Sentiment Analysis Text Classification

Idan Vidra (ID. 313157257)
Shahar Olewski (ID. 206895039)
Noa Olewski (ID. 206895047)

Submitted as final project report for the NLP course, IDC, 2021

1 Introduction

In this project we explore the subject of sentiment analysis while using different models. We chose this subject because we find it a very interesting and important tool in these days, specifically, for a deeper connection with the technology around us.

Our main idea is to compare between different models in order to find which one is the most useful model for a specific type of mission. In addition, we used these tools to create a human-computer interaction scenario.

1.1 Related Works

Comparing different models for sentiment analysis is a well-known subject and many other works have been published regarding this topic. While we didn't implement any of those works, we did looked at other papers for different scenarios - "Does Model Size Matter? A Comparison of BERT and DistilBERT", by Jack Morris, and "Working in Detail: How LSTM Hyperparameter Selection Influences Sentiment Analysis Result", by Nicholas Daniel Derra and Daniel Baier.

2 Solution

2.1 General approach

For our primary objective, we want to classify customers' reviews using sentiment analysis. In order to achieve better insights, we use different methods - BERT, LSTM and Rule Based. For each model we also use different hyperparameters so we get a better image of each model's abilities, and see what parameters get the best result. In addition, we used each model for binary classification (positive or negative review) and multi-label classification (0-4 rating

scale), in order to find whether models behave differently in each task.

For our secondary objective, we choose the best model according to the evaluation results, and setup a mock experiment, using Speech-To-Text, the chosen model, and the miLAB robot. This experiment simulates the human-computer interaction we seek to develop.

2.2 Design

We wrote the code in Google Colab in order to use GPU. Our dataset consists of Amazon customers' reviews, one for the multi-label case, which we got from ARHAM RUMI, Kaggle (https://www.kaggle.com/arhamrumi/amazon-product-reviews), and the other one for the binary case, which we got from Mohd Abdul Azeem, Kaggle (https://www.kaggle.com/muhammedabdulazeem/amazon-electronics-items-reviews).

We had technical difficulties with our GPU, since its use was limited by Google Colab. Therefore, in order to overcome it, we had to use smaller parameters in our cases, which prevented us from testing bigger values and the use of better parameters, in some cases.

In each notebook, we created a model for each case. The BERT models were trained for 1 to 2 hours, while the LSTM models were trained for more than 2 hours each.

LSTM			
CASE	CLASS.	EPOCHS	TRAINING TIME (in hours)
1	Binary	2	01:52
2	Binary	4	03:55
3	Multi	2	04:25
4	Binary	2	01:54

BERT			
CASE	CLASS.	EPOCHS	TRAINING TIME (in hours)
1	Binary	2	01:31
2	Binary	4	03:17
3	Multi	2	00:54*
4	Binary	2	01:36

3 Experimental results

We randomly split each dataset to train and test sets (80% and 20% in BERT models, 85% and 15% in LSTM models).
For BERT, we tested 4 cases, while using different task (binary/multi-label), different number of epochs, and different learning rate. We wished to test different batch sizes as well, however, we were limited by the GPU. Also, in case 3 we had to use half of the dataset because of GPU.

For LSTM, we tested 4 cases, while using different task (binary/multi-label), and different number of epochs.

The Rule-Based model does not need to be trained and there are no parameters needed to be used for the model, therefore, there are only two cases of Rule-Based classification - binary and multi. For the Rule-Based model, we used the well-known VADER model (https://github.com/cjhutto/vaderSentiment).

EVALUATION ACCURACY				
	CASE 1	CASE 2	CASE 3	CASE 4
LSTM	86.60%	87%	50.10%	86.90%
BERT	67.18%	67.18%	27.55%	84.80%
RULE-BASED	BINARY CASE		MULTI CASE	
	79.44%		31.60%	

4 Discussion

According to the results mentioned above, we can note some insights.

- LSTM produced the best results, better than BERT and Rule-Based. However, we know, from experimenting through the course and researching other works, that BERT tends to produce better results. This difference can be explained by the fact that we had to use smaller parameters, such as batch size, and couldn't run the recommended best parameters for BERT. But when we changed the learning rate, BERT produced similar results to those of LSTM.
- Multi-label classification is a much harder task for those models then binary classification. All three models produced much worse accuracy when training on multi-label dataset then on binary label dataset. A reasonable explanation is that for a fixed number of training examples it is generally easier to perform binary classification than multi-label classification.

- The bigger the number of epochs was, the better results the model produced. We can assume that epoch sizes can increase the accuracy up to a certain limit beyond which you begin to overfit your model, however, we couldn't reach this limit because of GPU limitations. In BERT model, changing the number of epochs didn't change at all, however, this change was very little.
- Small batch size produce better results. In the LSTM model, we checked the results between a batch size of 32 and 64, and the smaller one yielded the better accuracy. This matches what we read in some papers.
- Rule-Based model performed very well because our dataset was pretty stable with no drastic differences. However, we expect for this model to perform worse on a new and different dataset because it has a limited set of rules which, in this specific model, can't be expanded and we can't fine-tune this model as well.
- Changing the learning rate in BERT models drastically improve the results, as shown in case 4. Using the optimal batch size and epochs can make an impact on the results as well.

5 miLAB Robot Experiment

A video of our experiment using KIP in miLAB: https://youtu.be/Lvfol9lSAjc

The experiment's goal is to let people try and speak in a welcoming or non-welcoming manner to a non-humanoid robotic object (KIP) with only its gesture reactions as a meter to measure their success. This experiment used the best model we've found in this project, LSTM, and uses also a speech-to-text technology.

The user will stand in front of KIP and greet him by saying a sentence. Then KIP will react by turning to the user:
The more KIP is confident that the user was unwelcoming the less it'll face him.
The more KIP is confident that the user was welcoming the more it'll face him.

For more information about KIP: http://milab.idc.ac.il/teaching/projects/kip/.

6 Future Work

Since our project met some limitations, technical mainly, we couldn't expand our research to large scale targets. However, we can suggest what can be done in future work. Our experiment showed us the abilities of NLP as a medium of communication between human and technology.

- Find other, and more accurate, hyper-parameters as well as the optimal known parameters for the training phase in each model, such as batch size, number of epochs, etc.
- Test the models on bigger datasets and different type of datasets that can be relevant to this kind of task.
- Create a hybrid model that takes into account the result of each of the three models we've checked in this project.
- Expand our experiment with KIP - test it in a public place.

7 Code

Rule-Based notebook
https://colab.research.google.com/drive/1MhdBuuk_GTrv5mDliwxA5AvOkqbT9wjQ?usp=sharing

LSTM notebook
https://colab.research.google.com/drive/1ppsiwV8DuFsXSqjwqVqV03MXC0Yzn8Ly?usp=sharing

BERT notebook
https://colab.research.google.com/drive/1SnX7k_vVd83YH27GjWp-r1kvrksJG51h?usp=sharing

8 References

Rule-Based guide
https://towardsdatascience.com/lovecraft-with-natural-language-processing

LSTM guide
https://analyticsindiamag.com/how-to-implement-lstm-rnn-network-for-sentiment-analysis/

BERT guide
https://skimai.com/fine-tuning-bert-for-sentiment-analysis/

LSTM hyperparameters
https://publikationen.bibliothek.kit.edu/1000121378 Speech-to-Text guide
https://realpython.com/python-speech-recognition/