SemEval-2022: Patronizing and Condescending Language Detection

Pietro Vitiello
Imperial College London
pv2017@ic.ac.uk

Ivan Ereshchenko
Imperial College London
ie918@ic.ac.uk

Alex Holderness
Imperial College London
arh121@ic.ac.uk

Abstract

Detecting patronising and condescending language in media is an under-researched field which promises to improve inclusivity in the online world. However, this task poses many technical challenges, a few of which we discuss in this paper.

1 Introduction

In this paper we explore the efficacy of transformer-based approaches to a binary classification task, predicting whether a given text contains patronizing or condescending language (PCL). Model performance is primarily evaluated on the F-1 scores over the positive class, as detailed in SemEval-2022 Task 4 [1], although this paper also considers other useful metrics for assessment. We aim to present a comparison of the different models tested for this task, as well as explore a variety of data augmentation and preprocessing techniques, which when compounded together, surpass the baseline F-1 score.

2 Background and Challenges

This work looks at an unvalued area of natural language research: PCL. Modelling and classifying language such as hate speech [2], unhealthy comments [3], and misinformation [4] are tasks that NLP has previously had success in. The domain of PCL, due to its subtle and often subjective nature [5], has not yet enjoyed this level of attention and so PCL detection presents itself as a potentially high-impact technical challenge, with many so-far unexplored applications.

2.1 The Task and the Data

The aim of this task is to build a system for identifying PCL within a given paragraph, extracted from online news articles. Each paragraph was selected due to it containing a reference to a particular vulnerable community. 10,636 of these paragraphs make up the data relevant to this project, with attributes pertaining to the keyword signifying a vulnerable community, the country of origin, the text, and a label which determines the number of annotators that viewed the text as containing PCL.

2.2 Challenges Presented

The dataset used for this project presents an imbalanced classification scenario; whilst perhaps being more representative of PCL occurrence in online articles, the imbalance of approximately 90.52% not PCL samples compared with the 9.48% PCL samples complicates the task. This overabundance of samples from the majority class can eclipse the model's skill in predicting the correct class label as it becomes more challenging to learn the underlying features of the minority class. To combat this we explored upsampling through data augmentation and using a weighted loss function that penalized incorrect predictions on the minority class twice as much.

2.3 Motivations

PCL in news media can have harmful effects such as dehumanizing minorities or encouraging discriminatory behaviour [6]. In this study we present a prototype model which could have applications in automated PCL detection in media.

3 Method

The dataset used for this project and its accompanying paper details two benchmark results for classifying PCL, the highest of which being a RoBERTa model achieving an F-1 score over the positive class of 0.48. Our investigation examines variations on this approach as well as the effect

Variation	Precision	Recall	F1-Score
Baseline	0.40	0.69	0.51
PP	0.47	0.70	0.56
PP (stop)	0.46	0.55	0.50
DA	0.53	0.56	0.55
DA + PP	0.55	0.59	0.57

Table 1: Evaluating over the DEV set positive class (PCL) before and after preprocessing (PP), preprocessing with stopword removal (PP stop) and data augmentation (DA).

of data preprocessing and data augmentation on model performance. This section also details our model selection and hyper-parameter tuning processes.

3.1 Preprocessing and Data Augmentation

This project implemented the following text preprocessing steps: remove punctuation and capitalizations, remove digits and stopwords, and lemmatize all text. We felt these steps would have a net positive effect of reducing complexity in our text without losing semantics. Initially stopwords were removed also, as this was thought it would give more focus to tokens that contribute to sentence semantics. This was reinforced by stopword mean statistics of all the training samples which revealed for an average sample, approximately 35% of that text was included in our stopwords list. This showed no improvement in performance; we considered whether this was due to our transformer's maximum input sequence length (80 tokens) and so reduced this number to account for the shorter spans without stopwords. However, performance on both the Dev and Test set was improved with the inclusion of stopwords.

