, 10%, 10%.

3. Training procedure

The training of the models have been performed with the aid of the GPU accelerators provided by CUDA. The batch size chosen for the data loaders is 64. The loss has been computed with the cross entropy criterion, that calculate a separate loss for each batch and sum the results.

For the backpropagation phase, we chose Adam as the optimizer. We have also applied a weight regularization with the value of 2e-4 for the topics-based model and 5e-4 for the fine-grained one. Also the learning rate differs between the two models: we choose 0.009 for the first model and 0.008 for the second one. Both models have been trained for 35 epochs.

4. Experimental Results

The experiment conducted to the topic recognition model lead us to a test accuracy of 93.91%. The results can be seen in the following confusion matrix.

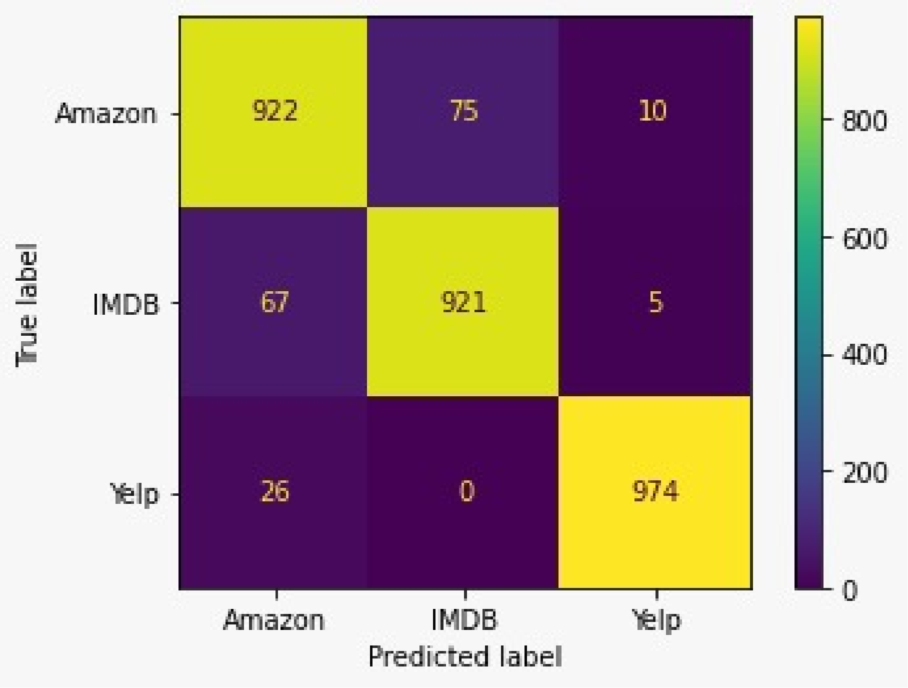

Figure 3: Confusion Matrix of the results obtained with the topic-based model.

Looking at the confusion matrix, we can do some considerations. First of all, the model have learn how to properly distinguish between IMDB and Yelp classes. The ratio of correctly classified on the total number of reviews for each class is: Amazon 91.56%, IMDB 92.75% and Yelp 97.4%. So, the best classified class is Yelp. The worst, instead, is Amazon and it could be due to the different kinds of products reviewed, that could be related to movies or catering and lead to some misclassifications.

The fine-grained model returned a test accuracy of 86.81%.

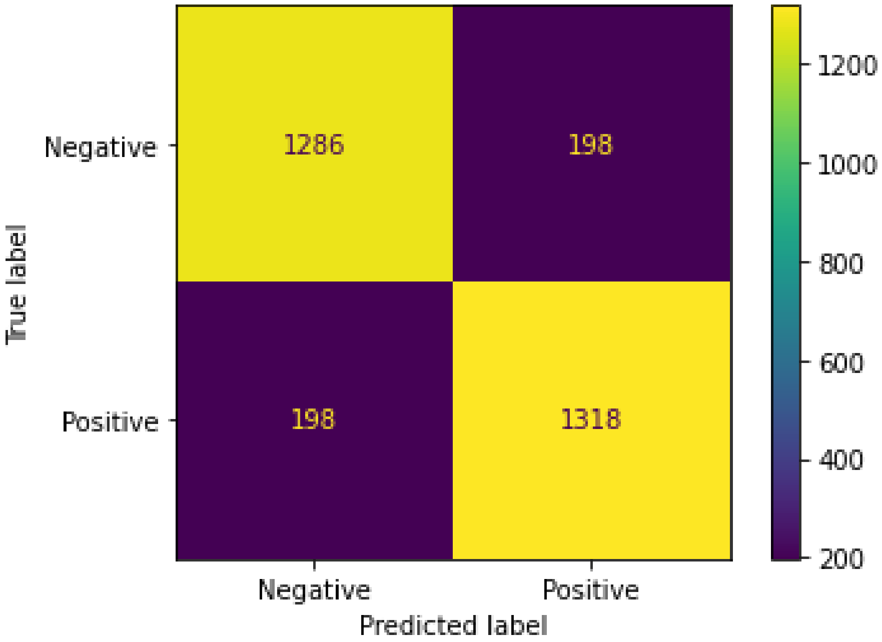

Figure 4: Confusion Matrix of the results obtained with the fine-grained model.

In this case, we can see that the 86.94% of the true positive reviews have been correctly classified, while the true negative reviews classified correctly are the 86.66%. The model have learnt how to distinguish between positive and negative reviews with a similar effectiveness.