Natural Language Processing Coursework 2024

Freddy Jiang
Imperial College London
ffj20@ic.ac.uk

Jian Zhao
Imperial College London
jz4120@ic.ac.uk

Abstract

We develop a binary classification model to address the task of identifying patronising and condescending language (PCL) in text (Perez Almendros et al., 2020). We explore various improvements on the RoBERTa base model (Liu et al., 2019), including data augmentation, ensemble methods and various preprocessing techniques. Our final ensemble model achieves an F1-score of 0.5731 on the official dataset, outperforming RoBERTa base by 9%. We analyse the performance of the model on different types of text to identify its strengths and weaknesses.[1]

1 Introduction

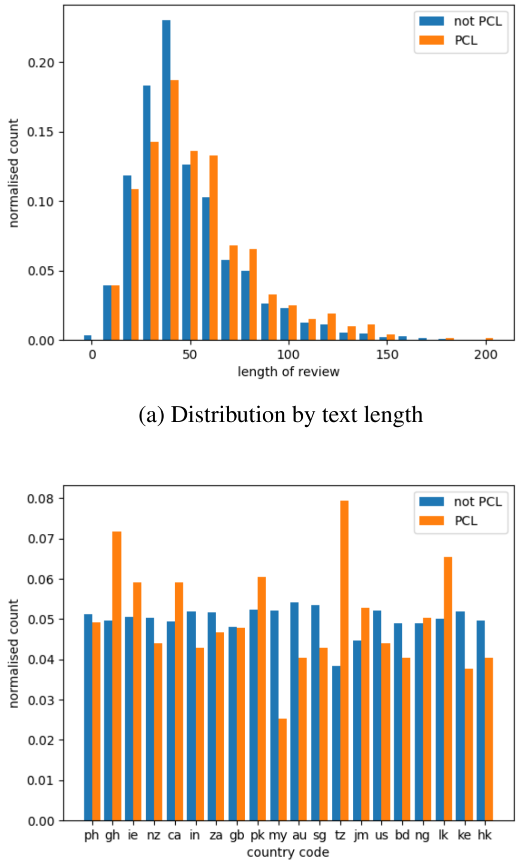
Natural language processing has made significant advancements in recent years, but it encounters difficulty with certain tasks. In this paper, we tackle the task of determining whether a piece of text uses patronising and condescending language (PCL) by developing a binary classification model. The dataset is described in the task paper (Perez Almendros et al., 2020), which also details the criteria for classifying a piece of text as PCL. The detection of PCL is of interest since PCL "makes it more difficult for vulnerable communities to overcome difficulties." This task appeared as Task 4.1 in the SemEval 2022 competition.

We start with the RoBERTa base model provided by FacebookAI (Liu et al., 2019) and explore improvements including data augementation and ensembling to achieve better performance.

2 Data Analysis

2.1 Quantitative Analysis

The provided dataset for PCL detection is heavily imbalanced, with the training set having 794 positive[2] examples and 7581 negative examples.



(a) Distribution by text length

(b) Distribution by country

Figure 1: Normalised distribution of PCL and non-PCL samples by text length and country.

Figure 1 presents the distribution of PCL and non-PCL samples. We observe that longer samples are more likely to be PCL, although the difference is marginal. We also observe that samples from Malaysia (my) are more likely to be non-PCL compared to other countries, while samples from Tanzania (tz) are more likely to be PCL.

Both the text length and country of origin provide weak indicators of whether a sample is PCL. We conclude that the dominating feature is the text

[1] Code available at https://gitlab.doc.ic.ac.uk/ffj20/nlp-cw.

[2] Text with PCL is considered a positive example.

itself, and use this as the basis for our model.

2.2 Qualitative Analysis

In the original dataset, samples are labelled with a score from 0 to 4 where a larger score indicates a larger patronising or condescending tendency in the language used. These scores are then mapped to a binary classification where 0-1 are classified as non-PCL and 2-4 are classified as PCL.