As mentioned in Section 2.2, we used data augmentation to mitigate the effects of the class imbalance. This entailed English to German back-translation [7], followed by synonym replacement and noisy random word injection, to ensure dissimilarity in the generated sample versus the original, which can cause a model to overfit whilst training. This improved performance substantially, the results are shown in 1.

3.2 Model Selection

Within the field of NLP there is the habit of considering bigger models to be more performing. As a result we chose to experiment with the large variant of RoBERTa, which proved to be more performing than the baseline. However, the size of the latter model made it very computationally demanding and lead to slow training times. Consequently, we experimented with the ELECTRA model, which approached things differently, by aiming at a better performance than BERT, while making use of fewer parameters. This model managed to beat the baseline, but was far from the performance of the RoBERTa-large, hence we decided to focus on the latter.

3.3 Hyperparameter Tuning

We performed Bayesian optimization to tune our model's hyperparameters, with a scoring metric of minimizing validation loss which was calculated on the official Dev set. The results of this experiment are shown in Figure 1. The following were indicative of good performance: a model trained for less than 10 epochs, smaller learning rates in the realm of 1e-5, batch sizes 32 or 64, weight decay around 5e-9, and transformer maximum sequence lengths between 60 and 90. Models that were trained for longer, had larger learning rates, and smaller maximum sequence lengths

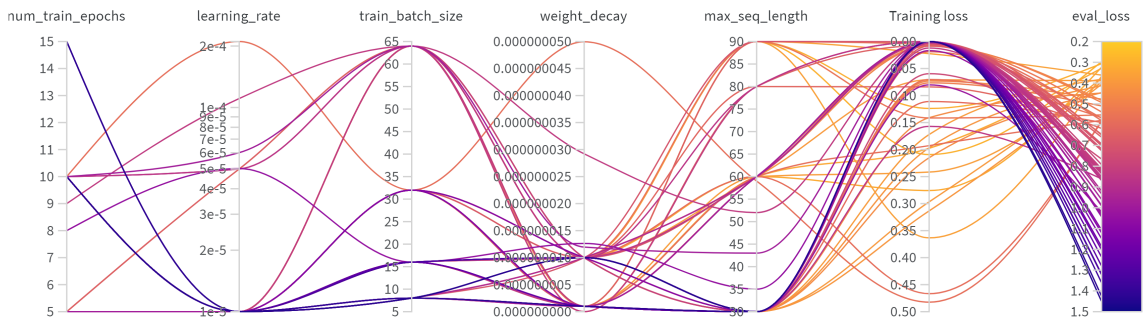


Figure 1: Hyperparameter tuning on final model.

Model	Precision	Recall	F1-Score
RoBERTa-L	0.58	0.56	0.57
ELECTRA-S	0.38	0.73	0.50

Table 2: Final model performance over the DEV set positive class (PCL).

were more prone to overfitting and would regularly achieve near 0 loss on the training data.

4 Experimental Results and Analysis

Our final RoBERTa-large model achieved the best score on both the Dev set and the official Test set (verified through CodaLab under username "arh121"). The results of evaluating our top performing models on the DEV set positive class are shown in Figure 2.

4.1 Predicting Instances with Higher Levels of Patronizing Content

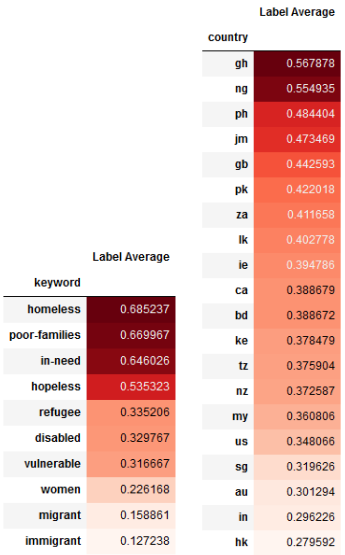
Our model made 169 incorrect predictions over both classes in the DEV set. The majority of these are contained in original labels of "3", which is one annotator assigned the label 2 and the other assigned the label 1, i.e. a split between borderline and certain PCL. Of samples with a label "4", only "25", which was the smallest of inaccuracies per label, were incorrectly predicted. Considering how the label "4" only makes up 4.39% of the labels in the Dev set, we can conclude our model does reasonably well at predicting instances with higher levels of PCL.

