Sentiment Analysis on SST-5 Fine-grained classification

Ariel RAMOS

May 30, 2022

***Abstract*—Sentiment analysis is an approach to Natural Language Processing whose task consists of categorising opinions from a text and thus, determining the writer's attitude towards a product, service or idea. In the present report, the RoBERTa Deep Learning architecture as well as the Random Forest ensemble learning method were utilised and evaluated for predicting the classes of movie reviews in the SST-5 (Stanford Sentiment Treebank) dataset.**

I. Introduction

A. Overview

The objective of this research report is to analyse and have an insight of the use of different approaches for multi-class classification. Indeed, the current dataset is referred to as SST-5 or SST fine-grained because it comprises 5 classes: negative, somewhat negative, neutral, somewhat positive and positive.

Therefore, two implementation approaches were proposed:

- **RoBERTa:** a transformer model that belongs to deep neural network models.
- **Random Forest classifier:** a meta estimator that fits several decision tree classifiers for improving accuracy and handling over-fitting.

For each approach, the preprocessing of the data and feature extraction will be explained as well as the architecture of the Deep Learning model for RoBERTa. Besides, the results obtained with both classifiers will be shown along with the drawbacks and advantages of each one.

B. Evaluation Metric

The performance of both models will be assessed using F1, recall and accuracy having as a reference the results obtained in the SST-5 Fine-grained classification leaderboard.[1] Additionally, the objective of this work is to obtain a competitive accuracy by employing a custom neural network architecture.

II. Data Analysis

A. Dataset description

The training dataset comprises 8544 movie reviews while the validation and testing datasets consist of 1101 and 2210 reviews respectively (all with their labels).

From Figure 1 it can be observed that the distribution of labels is unbalanced which can induce the models to predict the class that has a larger number of samples. Thus, oversampling (minority classes) and undersampling (majority classes) techniques were utilised to overcome this problem in the training dataset. Nonetheless, it was only used for the Random Forest Classifier since its baseline predicted only samples that belonged to 3 classes.

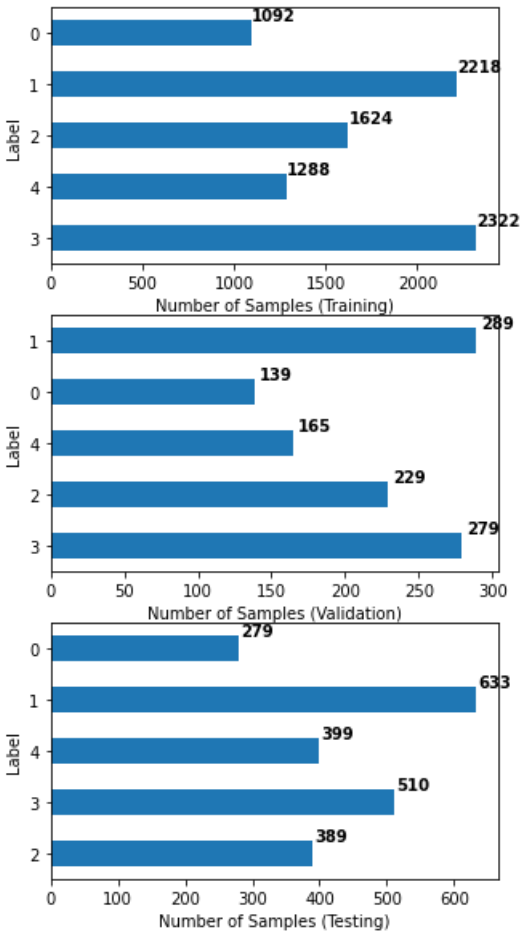

Fig. 1. Distribution of the SST-5 Dataset.

III. Random Forest

A. Preprocessing

As mentioned in the previous section, the training dataset was oversample and undersampled independently to deal with unbalanced data. Subsequently, the following preprocessing steps were taken:

- **Step 1 :** Lowercase each review to standardise the data.
- **Step 2 :** Remove special characters and punctuation.
- **Step 3 :** Strip multiple white spaces.

[1] https://paperswithcode.com/sota/sentiment-analysis-on-sst-5-fine-grained

- **Step 4 :** Remove stopwords with the help of the NLTK library and apply stemming OR lemmatization as shown in Figure 2.

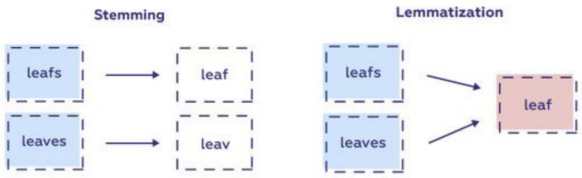

Fig. 2. Illustration of stemming and lemmatization

B. Word Embeddings

For obtaining word vectors of reviews, the Word2Vec algorithm from gensim[2] was employed. It was used to extract the notion of relatedness across words. Besides, the skip-gram model was selected since it generally works better with small datasets and it better represents less frequent words according to [2].