In order to assess how subjective the task is, we find example text that have scores 1 and 2, i. e. on the classification boundary. The chosen examples are regarding the "disabled" category and are presented in Appendix A.

Example A.1 (score: 2) talks about an author publishing a book featuring the works of ill or disabled artists globally, alongside her own works, to showcase their talent. Referring to Perez Almendros et al. (2020), we hypothesise that this sample has its score due to "the privileged community being presented as saviours of the vulnerable community." However, an argument can be made that the text puts the vulnerable community on equal footing with the privileged community, since their works are placed side by side and the word choice is respectful, a direct contrast to the PCL feature of "creating a difference between 'us' and 'them'".

Example A.2 (score: 1) discusses a doctor's assurance, given in an interview, that marginalized communities like the disabled, will receive equal opportunities within the education system. Referring to Perez Almendros et al. (2020), an argument can be made that this sample also has the undertone of "the privileged community being presented as saviours of the vulnerable community," since disabled people are depicted as needing to be given equal opportunities by the privileged community.

These examples demonstrate that the task of classifying text as PCL or non-PCL is subjective. Therefore, we expect the model to have difficulty in classifying text, particularly those with weak patronising tendency.

3 Setup

RoBERTa (Liu et al., 2019) is a BERT-based transformer model that achieves state-of-the-art performance on NLP tasks. We will use RoBERTa and DistilBERT (Sanh et al., 2019), a smaller distilled version of RoBERTa, as our model foundation.

3.1 Training Data

We partition the PCL training set with an 80:20 split into a internal training set and an internal dev set. We use the internal dev set for validation and the official labelled dev set as our test set.

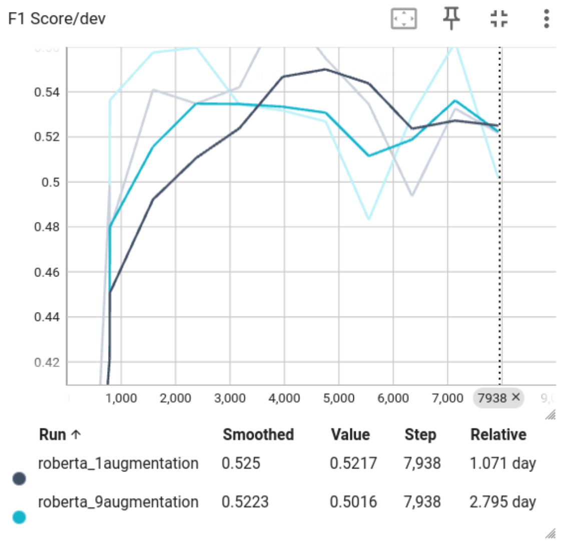
The provided labels range from 0 to 4. As described by Perez Almendros et al. (2020), we preprocess the labels by mapping 0-1 to 0 (non-PCL) and 2-4 to 1 (PCL) for binary classification.

Following our quantitative analysis in Section 2.1, we conclude the dominating feature of PCL to be the text itself. We use the text as the sole input feature. We experiment with cased and uncased models, as well as removing punctuation. We find minimal difference in performance, and our final setup involves a combination of cased and uncased models with punctuation.

We also experiment with arranging the training set by the text length in terms of word count, starting with shorter samples. The intuition is that for non-pretrained base LLMs, the model may more easily learn features from simpler, shorter samples. However, we find this to have minimal effect on performance and opt not to use this for our final setup.

3.2 Data Sampling and Augmentation

We address dataset imbalance by using a combination of downsampling non-PCL data and upsampling PCL data. Using RoBERTa base, we find the



	Run ↑	Smoothed	Value	Step	Relative
●	roberta_1augmentation	0.525	0.5217	7,938	1.071 day
●	roberta_9augmentation	0.5223	0.5016	7,938	2.795 day

Figure 2: F1-score of RoBERTa base trained with 1 batch and 9 batches of augmentation via translation. Plots are smoothed with an exponential moving average with $\alpha = 0.8$.

optimal mix of downsampling and upsampling to be duplicating the PCL data, followed by downsampling the PCL data to match the amount of non-PCL data.