4.2 Effect of Input Sequence Length on Model Performance

As mentioned in Section 3.3, we found that models using a smaller input sequence length were more prone to overfitting. At an input length of 31 would mean only 25% of samples from the original training text would be included in full by the model. This would mean fewer samples potentially containing the part of text indicating PCL or not. The negative effect of small sequence lengths was worsened by our data augmentation which contributed to the overfitting as augmented samples would appear even more similar to their originals once the text was restricted to smaller lengths. We found sizes of 80 plus were not computationally expensive and still maintained sufficient focus on areas of the text contributing to semantics.

4.3 Can we Incorporate Categorical Data?

Presented below are representations of correlations between keyword, country and the associated labels. The label average values shown are the average values of the origin labels, which range between 0 and 4.



keyword	Label Average
homeless	0.685237
poor-families	0.669967
in-need	0.646026
hopeless	0.535323
refugee	0.335206
disabled	0.329767
vulnerable	0.316667
women	0.226168
migrant	0.158861
immigrant	0.127238

country	Label Average
gh	0.567878
ng	0.554935
ph	0.484404
jm	0.473469
gb	0.442593
pk	0.422018
za	0.411658
lk	0.402778
ie	0.394786
ca	0.388679
bd	0.388672
ke	0.378479
tz	0.375904
nz	0.372587
my	0.360806
us	0.348066
sg	0.319626
au	0.301294
in	0.296226
hk	0.279592

Figure 2: Visualisation of correlation between keywords and countries and assigned labels

While we did not consider these categorical data when designing the model, they could provide some baseline statistics which could potentially be used as another input into the model. The high correlation demonstrated by 2 shows that this type of approach could be useful in providing initial data segmentation.

5 Conclusion

In this project we have demonstrated that through appropriate preprocessing, data augmentation, and hyperparameter tuning that a model can be trained to surpass the baseline performance and reasonably classify instances of PCL. Future works might improve on this by experimenting with different pretrained transformer models, or alternatively finding other ways to mitigate the negative effects of the class imbalance. One interesting avenue to explore would have been to split sample texts if there were multiple occurrences of the PCL keyword within.

The GitHub repository containing code can be found: https://github.com/LordLean/SemEval-2022-Task-4

References

[1] Carla Perez-Almendros, Luis Espinosa-Anke, and Steven Schockaert. 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, 2020.

[2] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). pages 75–86, 01 2019.

[3] Ilan Price, Jordan Gifford-Moore, Jory Flemming, Saul Musker, Maayan Roichman, Guillaume Sylvain, Nithum Thain, Lucas Dixon, and Jeffrey Scott Sorensen. Six attributes of unhealthy conversation. *ArXiv*, abs/2010.07410, 2020.

[4] Chengcheng Shao, Giovanni Ciampaglia, Alessandro Flammini, and Filippo Menczer. Hoaxy: A platform for tracking online misinformation. *WWW '16 Companion: Proceedings of the 25th International Conference Companion on World Wide Web*, 03 2016.

[5] G. Wong, A. O. Derthick, E. J. David, A. Saw, and S. Okazaki. : A Review of Racial Microaggressions Research in Psychology. *Race Soc Probl*, 6(2):181–200, Jun 2014.

[6] Julia Mendelsohn, Yulia Tsvetkov, and Dan Jurafsky. A framework for the computational linguistic analysis of dehumanization. *CoRR*, abs/2003.03014, 2020.

[7] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics.