NLP Coursework: SemEval-2022 Task 4 - Patronizing and Condescending Language Detection

Andrew Wu
Imperial College London
aw122@imperial.ac.uk

Zhaoyu Wang
Imperial College London
zw1222@imperial.ac.uk

Yanda Tao
Imperial College London
yt522@imperial.ac.uk

1 Introduction

We used the Don't Patronize Me! dataset and corresponding labels to develop a Roberta-based model that detects patronizing and condescending language (PCL). PCL can have negative effects in various contexts, damaging relationships and trust. We employed data preprocessing and fine-tuning techniques to improve the model's binary classification performance. Our model achieved a 0.61 F1 score on positive labels, surpassing the baseline RoBERTa model by 0.13 on the dev dataset.

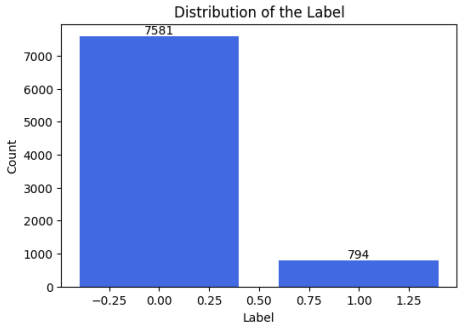

Figure 1: Label distribution in the PCL training set

2 Data Analysis

The Don't Patronize Me! dataset contains 10469 paragraphs from vulnerable social groups in total. Each paragraph has a unique id, a community keyword, a country code and a label of 0-4 determining whether the text contains patronizing language. If the label is $\{0,1\}$ then it is not PCL, else otherwise.

2.1 Summary of Train Dataset

The training dataset has 8375 data samples, with 794 positive labels (patronizing language), 7581 negative labels (non-patronizing language). The maximum number of words in a sample's text is 909, but the median number of words is 42. Only 6 sample texts have more than 256 words.

2.2 Analysis of Class Label

Figure 1 shows label imbalance issue in the dataset, where the number of negative labels is 10 times higher than positive labels. This can cause bias towards the majority class and poor model performance. Correlation was found between text length and the likelihood of PCL, with the highest rate in the 120-150 word range with a positive rate of 17.4%. However, since we have limited

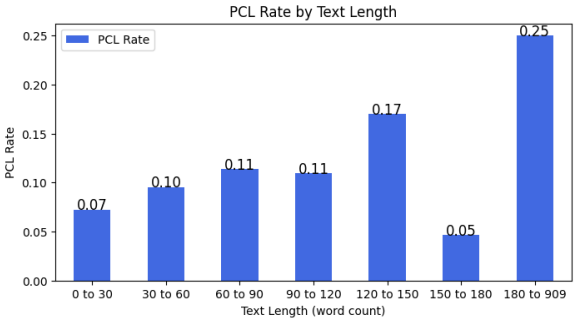

Figure 2: PCL Rate of Different Text Length Intervals

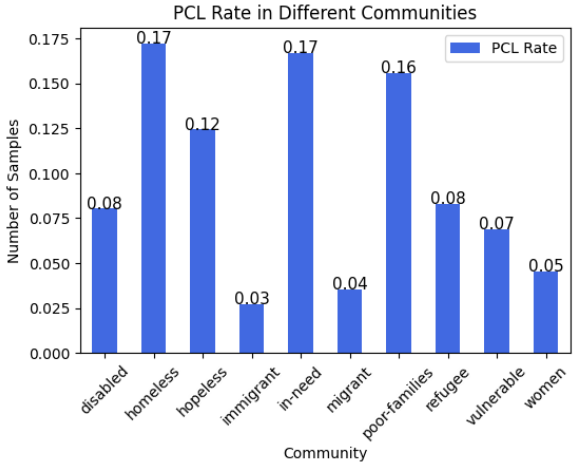

Figure 3: PCL Rate in Different Communities

samples for lengths exceeding 150 words, the relationship between longer sequences of text on PCL remains unknown. Furthermore, different community labels have significantly varying PCL likelihood, with "homeless" having the highest and "immigrant" having the least. This prompted the use of community labels as additional input for classification, along with text embeddings.

2.3 Qualitative Assessment

Label	Example
PCL	Housing Minister Grant Shapps added: ' The plight of homeless people should be on our minds all year round - not just at Christmas.
Not PCL	Donald Trump says he will sign an executive order to end migrant family separations.