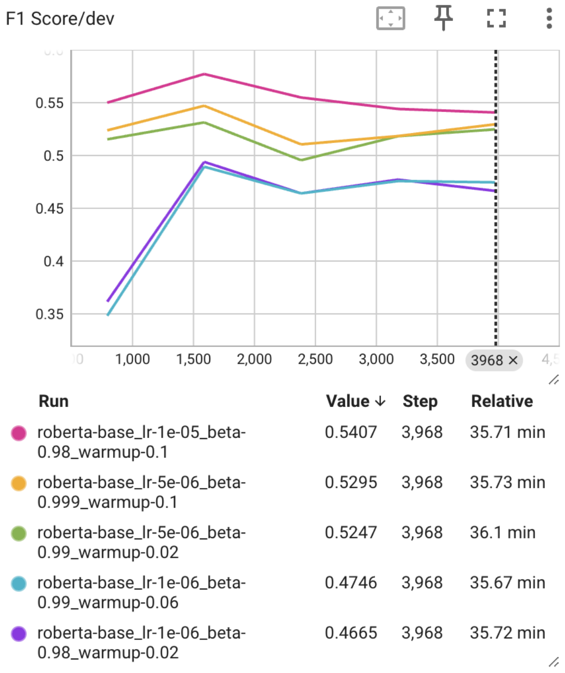
On top of data sampling, we also use data augmentation to further increase the robustness of the model and reduce overfitting to the original data. Augmentation is done via machine translation provided by API calls to Google Translate. The original data is translated into a target language, and then translated back into English. We choose Korean as our target language as it has a different grammar structure to English, thereby introducing more variation. The data augmentation improved the performance of the RoBERTa base from 0.50 to 0.53 on the official dev set. We experiment with augmentation using 9 target languages to get a larger corpus, but it does not result in significant improvement in performance (Figure 2), so we opt to use Korean only.

3.3 Model Architecture

For RoBERTa base, we freeze the tokeniser and pass the training set, then finetune the base encoder on the tokenised training set.

We adopt a similar training procedure as Liu et al. (2019), using AdamW and linear scheduler with warmup. We find using a scheduler to improve performance. We perform hyperparameter tuning on the learning rate, β_2 of AdamW and warmup percentage. Selected results are presented in Figure 3. Table 1 presents our optimal parameter findings.

During training, we save a model checkpoint every epoch and choose the checkpoint with the highest F1-score on the internal dev set. This is a form of early stopping / implicit regularisation.

