NLP Coursework:Patronising and Condescending Language Detection

Elizabeth Bates
ejb121@ic.ac.uk

Venus Cheung
vwc21@ic.ac.uk

Mathilde Outters
mo220@ic.ac.uk

1 Introduction

The aim of this project is to implement a transformer-based model from Huggingface to predict whether a given text is patronising / condescending or not (Perez Almendros et al., 2020). The dataset consists of labeled text paragraphs, extracted from news articles. In order to exceed the performance of the baseline in F_1 score (0.48) with RoBeRTa, our primary focus was to address the issue of having a small unbalanced dataset. The F_1 score measures the classification accuracy for the "patronising" class on test data; it's the harmonic mean of the recall and precision values (Sasaki et al., 2007).

Robustly Optimized BERT Pre-training Approach (RoBERTa) is model relies on BERT network architecture, with an improved pre-training methodology and a byte-level BPE tokenizer. The final F_1 score was improved from the baseline 0.48 to 0.551 through various techniques like adjusting the training dataset balance, using resampling methods like downsampling, upsampling, and performing data augmentation.

2 Data analysis

Analysing of the class labels revealed a strong imbalance in the raw data; 10% of the paragraphs were patronising in both the training set (794 vs 7581) and the official dev set (199 vs 1895).

It was found that the class labels correlates with some features of the data, like the input length. Longer sequences (> 60 words) were more likely to be patronising when compared to shorter sequences (< 35 words): the former percentage being 13.9% and the later 7.3%. Sentence length is further discussed in section 4.2 and in Table 5.

Qualitative assessment of the dataset must be considered given the subtle nature of the task. It is well-known that data quality is key in any supervised learning framework. As emphasised by the authors of the original paper (Perez Almendros et al., 2020), patronising and condescending language (PCL) is more subjective than the other types of discourse typically targeted in natural language processing (NLP) classification. We could find positive examples for which we disagreed with the provided label e.g. "*Mother of three on the brink of being homeless again*" (par_id 2226). To us, it seemed that some paragraphs labeled as "patronising" like the one aforementioned are purely stating facts with no condescending language towards vulnerable communities. It should be noted that the two expert annotators did not exactly agree on the label for 16% of the paragraphs and were in total disagreements, with requirement of a referee final decision for 6%. Other authors (Wang and Potts, 2019) have highlighted the need for high-quality dataset annotated by experts for such difficult tasks.

3 Modelling and Results

It was decided to implemententent gradual pre-processing methods to the baseline model, and each time compute the score using a RoBERTa pretrained model which is case sensitive (Liu et al., 2019). The hyperparameter like learning rate and epochs were experimented with at each stage of testing, though it was continuously found that a learning rate of 4e-5 and 1 epoch gave best results. The tuning was done based on the model's performance on the official dev set at each stage.

3.1 Training on raw unbalanced dataset

The outcome of training the model on the raw training dataset produced an F_1 score of 0, all samples were predicted to be non-patronising. The imbalance in the training data was such that it caused the model to simply ignore minority samples.

3.2 Downsampling majority class and tuning class balance

The ratio of labels in the dataset is a pre-processing hyperparameter to control class imbalance. Our first approach to tune this ratio was to downsample the majority class (non-patronising) in the training data. This dramatically decreased

Table 1: Table of varying F_1 scores as the ratio of patronising:non-patronising data is altered, after having downsampled the non-patronising sentences in the training set

Data ratio	F_1 scores over 3 model trials			Average F_1 score
	1	2	3	
1:1	0.449	0.428	0.457	0.44
1:2	0.417	0.511	0.434	0.45
1:3	0.502	0.521	0.520	**0.51**
1:4	0.493	0.447	0.498	0.48
1:5	0.485	0.461	0.263	0.40
1:6	0.412	0.391	0.452	0.41

the size of the training dataset. The following ratios were tried: 1:1, 1:2, 1:3, 1:4, 1:5, 1:6, as summarized in Table 1. The ratio of 1:3 for our final model, which gave best F_1 score on the official dev set.

3.3 Upsampling minority class and tune class balance

In order to address the concern resulting from the previous section of training on a small dataset, the minority class (patronising data) was upsampled. This was done in a trivial way, by doubling the amount of patronising samples in the training set and then training a model using various different data balances, see table 2. Again, it can be seen that the best F_1 score obtained was from a 1:3 ratio of patronising:non-patronising training data. The general trend for the average F_1 scores as the ratio increases shows that F_1 scores are slightly higher than in table 1. This is likely as a result of the trial models having much more training data to learn from, and hence is able to better differentiate between patronising and non-patronising text.

3.4 Data Augmentation

As a result of the better F_1 values obtained from upsampling the training dataset, it seemed like a good progression of this would be to augment the upsampled data using the synonym replacement method. This process would create a more diverse training set than in section 3.3 for the model to learn from. The main method used for the data augmentation in the code is using EDA (Easy Data Augmentation) (Wei and Zou, 2019), which contains a hyperparameter “alpha_sr” which can be adjusted to alter the amount of word replacements in the sentence, i.e alpha_sr = 0.1 indicates that 10% of the non-stopwords in the text will be replaced with a synonym. The alpha_sr value was trialed over a few values, see table 3 and the best F_1 score (0.56) found was when tested on the dev set using alpha_sr = 0.1. This score was better than of that obtained in section 3.3 where the upsampled data had not been augmented, indicating this was a worthwhile technique.