Table 1: Examples of sentences with different labels

Detecting patronizing language is challenging as it's often concealed in superior or condescending attitudes, making it subjective and dependent on the receiver's sensitivity. Table 1 includes examples of detected PCL and no-PCL, where Example 1 is a type of PCL involving unbalanced power relationship. The example highlights the difficulty of distinguishing subtle differences in phrasing, where Housing Minister Grant Shapps was viewed as the PCL user who saw himself as the savior of the homeless. Example 2 is labeled as non-PCL, but it could also be viewed as Donald Trump seeing himself as the savior to migrant families. These nuances require a deep understanding of the hidden meaning of language and make it challenging for language models to capture these differences, especially with a small dataset size of 8375.

3 Method

3.1 Data Augmentation

To address the high class imbalance dataset we are given with, and prevent model from learning the pattern of predicting only negative labels, three data augmentation strategies were explored:

EasyDataAugmentation: (Wei and Zou, 2019) proposed a method that combines 4 strategies together into a pipeline to augment the data. The four strategies consists: synonym replacement, random insertion, random swap and random deletion.

Back Translation: (Kay and Roscheisen, 1993) suggested a method using a target language model to translate source language to another language and translating it back, such that wordings of the source language is changed while keeping the most contextual information.

Augmentation by Checklist Invariance Testing (CIT): A data augmentation method derived from CI-Test (Ribeiro et al., 2020) that involves: name replacement, location replacement, number alternation and contraction/extenstion.

	EDA	BackTranslation	CheckList
F1-score	0.52	0.56	0.61

Table 2: Effect on maximum PCL F1-score obtained using each data augmentation method, with RoBERTa-large + WRS + LLRD

Comparison Table 2 shows the effect of different data augmentation methods on the F1 score of the positive label, with Augmentation by CIT resulting in the greatest improvement. EDA's approach is considered too coarse and may result in loss of original PCL information, while Augmentation by CIT replaces names, locations, and numbers without affecting the text's meaning. Therefore, Augmentation by CIT is chosen as the data augmentation method.

3.2 Sampling

After augmentation, the positive-to-negative label ratio rises from 1:10 to 1:4, yet still being quite imbalanced. To further address the imbalanced class label issue, weighted random sampling (WRS) is implemented during the training process. WRS assigns each datapoint a weight, with datapoints with a larger weight being sampled more frequently. By assigning a higher weight to samples with a positive label, WRS further alleviates the imbalance problem.

3 different sets of weights are explored in our method:

$$w_i = \begin{cases} \frac{1}{c_p} & \text{if it contains PCL} \\ \frac{1}{c_n} & \text{otherwise} \end{cases} \quad (1)$$

$$w_i = \begin{cases} \frac{c_n}{c_p} & \text{if it contains PCL} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

$$w_i = \begin{cases} \frac{1}{\sqrt{r_p}} & \text{if it contains PCL} \\ \frac{1}{\sqrt{r_n}} & \text{otherwise} \end{cases} \quad (3)$$

where r_p and r_n denotes the ratio of samples containing PCL and not containing PCL respectively, r_p and r_n denotes the count of samples containing PCL and not containing PCL respectively.

Based on the empirical result that we have obtained, weights calculated in equation (3) is the most effective and has the most improvement on the F1 score of positive labels. Therefore, method 3 is selected as the method to compute the weight for each sample in WRS.

3.3 Layer-wise Learning Rate Decay

To improve results obtained after training, we apply layer-wise learning rate decay (LLRD) to the model. The idea behind is that layers at different depth have different functionalities, some may focus on general features and some on specific information (Yosinski et al., 2014). LLRD introduces different learning rates to different layers to find the best-ft learning strategies. It has already improved improved a variety neural-network-based tasks.

