Fine-tuning transformer-based language models for the detection of patronizing and condescending language

Radu Burtea
CID: 02139239
Imperial College London
rb1321@ic.ac.uk

Guillaume Requena
CID: 02117168
Imperial College London
gr221@ic.ac.uk

Moritz Knolle
CID: 01973936
Imperial College London
mak221@ic.ac.uk

1 Introduction

In this work we will explore the application of neural language models for the detection of patronizing and condescending language (*PCL*). The use of hate speech and discriminating language is on the rise in online content and media[16] and thus its automatic detection is of great importance to society. Further, language models are often trained on data scraped from online sources and have been shown to encode such bias contained in the original training data [4]. Hence, there have been ongoing efforts in the natural language processing community to de-bias language models [3] and make advances towards the automated detection of offensive, discriminating and biased language [2] as well as more recent developments [15] focusing on more subtle subcategories such as *PCL*.

2 The "Don't Patroninze Me!"- Dataset

This paper will focus on task 1 of the *"Don't Patronize Me!"*-Dataset [15] (*DPMD*). *DPMD* is made up of $10,637$ paragraphs about potentially vulnerable groups and was annotated by three expert annotators on each paragraph's *PCL* content.

2.1 Qualitative assesment

The detection of *PCL* is challenging, even for expert annotators, as shown by a Kappa Inter-Annotator agreement score of only 41% [15] between the annotators. The labels provided in the dataset encode the agreement between the annotators on a scale from 0-4 (0: no patronizing content, annotators agree; 4: clearly patronizing content, annotators agree), see original paper for details [15]. We are now interested in learning a binary classifier to detect *PCL* (0: Control, 1: *PCL*). Two example sentences from *DPMD* to showcase the challenging nature of this task is shown below.

Text	Label
After Vatican controversy, McDonald's helps feed homeless in Rome.	Control
Selective kindness: In Europe, some refugees are more equal than others	*PCL*

Table 1: Two example sentences and their respective labels from *DPMD*.

The authors of DPMD, suggest to consider all paragraphs with label below 1 as non-*PCL*, and everything else as PCL. This strategy will later be improved upon as we explore a method to encode annotation confidence into our labels in later parts of this paper.

2.2 Train- validation and -test split

To allow for model selection and unbiased estimation of generalization error, the training dataset was split into train- and validation-set. We refer to the ***official development set*** from *DPMD* as the ***test-set*** from now on, as this is what we use to report performance with in this work. Note that, only the validation-set was used for model selection purposes. To corroborate the generalization performance of our model we also report performance on the ***CodaLab*** **test-set**. Below, a pie-chart visualisation of the class-label distribution of the different splits is shown.

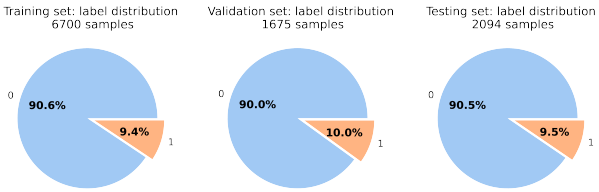

Figure 1: Pie chart illustration of class distribution in train- val and test- split of the dataset

2.3 Sequence length

The paragraph or sequence length in *DPMD* varies greatly, however the distribution has a very long tail and as a result we deem it sensible to crop sequences to a maximum length of 128 (shown in Fig. 2.a). This however will be re-evaluated in further experiments in following sections and was also included in a hyperparameter search. Additionally, the distribution of the frequency of input length is very similar for both patronizing and non patronizing content, both having median frequency values around inputs of length 40 (see Fig.2).

3 Methods

We will now outline the methodological changes made in our approach to *PCL* classification:

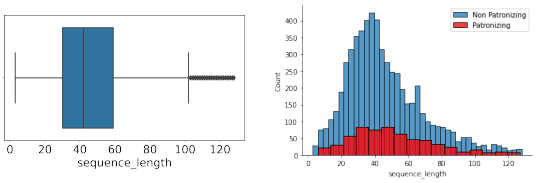

Figure 2: a) Box plot showing *DPMD* sequence length (cropped to max length 128) b) Distribution of input length frequency per label on the train-set

3.1 Fine-tuning pre-trained language models

Fine-tuning refers to the training of a neural networks, where the model is initialized with weights obtained from (pre-)training the network on another task. This can boost performance, as the network already has a good weight configuration and gradient-descent-based optimization has an easier time finding higher performing optima in the inherently very high-dimensional optimization landscape. We chose to fine-tune a range of different pre-trained, transformer-based masked language models/architectures [8, 12, 10] as well as one transformer pre-trained on a discriminative task [6]. To fine-tune these different backbone models we simply add a new head to the pre-trained architecture and train this new resulting model with binary cross entropy on *DPMD*.

