Twitter sentiment analysis using a fine-tuned DistilBERT model

Lazar Jelić

January 5, 2021

1 Dataset

Exploration. Dataset comes with a validation split of 91/9 for 44955 tweets in total. It can be observed from the image below that the data is quite balanced, meaning that, to create a model that generalizes well, there is no need to generate new training samples or to use weighted loss function.

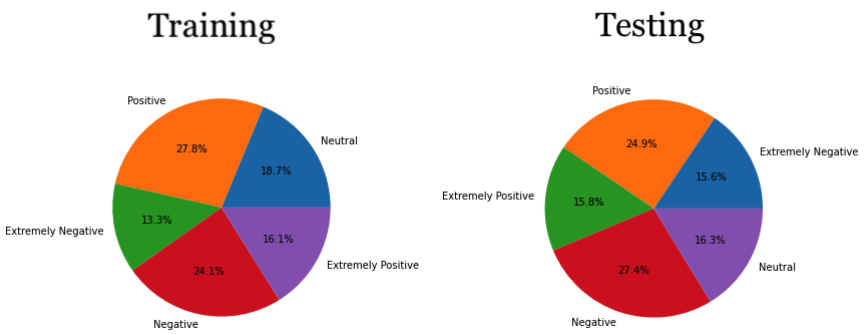

Figure 1: Dataset composition

The image below depicts the most frequent words present in tweets grouped by a class. Interestingly enough, it can be concluded that words such as *"supermarket"* and *"grocery store"* are highly correlated with a tweet being classified as positive. On the other hand, word *"people"* is one of the most frequent words in extremely negative tweets.

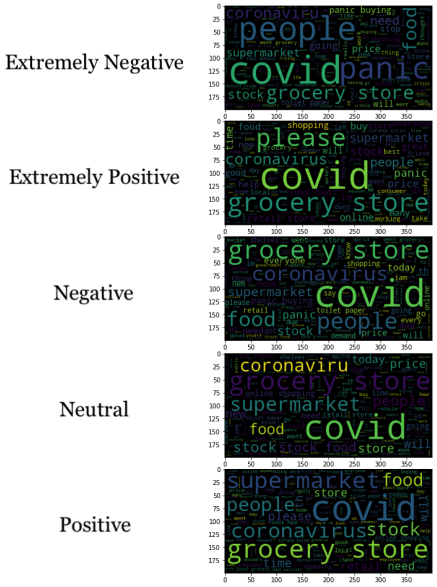

Figure 2: Most frequent words grouped by a class

Hashtag ratio, which is calculated by diving a total number of hashtags by a total number of tweets in a certain class, is shown below. Neutral tweets tend to have more hashtags.

Extremely Negative	0.16722972972972974
Extremely Positive	0.21035058430717862
Negative	0.21805955811719502
Neutral	**0.31179321486268174**
Positive	0.21013727560718057

Table 1: Hashtag ratio

Preparation. Dataset preparation phase involves basic formatting of tweets stored in *Pandas DataFrame* objects mapped to five classes (*Extremely Negative*, *Extremely Negative*, *Negative*, *Neutral*, *Positive*). Text formatting is done in a specific order which is described as follows.

- Store original tweet for data exploration purpose
- Remove unused columns to save memory
- Unescape HTML tags
- Remove URLs
- Remove numbers

- Separate words in hashtags
- Convert tweet to lowercase representation
- Convert UTF-8 charset to ASCII
- Compress 3 or more character occurrences to a length of 2

2 Models

Training. The table below shows the total training time per model. Reason for such a long training process of DistilBERT [2] 5-class classifier is the usage of hyperparameter search.

LogisticRegression	40 seconds
DistilBERT 5-class	2 hours 46 minutes
DistilBERT 3-class	7 minutes 14 seconds

Table 2: Training time per model

Hyperparameters. Instead of doing a grid search and manually testing each combination of hyperparameters, in this case, automated hyperparameter search was performed using Optuna [1] library over 20 trials which resulted in a set of the following hyperparameters.