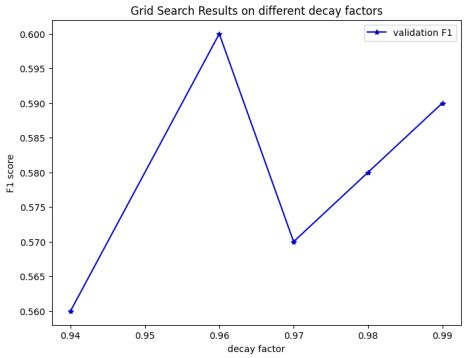

Figure 4: Investigation with different decay factor

Inspired by this approach, we assign a different learning rate to each of the layers appears in the model, including the ones from the pre-trained model's encoder, hidden layer and linear layer from the classifier, with a initial learning rate of 10^{-5} at the first layer. The initial learning rate propagates through the layers with a factor. To find out the optimal learning rate schedule, we perform a grid-search on learning rate propagation factors, 0.94, 0.96, 0.97, 0.98 and 0.99. We find that a decreasing factor of 0.96 results in the best performance. We could have done a thorough grid search over the learning rate of each layer (and not the propagation factor), but it would require a computation time too long for this particular task.

3.4 Model Choices

The objective of this task is to implement a transformer-based model that achieves a higher F1 score than the baseline RoBERTa-base model provided for the PCL classification task. As a starting point, we chose to use RoBERTa-base and explore other options from there. We first experimented with increasing the model's complexity by using RoBERTa-large, and then added various empirical methods to further improve performance.

During our exploration, we considered several different transformer-based models, including different versions of DeBERTa (He et al., 2021b) such as DeBERTaV3. DeBERTa is a transformer-based model that leverages techniques such as Disentangled Attention and Enhanced Masked Decoder to improve performance on a range of downstream tasks. DeBERTaV3 (He et al., 2021a) was used for comparison with RoBERTa-large due to some of the improvements it brings over its predecessors.

After conducting experiments, we observed that while DeBERTaV3 had a better baseline performance on the PCL classification task with an F1 score of 0.56-0.58 on PCL after training with an un-augmented dataset, it did not adapt well to the series of empirical methods we applied to improve performance. In contrast, RoBERTa-large had a lower baseline performance but was able to improve its F1 score after the application of the empirical methods we implemented. Based on these observations, we decided to use RoBERTa-large as the basis for our final implementation.

4 Results and Anlysis

4.1 Implementation Setup

The goal of this study is to perform a classification task on the annotated dataset "Don't Patronize Me!" of patronizing and condescending languages (PCL) provided by (Pérez-Almendros et al., 2020). All our trials and evaluations are performed with Nvidia P6000 GPU and pre-trained RoBERTa-large model as the base of our implementations. Moreover, hyperparameter tuning of all empirical methods is done by creating an internal dev set using the given training data, however, the results in this report are the results on the official dev set after the hyperparameter tuning is

done. Putting the empirical methods we have presented above together, we use the following hyperparameters and setup in our implementations.

We use checklist augmentation implemented by the package (TextAttack, v0.3.8, 2022) (see 3.1) to perform oversampling on the texts containing PCL. We augment the text column by altering 20% of each example's words, generating 4× as many augmentations as original inputs. This gives us a final train dataset of a more balanced composition, containing : 3149 sentences flagged PCL (instead of 794), and 7581 sentences without.

In terms of weighted random sampling, we use the equation (3) to calculate weights for samples of each label.

For layer-wise learning rate decay, we use a decay factor of 0.96.

During the training process, we define the following hyperparameters: we use AdamW optimizer, training over 10 epochs while including an early stopping strategy based on the monitored validation loss, with a batch size of 16. We also set the maximum sequence length of samples to be 256, after padding and truncation.

4.2 Result

Overview of Result

Table 3 summarizes the F1 scores of models from the official dev dataset with different improvements, and the final model with all improvements achieves an F1 score of 0.61. Moving from Roberta-base to Roberta-large improves the F1 score by 0.05, this is expected as Roberta_large has 355M parameters whereas Roberta_base has only 125M and larger pre-trained models are usually more capable of understanding subtle features hidden in the language. Data augmentation by CIT on Roberta-large increases the F1 score by 0.02, and when used with weight sampling, a further increase of 0.03 is observed, addressing the imbalance issue in the dataset. Lastly, adding LLRD scheduler with a decay factor of 0.96 found by grid search improves the model to 0.61 F1 score.

Comparison with Simple Baselines

Table 3 compares the F1 score of our final model to two baseline models that use the Bag of Words (BOW) method to represent text input as numerical embeddings, by computeding the frequency of each word in the training data. The second baseline method trains a Naïve Bayes Classifier using the numerical embeddings from BOW and their corresponding labels, by estimating the conditional probabilities of each word given the PCL label. Our final model significantly outperforms these simple baseline models.