3.2 Dealing with class imbalance

DPMD is highly imbalanced as shown in Fig. 1, with a positive (*PCL*) to negative class ratio of ∼1:9, which makes accurate *PCL* detection challenging. A common approach to help with class imbalance in classification probmlems is to weigh the impact of each class on the loss. In our binary case the binary cross entropy for two vectors y and $\hat{y}$ is given by:

$$b(y, \hat{y}) = -\left[w_n y \log \hat{y} + (1 - y) \log (1 - \hat{y})\right] \quad (1)$$

Whereby, w_n is commonly set to the empirical ratio of positive to negative examples in the training data [7]. We utilise the binary cross-entropy loss but chose to learn the optimal value for w_n through a hyper parameter search instead.

Another approach to solve disparate performance for minority classes in imbalanced statistical learning problems is the over- or under-sampling of the under- or over-represented class respectively. Both however come with downsides: under-sampling means that we are effectively throwing away data, while over-sampling can lead to overfitting. As a result more advanced distributional approaches have been developed [14, 5]. These however can still lead to overfitting with transformer models and as a result we turned to more sophisticated, natural language processing specific, data augmentation techniques, to over-sample the under-represented class.

3.3 Data Augmentation

Data augmentation plays a central part in obtaining high performance from deep neural networks. We thus consulted [9] and decided to apply back-translation [18] as well as contextual word embeddings [11] to over-sample the *PCL* samples. We notices a substantial increase in generalization performance as a result and utilized *NLPaug*[13] to implement these.

3.4 Hyperparameter tuning

We conducted an extensive hyper parameter search over a range of different hyperparameters with 800 trials using the *Tree-structured Parzen Estimator* from *optuna* [1]. The results of this search and the parameters explored as well as their impact on performance are visualised in Fig. 3. Optimal values found by the search were as follows:{*backbone*: BERT(cased), *learning*-rate(lr): 0.00006, *lr-warmup-steps*: 80, *batch-size*: 176,*max-seq-length*: 160, *bce-weight*: 4.09, *weight-decay*: 0.009, *epochs*: 3}

3.5 Incorporating annotation confidence

To overcome the limitation of the naive label binarization proposed by the original *DPMD* authors, we expand their approach to incorporate annotation confidence. The naive approach discussed previously does not encode label confidence that our binary classifier can meaningfully make use of during training. We on the other hand propose to simply modify the combined annotator labels that take values from 0-4 and divide them by 4. This means clear-cut control and *PCL* cases get a label of 0.0 and 1.0 respectively, while intermediary values get a value in between to encode the type of disagreement between the annotators. Notice that the case when both annotators agree on the borderline label we obtain a label of 0.5, which is at the border of our decision boundary, exactly the behaviour we would want from a classifier for *PCL* detection.

4 Results

Class	Dataset	Precision	Recall	F1-Score
Patronizing	Test-set	0.49	0.64	0.56
Control	Test-set	0.96	0.93	0.95
Patronizing	*CodaLab*	0.48	0.53	0.50

Table 2: Performance of our best model (BERT) on the test-set and *CodaLab* challenge test-set

5 Discussion & Analysis

5.1 Level of Patronizing Content

As shown in Fig. 5, the model performance is very good for highly patronizing content (label 4, 84%) as well as the control content (label 0, 90%). On the other hand performance is substantially worse for labels in between. For label 1 and This is most likely due to the fact that these labels in between are more ambiguous, harder examples as illustrated by the disagreement between the labelers or the fact that both annotators labeled the examples as borderline *PCL*. Thus, this this

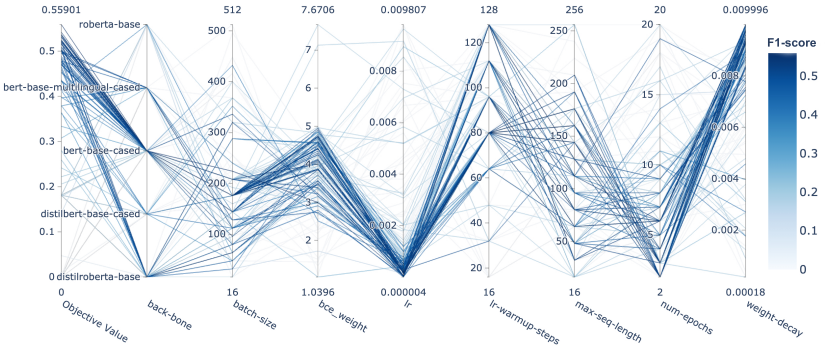

Figure 3: Coordinate plot showing the effect of different hyper parameters on model performance (validation-set F1-score computed for the positive class). Darker colors, mean higher resulting F1-scores.