Subsequently the vector representation of each review was computed by averaging the vectors of all words contained in it.

C. Hypertuning parameters

For setting the optimal parameters of this classifier, the GridSearchCV functionality from sklearn was exploited. The parameters to tune along with their values were:

- n estimators: 50, 55, 61, 66, 72, 77, 83, 88, 94, **100**.
- max features: **auto**, sqrt.
- max depth: 2, **4**.
- min samples split: **2**, 5.
- min samples leaf: 1, **2**.
- bootstrap: **True**, False.

The bold values were the ones that achieved the highest accuracy when tested using the validation set.

D. Results

For analysing the performance of the classifier, the confusion matrix of its predictions (testing set) was computed. As shown in Fig 3 and 4, the classifier obtained a better accuracy when the training dataset was oversampled.

Additionally, Table I and II display the evaluation metrics obtained when running the classifier (with optimal parameters) on the testing dataset.

E. Discussion of results

From the confusion matrices it can be observed that the model has problems predicting the reviews belonging to classes 0,2 and 4 in both scenarios when the training dataset has been oversampled or undersampled. This is because of the lack of samples and diversity that these classes have in

[2]https://radimrehurek.com/gensim/models/word2vec.html

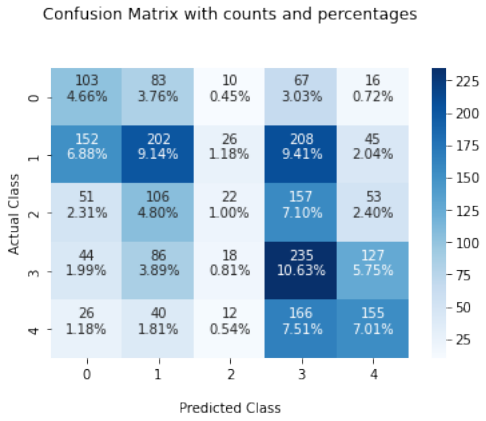

Fig. 3. Confusion matrix - oversampled dataset

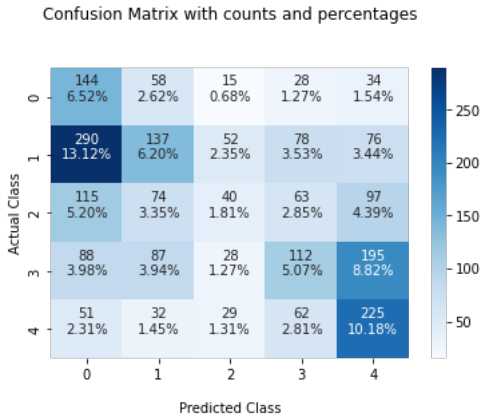

Fig. 4. Confusion matrix - undersampled dataset

comparison with classes 1 and 3 where the accuracy is higher. In spite of this, the results achieved are better than using the initial unbalanced dataset since it implies a poor predictive performance for the minority classes.

Likewise, it is interesting to analyse that the presence of the word 'not' in reviews e.g 'not interesting', affects the performance of the classifier given that it is considered a stop word and thus, removed during the preprocessing phase of data.

In terms of time complexity, the algorithm does not take a long time to train. Indeed, what takes longer is the hypertuning of parameters due to the grid search approach. Hence, the model is scalable and straightforward to set up. The drawback of the current approach relies on the obtained accuracy (30 %). It is not competitive with other approaches that have been used for the SST-5 dataset. Nonetheless, Random Forests are still a choice that pushes to the limit the interpretability-accuracy trade-off which is not the case with neural networks as it will be explained in the next section.

IV. RoBERTa

A. Background

RoBERTa (Robustly Optimised BERT Pretraining Approach) is a transformer model pretrained on a large corpus of

Label	Precision	Recall	f1-score	support
0	0.27	0.37	0.31	279
1	0.39	0.32	0.35	633
2	0.25	0.06	0.09	389
3	0.28	0.46	0.35	510
4	0.39	0.39	0.39	399
accuracy			0.32	2210
macro avg	0.32	0.32	0.30	2210
weighted avg	0.33	0.32	0.31	2210

TABLE I
PERFORMANCE OF THE RANDOM FOREST CLASSIFIER USING AN OVERSAMPLED DATASET.

Label	Precision	Recall	f1-score	support
0	0.21	0.52	0.30	279
1	0.35	0.22	0.27	633
2	0.24	0.10	0.14	389
3	0.33	0.22	0.26	510
4	0.36	0.56	0.44	399
accuracy			0.30	2210
macro avg	0.30	0.32	0.28	2210
weighted avg	0.31	0.30	0.28	2210

TABLE II
PERFORMANCE OF THE RANDOM FOREST CLASSIFIER USING AN UNDERSAMPLED DATASET.