"The Minister said a society 's measure of its humanity is how it treats its weakest and most vulnerable members ."

The simple baseline method misclassified the above sample of PCL from the external dev dataset that was correctly classified by our model. One possible reason for this performance difference is that the simple baseline method used BOW to represent the text input, which only counts the occurrence of each word in a document and represents it as a vector of word count. This method does not include fine-grained information such as word order and grammar, which are crucial in understanding the context and meaning behind a piece of text. As we previously noted, detecting PCL language is a difficult task that requires understanding of subtle meanings hidden behind words. Therefore, using BOW may not be sufficient in representing the detailed meaning of a text, which may lead to the failure of PCL classification.

5 Analysis Questions

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

From the training log and the test results of our baseline and designed model, it is easy to observe that there is a noticeable improvement in model performance, especially on predicting examples with a higher level of the patronising text. The comparison between our best model and the 2 simple baselines is illustrated by table 3, in terms of F1-score.

When predicting examples with a higher level of patronising content, the precision, recall and f1-score are 0.52, 0.71, 0.60 respectively. Compared with the the metrics (0.35, 0.38, 0.31) for BoW + Naive Bayse and (0.36, 0.36, 0.36) for BoW + Logistic Regression, the model's receives 96% increase in f1-score, over 40%increase in precision and over 66% increase in recall.

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

Table 4 we can see that there is no significant increasing or decreasing trend of how increasing the

	BoW + Logistic Regression	BoW + Naive Bayes	RoBERTa baseline	RoBERTa-large	RoBERTa-large + CIT	RoBERTa-large + CIT + WRS	RoBERTa-large + CIT + WRS + LLRD
F1-score	0.31	0.36	0.49	0.54	0.56	0.58	0.61

Table 3: Effect on maximum PCL F1-scores obtained with different empirical methods and pre-trained models

length of the input would affect the likelihood of misclassification of PCL by our model. It is worth noting that for a number of words that exceeded 150 in a text, the misclassification likelihood is equals to 100%. However, since there are only 2 available samples that are PCL and also exceed a word count of 150, the samples are too small in order to reflect the true performance of the model on this range of length of input.

5.3 To what extent does model performance depend on the data categories?

By analysing the performance we have obtained with our best model, we can notice some interesting results that are both meaningful for social studies and for text analysis. We looked at the data categories we estimate to be the most impactful on PCL content - community.

Table 5 shows the top 3 f1-score and the community that has the least f1-score on label "with PCL", together with other metrics. In this table is highlighted the significant difference of model performance between data categories. Our hypothesis is that some intrinsic linguistic features and frequent topics may be the cause of these differences.

Word count	Mis-classified count	total count	Mis-classify rate
(0,30]	10	37	0.27
(30,60]	27	95	0.28
(60,90]	15	47	0.31
(90,120]	3	13	0.23
(120,150]	0	5	0
(150,180]	2	2	1

Table 4: Misclassification Rate with different Text Input

community	f1 score	precision	recall
in-need	0.732	0.612	0.909
vulnerable	0.651	0.609	0.700
migrant	0.615	0.500	0.800
immigrant	0.444	1	0.286

Table 5: Prediction PCL f1-score vs. community

6 conclusion

Our proposed Roberta-based model, incorporating 4 improvement methods customized for detecting patronizing and condescending language, achieved an F1 score of 0.61 on positive labels, surpassing the baseline model by 0.13. We also highlighted the importance of addressing class label imbalances in the model-tuning process. For future experiments, we suggest exploring the use of the Conditional Transformer Language Model (CTRL) instead of Roberta. Using the vulnerable group label as a control code in the context may result in better embeddings, improving PCL classification performance.

References

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Martin Kay and Martin Roscheisen. 1993. Text-translation alignment. *Computational Linguistics*, 19(1):121–142.

Carla Pérez-Almendros, Luis Espinosa-Anke, and Steven Schockaert. 2020. Don't patronize me! an annotated dataset with patronizing and condescending language towards vulnerable communities.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accu-

racy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

TextAttack. v0.3.8, 2022. Qdata/textattack. https://github.com/QData/TextAttack.

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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?