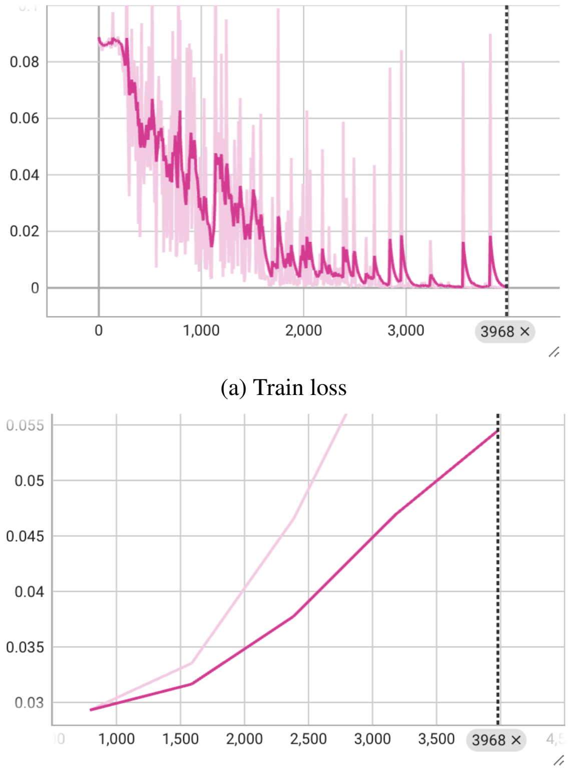


Run	Value ↓	Step	Relative
● roberta-base_lr-1e-05_beta-0.98_warmup-0.1	0.5407	3,968	35.71 min
● roberta-base_lr-5e-06_beta-0.999_warmup-0.1	0.5295	3,968	35.73 min
● roberta-base_lr-5e-06_beta-0.99_warmup-0.02	0.5247	3,968	36.1 min
● roberta-base_lr-1e-06_beta-0.99_warmup-0.06	0.4746	3,968	35.67 min
● roberta-base_lr-1e-06_beta-0.98_warmup-0.02	0.4665	3,968	35.72 min

Figure 3: F1-score of RoBERTa base trained with different learning rates and scheduler settings, with 5 epochs. Using AdamW with linear scheduler with warmup. Evaluated on the internal dev set. For full results, see Appendix B.

Hyperparameter	Value
Warmup Percentage	0.1
Batch Size	8
Learning Rate Decay	Linear
AdamW ϵ	1e-5
Adam β_1	0.9
Adam β_2	0.98
Max Epochs	5

Table 1: Hyperparameters for finetuning RoBERTa and DistilBERT encoders. If a hyperparameter is not listed, it remains unchanged from the base setup. (Liu et al., 2019).



(a) Train loss

(b) Dev loss

Figure 4: Train and dev loss of RoBERTa base over 5 epochs, using parameters in Table 1. Plots are smoothed with an exponential moving average with $\alpha = 0.8$.

Original Model	F1-score
SamLowe/roberta-base-go_emotions	**0.5511**
Seethal/sentiment_analysis_generic_dataset	0.4542
distilbert/distilbert-base-uncased-finetuned-sst-2-english	0.4545
martin-ha/toxic-comment-model	0.3699
mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis	**0.5206**
roberta-base	**0.5407**

Table 2: Internal dev F1-scores of finetuned individual models. The top 3 models are bolded, with the top 2 being RoBERTa models and the other 4 models being DistilRoBERTa.

Figure 4 presents the train and dev loss curves with our optimal parameters. Interestingly, the dev loss increases over time, potentially indicating some overfitting. We have attempted to address this with early stopping, using a scheduler and augmentation techniques.

The base model is a binary classifier. We also use pretrained RoBERTa and DistilBERT encoders trained on other tasks, such as sentiment analysis on financial news[3]. We use the same training procedure and hyperparameters for pretrained models as the base model. For encoders with a multi-class classifier, we add a linear layer on top of the encoder to convert it into a binary classifier, then train both the linear layer and the encoder. See Table 2 and Section 4 for more detail.

4 Ensemble Methods

Ensemble methods involve combining multiple models to improve overall performance (Fattahi and Mejri, 2021; Kanakaraj and Guddeti, 2015). Our approach involves training multiple models, then averaging their predictions[4] at inference time to obtain the final prediction. The intuition is that different models may capture different underlying features of the data, resulting in different strengths and weaknesses. Combining them can result in a more robust classifier.

4.1 Model Training

We train a variety of RoBERTa and DistilBERT models as described in Section 3.3. Table 2 presents the performance of individual models.

We use the setup described by Table 1 for all models.

4.2 Model Selection

We select the top 3 models from Table 2 to form our ensemble classifier. This yields an F1-score of 0.5621 on the internal dev set, a 1.1% improvement from the best individual model.

For the final ensemble classifier, we retrain the 3 models on the entire training set, yielding an F1-score of **0.5731** on the official dev set, with a precision of 0.4723 and recall of 0.7296.