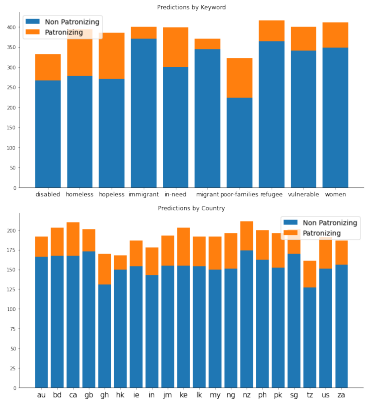

Figure 4: *CodaLab* predictions by Keyword (above) and Country (below)

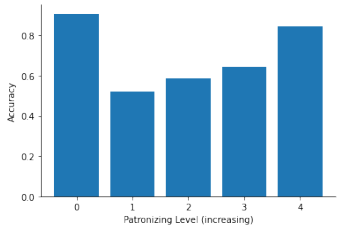

Figure 5: Bar chart illustrating relationship between test-set accuracy and original (added) annotator labels. Accuracy is highest for clear cut cases 0 and 4 as expected.

trend is to be expected and could potentially be alleviated by re-annotating the dataset with more annotators.

5.2 Sequence Length

0-20	20-40	40-60	60-80	80-100	100+
0.473	0.529	0.497	0.521	0.528	0.577

Table 3: Sequence Length (top) and resulting F1-Score (bottom) for the patronizing class (test-set)

Upon inspecting Table.3, we notice a roughly increasing trend in test-set F1-score (positive class) as the sequence length increases. This is likely due to the fact that with longer sequences the model gradually gains access to more and more context information to base its decisions on. The Kernel density estimation data (Fig. 6), obtained from our hyperparameter search however suggests that there might be a complex non-linear relationship between training batch-size and sequence length, that would require further experiments to fully understand.

5.3 Categorical Data

Analyzing Fig .4 we observe that the keywords *migrant* and *immigrant* are the least likely to be in sequences considered by our model as Patronizing, while keywords *homeless* and *hopeless* are the most likely. In terms of the level of Patronizing content as predicted by our model by country, most of them have very similar values, with slightly higher levels of patronizing content being registered in *Canada*, *Kenya* and *Nigeria* and significantly lower levels in *Hong Kong*.

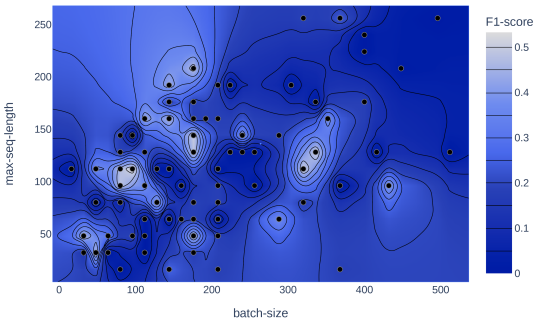

Figure 6: Kernel density estimation of different hyperameter search run F1-scores (val-set, positive class), showing the complex relationship between batch size and maximum sequence length

6 Further work

We have explored a range of different methods to improve the performance of *PCL* classification. However this work's scope was limited and thus further work could potentially explore the relationship between batch size and sequence length in more detail. Another interesting direction for research might be explicitly learning how to weight different training examples as first proposed in [17].

7 Conclusion

In conclusion, we have touched upon and explored a range of different methods for improving the performance of fine-tuned transformer-based language models for *PCL* detection. Further, we have shown that the changes outlined in the methods section, successfully transformed a model which initially predicted only zeros into a high-performance detector for *PCL* with real word potential impact.

8 Source Code Availability

Source code for this project is available here[1]

[1] https://gitlab.doc.ic.ac.uk/rb1321/nlp-coursework

References

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[2] V. Basile, C. Bosco, E. Fersini, N. Debora, V. Patti, F. M. R. Pardo, P. Rosso, M. Sanguinetti, et al. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics, 2019.

[3] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

[4] A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[6] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[7] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

[10] P. He, X. Liu, J. Gao, and W. Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[11] S. Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[13] E. Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019.

[14] A. Moreo, A. Esuli, and F. Sebastiani. Distributional random oversampling for imbalanced text classification. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 805–808, 2016.

[15] C. Pérez-Almendros, L. Espinosa-Anke, and S. Schockaert. Don't patronize me! an annotated dataset with patronizing and condescending language towards vulnerable communities. *arXiv preprint arXiv:2011.08320*, 2020.

[16] B. Perry and P. Olsson. Cyberhate: the globalization of hate. *Information & Communications Technology Law*, 18(2):185–199, 2009.

[17] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

[18] R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.