Table 2: Table of varying F_1 scores as the ratio of patronising:non-patronising data is altered, after having upsampled the patronising sentences in the training data

Data ratio	F_1 scores over 3 model trials			Average F_1 score
	1	2	3	
1:1	0.404	0.455	0.463	0.44
1:2	0.440	0.493	0.496	0.48
1:3	0.555	0.521	0.522	**0.53**
1:4	0.381	0.456	0.471	0.44

Table 3: Table of varying F_1 scores as the alpha_sr value increases, for a model with a 1:3 (patronising:non-patronising) data balance

alpha_sr	F_1 scores over 3 model trials			Average F_1 score
	1	2	3	
0.05	0.531	0.538	0.521	0.53
0.1	0.552	0.589	0.536	**0.56**
0.25	0.103	0.119	0.152	0.13

This process does however have its own issues, mainly in that some of the words being replaced have multiple meanings in different contexts, and once replaced by the given synonym, the sentence doesn’t quite make sense. I.e If “ran” in the phrase “The program ran well” is replaced with “sprinted” then the sentence no longer reads as semantically correct. It can be seen in table 3 that when alpha_sr = 0.25, the F_1 average was much lower. It seems likely that this is as a result of the aforementioned problem in combination with key patronising words potentially being replaced with words that are less condescending in the context.

3.5 Results on official test data

The model which performed the best on the dev data set according to the various methods tried was a RoBERTa pretrained model, finetuned with a patronising:non-patronising data balance of 1:3 and having augmented both classes of the training data. Therefore this model was used to predict the labels of the official test dataset. The predictions were submitted to CodaLab under the username lb8s and resulted in an F_1 score of 0.551.

4 Analysis

4.1 To what extent is the model better at predicting examples with a higher level of patronising content?

In order to see how the model performed on higher levels of patronising content than the balance given in the training and dev set, another smaller dataset was made from the dev set containing the full 199 patronising text samples and only 199 of the non-patronising, making it a balanced set. The accuracy of the model was calculated for this ratio and ones of increasing patronising, see table 4. In this case, accuracy was a more relevant score to look at and it can be seen that the more patronising content to non patronising content in the test set, the worse the model is at correctly predicting the labels. This is plausible though since the model trained on a training set that had three times more non-patronising content than patronising, and hence is better at predicting the non-patronising cases.

4.2 How does the length of the input sequence impact the model performance? If there is any difference, speculate why.

It was found that the average number of words for each sample input text in the test set was 47. It was then decided to take two subsets of the test set, keeping only sentences of more than 60 words in one and sentences of less than 35 words in the other. The class balance was not affected too much in either subset, so the F_1 score is still a useful comparison of the real performance. As mentioned in section 2, more of the longer sentences are labeled as patronising than the shorter sentences, in terms of ratio and quantity, see table 5. Due to the model being trained on more patronising than non-patronising data, it makes sense that the average F_1 score for shorter sentences is much higher than that of the longer sentences and this is reflected in the results of the table.

Table 4: Table of decreasing accuracy scores as the ratio (patronising:non-patronising) increases

Ratio	Accuracy
1:1	0.763
2:1	0.694
3:1	0.664
4:1	0.641
5:1	0.626

Table 5: Table of varying F_1 scores and amount of sentences for sentences of lengths larger than 60 or shorter than 35

Sentence length (words)	Ratio	No. of sentences	Average F_1
<35	54:686	740	0.54
>60	67:414	481	0.45

4.3 To what extent does the categorical data provided influence the model predictions?

To break down the types of PCL further, seven categories were used to label each paragraph in the dataset (Perez Almendros et al., 2020). The categories could allow models to gain additional information about types of PCL, and help improve performance during training. Additionally, a "score" label from 0-5 was given for each piece of text derived from the combined scores given from both annotators, with a higher score indicating a stronger element of PCL portrayed in the text. For this task, we have labelled a piece of text as positive for PCL if the combined score is 2 or above. This is the lowest score where both the annotators agreed that a given sentence was patronising, and even this resulted in a training set that was far more unbalanced in the amount of non-patronising data present. If scores of 3 (or higher) were taken, it's likely that even fewer patronising samples would be present in the train set, and hence the model used would be less able to accurately predict patronising text due to lack of data.

5 Conclusions

To conclude, a gradual improvement of the model performance can be seen throughout the experimentation stages of this project. First more data with over-sampling, then more diverse data thanks to data augmentation. To improve this more, the next step would be to use a more powerful sentence classification model than RoBERTa. We came across DeBERTa (Decoding-enhanced BERT with disentangled attention) in the literature (He et al., 2020). This model improves the BERT and RoBERTa models using two key techniques: disentangled attention mechanism and enhanced mask decoder.

6 Code Link

https://colab.research.google.com/drive/1f53WHpvVYYuDRaiAxLsxpwOQXvJhIsGz?usp=sharing

References

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Carla Perez Almendros, Luis Espinosa Anke, and Steven Schockaert. 2020. Don't patronize me! an annotated dataset with patronizing and condescending language towards vulnerable communities. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5891–5902, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yutaka Sasaki et al. 2007. The truth of the f-measure. 2007. *URL: https://www. cs. odu. edu/~ mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07. pdf [accessed 2021-05-26]*.

Zijian Wang and Christopher Potts. 2019. Talkdown: A corpus for condescension detection in context. *CoRR*, abs/1909.11272.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China. Association for Computational Linguistics.