5 Baseline Models

We compare the performance of our final model to two baseline models based on the bag of word (BOW) representation. One of the models uses the Complement Naive Bayes algorithm for predictions, achieving an F1-score of 0.1500 on the official dev set. The other uses the Gaussian Naive Bayes algorithm with data balancing by undersampling, achieving an F1-score of 0.2030 on the official dev set. Our model clearly outperforms these baseline models.

The Gaussian Naive Bayes classifier makes the assumption that all features in the representation are independent and that the data follows a Gaussian distribution in each category. An example of a false positive for PCL given by this classifier is a sample regarding changes to tariff schemes to give relief to poor families (Appendix A.3). We suggest that it is misclassified as PCL because it contains the word “poor”, which is associated with PCL.

6 Analysis

We analyse the prediction of the final model on the results of the official dev set.

We notice that the model is significantly better at identifying PCL in text that have a higher level of patronising and condescending content. We look at the prediction metrics for samples with raw scores from 2 to 4 (placing the sample in the PCL category), and find that the model predicts those with

[3] For ensemble methods

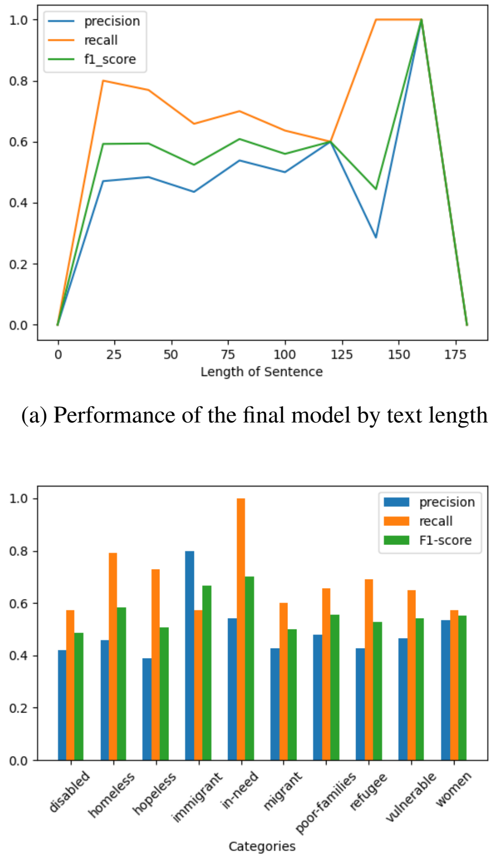
[4] Assume an ensemble classifier with 3 models predicting a 0.3, 0.7 and 0.9. Its average is 0.63, so the final prediction is 1.

higher raw scores at a much higher recall, as shown in Table 3. In all these cases, the model has near perfect precision.

Raw Score	Recall
2	0.3333
3	0.6742
4	0.8587

Table 3: The recall of the final model on official dev data with raw scores from 2 to 4

We investigate the performance of the model on different sentence lengths. As presented in Figure 5, although the model performs similarly in terms of F1-score across all sentence lengths up to 125 words, the recall generally decreases as sentence length increases, while precision increases. We speculate that in shorter sentences, the PCL features are more prominent for the model to pick out, hence giving a higher recall. With longer sentences, PCL may be more difficult to detect where only part of the sentence is PCL, but when a full sentence is PCL, the model is precise in its detection. At lengths above 125 words, there does not seem to be a general trend. Note that the sample size is small here so the performance there is not indicative.



(a) Performance of the final model by text length

(b) Performance of the final model by data category

Figure 5: Normalised distribution of PCL and non-PCL samples by text length and country.

We also investigate the performance of the model according to data categories. As Figure 5 shows, the model performs different in different date categories. We observe that in most categories, the recall is higher than the precision except the "immigrant" category. In general, the model performs the best in the "in-need" category and the worst in the "disabled" category.