Learning rate	9.654260494767733e-05
Training epochs	3
Training batch size	64
Random seed	22

Table 3: Optuna generated hyperparameters

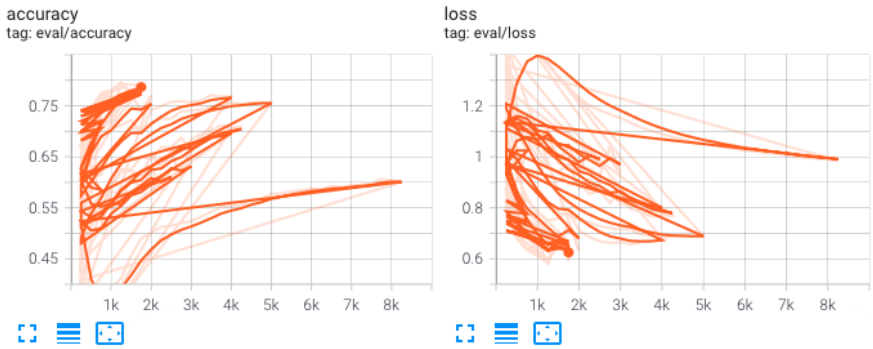

Figure 3: Optuna hyperparameter search

3 Results

Metrics. Accuracy, precision, recall and F1-score are used to measure models' performances. Classification reports, as well as confusion matrices, are shown below.

3.1 LogisticRegression

	precision	recall	f1-score	support
Extremely Negative	0.60	0.41	0.49	592
Extremely Positive	0.67	0.50	0.57	599
Negative	0.49	0.49	0.49	1041
Neutral	0.58	0.63	0.61	619
Positive	0.46	0.60	0.52	947
accuracy			0.53	3798
macro avg	0.56	0.53	0.54	3798
weighted avg	0.54	0.53	0.53	3798

Table 4: LogisticRegression classification report

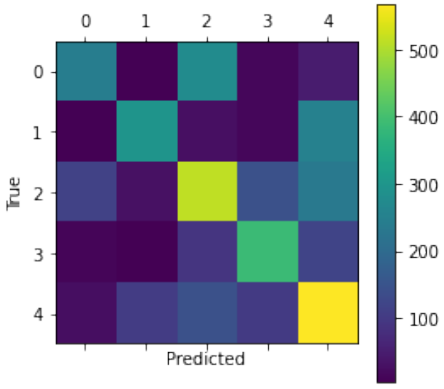

Figure 4: LogisticRegression confusion matrix

3.2 DistilBERT 5-class

	precision	recall	f1-score	support
Extremely Negative	0.83	0.80	0.81	592
Extremely Positive	0.88	0.80	0.84	599
Negative	0.76	0.78	0.77	1041
Neutral	0.86	0.80	0.83	619
Positive	0.73	0.80	0.77	947
accuracy			0.79	3798
macro avg	0.81	0.80	0.80	3798
weighted avg	0.80	0.79	0.80	3798

Table 5: DistilBERT 5-class classification report

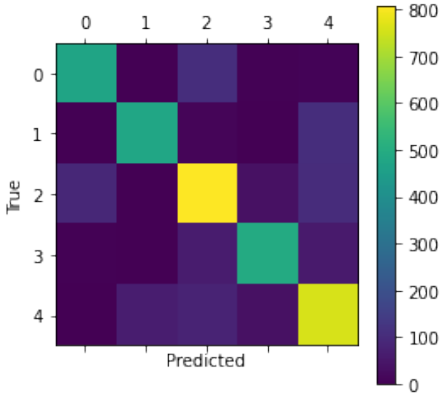

Figure 5: DistilBERT 5-class confusion matrix

3.3 DistilBERT 3-class

	precision	recall	f1-score	support
Negative	0.88	0.89	0.89	1633
Neutral	0.87	0.78	0.82	619
Positive	0.87	0.90	0.88	1546
accuracy			0.88	3798
macro avg	0.87	0.86	0.86	3798
weighted avg	0.88	0.88	0.87	3798

Table 6: DistilBERT 3-class classification report

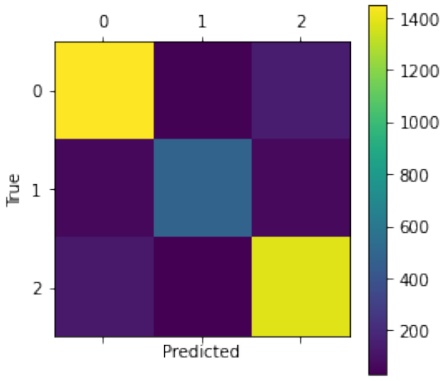

Figure 6: DistilBERT 3-class confusion matrix

4 Resources

Dataset: `https://www.kaggle.com/datatattle/covid-19-nlp-text-classification`

Code: `https://github.com/jelic98/raf_du/tree/main/homework_2`

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

[2] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.