English data that has managed to improve the state-of-the-art scores in a variety of NLP tasks. It is built on top of BERT, having as the main difference, the removal of the next-sentence pretraining objective and the fact that it is trained with larger mini-batches and learning rates.

For the present project, the pretrained models provided by the **Huggingface library** [3] were utilised and modified accordingly to solve the current classification task.

B. Model architecture

The model was inspired by [1]. Similar to a traditional classifier it fits the summaries to the labels sent as inputs. The key difference is that the transformer first performs the encoding of the corpus based on its tokenizer and subsequently it realises a standard forward pass on the neural network followed by an optimization step on cross entropy loss.

The main architecture can be summarised as follows:

- Roberta Model from pretrained 'roberta-base'
- Dropout layer with rate 0.3
- Linear layer from 768 to 64
- Normalization layer
- Dropout layer with rate 0.3
- Linear layer from 64 to 5 (number of classes)

The inputs for the Roberta Model are the following:

- The input ids from movie reviews which were computed by the Roberta tokenizer (also from 'roberta base').
- The attention mask list that indicates which tokens should be attended to, and which should not. This is useful when the reviews are shorter than 512 tokens.

C. Hyperparameters

For better performance, some parameters were tuned according to several combinations:

- Linear transformations: (768 to 64 and 64 to 5), (768 to 128 and 128 to 5), (768 to 256 and 256 to 5).
- Learning rate: 1e-05, 1e-06, 1e-07.
- Dropout rate: 0.1, 0.2, 0.3

The best accuracy obtained was 53%. It was achieved by employing the following combination of hyperparameters: linear transformations (768 to 64 and 64 to 5), learning rate 1e-05 and dropout rate 0.3.

Likewise, in order to stop the algorithm from overfitting, the number of epochs is only increased if the validation loss has decreased.

D. Results

From Figure 5 it can be observed that the percentages along the main diagonal are higher meaning that the predicted labels significantly agree with the actual labels.

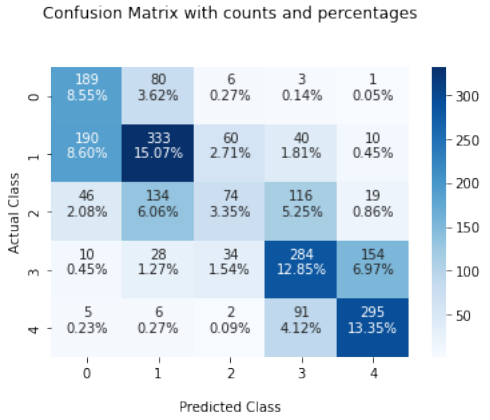

Fig. 5. Confusion matrix using RoBERTa model

Furthermore, from Table III it can be seen that the accuracy obtained actually shows what the confusion matrix displays. The model reached 53% which largely outperforms the Random Forests classifier.

Label	Precision	Recall	f1-score	support
0	0.43	0.68	0.53	279
1	0.57	0.53	0.55	633
2	0.42	0.19	0.26	389
3	0.53	0.56	0.54	510
4	0.62	0.74	0.67	399
accuracy			0.53	2210
macro avg	0.51	0.54	0.51	2210
weighted avg	0.53	0.53	0.52	2210

TABLE III
PERFORMANCE OF THE ROBERTA MODEL.

V. CONCLUSIONS AND FUTURE WORK

From the Random Forests and RoBERTa models, it can be seen that both have different advantages and disadvantages. On one hand, the Random Forests approach takes less time to train and up to some point, the predictions of labels can be explained and thus, be interpretable. However, the obtained accuracy is not even close to the state-of-the-art approaches in the SST-5 leaderboard.

On the other hand, the RoBERTa model achieved a much higher accuracy (53%) that is even competitive with the top accuracies in the leaderboard. Nonetheless, it takes longer to train. In fact, without the Pro functionality of Google Colab, it would have been really difficult to test it given the limiting GPU and memory resources. Besides, the use of neural networks is still a black box which means that the predictions made by the model can not be explained. This in turn affects its practicality given that in an industry environment like banks, decisions might need to be explained to customers e.g when asking for a denied loan.

With respect to future improvements, a different approach to obtain word embeddings could be employed such as GloVe. In addition, further preprocessing could be performed to reviews to deal with negation e.g 'not good' and contractions. Likewise, for the RoBERTa model, the 'roberta-large' base can be utilised to further increase the accuracy of the model. For doing this, more GPU and memory resources will be needed which is why it could not be attempted by the present work.

Finally, the hypertuning of parameters is still a big challenge that needs to be solved to avoid its manual setting and to reduce the amount of effort and time, especially if a Grid Search approach is applied.

REFERENCES

[1] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. cite arxiv:1907.11692.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[3] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.