Looking at the results of the SemEval 2022 competition, the top models achieve an F1-score above 0.6. These models tend to have similar precision and recall (between 0.6 and 0.7) and are also BERT-based models. Our model has a higher recall but a significantly lower precision, suggesting that it flags for PCL excessively. We believe this is caused by the aggressive downsampling of non-PCL training data.

7 Conclusion

We develop a binary classification ensemble model for detecting patronising and condescending language in text, using a combination of RoBERTa and DistilBERT finetuned encoders. We find data augmentation and ensembling to be effective methods. We achieve an F1-score of 0.5731 on the official dev set, outperforming the RoBERTa base baseline by 9%. We also explore various pre-processing methods like reordering by text length and removing punctuation, but find insignificant levels of improvement.

We analyse the performance of the final model on the official dev set, and find that the model is better at identifying PCL where there is a lot of patronising content. We find that the model has varying performance across different data categories and sentence lengths. In particular, it performs best in the "in-need" category and worst in the "disabled" category.

Further work involves using larger batch sizes and exploring state-of-the-art model architectures other than RoBERTa and DistilBERT, as well as techniques like adversarial training and chain-of-thought question answering (Wei et al., 2022). It also involves investigating how our methods scale with a larger training corpus.

References

Jaouhar Fattahi and Mohamed Mejri. 2021. Spaml: a bimodal ensemble learning spam detector based on nlp techniques. In *2021 IEEE 5th international conference on cryptography, security and privacy (CSP)*, pages 107–112. IEEE.

Monisha Kanakaraj and Ram Mohana Reddy Guddeti. 2015. Nlp based sentiment analysis on twitter data using ensemble classifiers. In *2015 3Rd international conference on signal processing, communication and networking (ICSCN)*, pages 1–5. IEEE.

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. *arXiv preprint arXiv: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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

A Example Text

Text taken from the official dev set (Perez Almendros et al., 2020).

A.1 Example 1

"Krueger recently harnessed that creativity to self-publish a book featuring the poems, artwork, photography and short stories of 16 ill or disabled artists from around the world. She hopes the book, which contains some of her own work as well, will show how talented disabled people can be."

A.2 Example 2

"Last month, Dr Maszlee, in an interview with RTM, assured marginalised communities like the disabled that they would be given equal opportunities under the education system."

A.3 Example 3

"Recommended changes to residential consumer tariff schemes will be designed to give more relief to poor families, especially those living in compound houses. <h> National Builders Corps"

B Hyperparameter Tuning Results

lr	AdamW β_2	Warmup	F1-Score
1e-5	0.98	0.02	0.5323
1e-5	0.98	0.06	0.5341
1e-5	0.98	0.1	0.5407
1e-5	0.999	0.02	0.5255
1e-5	0.999	0.06	0.5444
1e-5	0.999	0.1	0.5333
1e-5	0.99	0.02	0.5484
1e-5	0.99	0.06	0.5311
1e-5	0.99	0.1	0.5387
1e-6	0.98	0.02	0.4665
1e-6	0.98	0.06	0.4746
1e-6	0.98	0.1	0.4727
1e-6	0.999	0.02	0.4913
1e-6	0.999	0.06	0.4906
1e-6	0.999	0.1	0.4963
1e-6	0.99	0.02	0.4726
1e-6	0.99	0.06	0.4746
1e-6	0.99	0.1	0.4732
5e-6	0.98	0.02	0.529
5e-6	0.98	0.06	0.5075
5e-6	0.98	0.1	0.5359
5e-6	0.999	0.02	0.5282
5e-6	0.999	0.06	0.5265
5e-6	0.999	0.1	0.5295
5e-6	0.99	0.02	0.5247
5e-6	0.99	0.06	0.5293
5e-6	0.99	0.1	0.5272

Table 4: Full hyperparameter tuning results on RoBERTa base: presenting F1-score, trained with different learning rates and scheduler settings, with 5 epochs. Learning rate refers to the AdamW ϵ value. Evaluated on the internal dev set.