Semeval: Building a model for Don'tPatronizeMe!

Shashwat Kansal
sk4319@ic.ac.uk

Hyunhoi Koo
hk619@ic.ac.uk

Karan Obhrai
kjo19@ic.ac.uk

1 Introduction

The task paper (Pérez-Almendros et al., 2020) defines Patronising and Condescending Language (PCL) as messages that depict the vulnerable communities in a compassionate way whilst showing a subtle (sometimes unconscious) tone of superiority. Unlike openly offensive messages that contain discriminatory language, the intention is often to help the vulnerable by raising pity and awareness from the readers.

Detecting the use of PCL is difficult as it is often subtler and more subjective than other forms of more deliberate language. Prior work mainly focuses on detecting those explicit languages like offensive language and hate speech (Zampieri et al., 2019; Basile et al., 2019). Research that focuses on PCL (Wang and Potts, 2019) highlights the difficulty of the task and a need for a high-quality dataset annotated by experts.

The task paper proposes the *Don't patronize Me!* Dataset, which is a collection of 10,000+ paragraphs from the News on Web corpus (Davies, 2018), comprising of news articles published between 2010-2018 from 20 English-speaking countries. Each paragraph was filtered on 10 select keywords that potentially relate to vulnerable communities (e.g. 'disabled', 'immigrant', 'vulnerable'), ensuring a balance in the number of paragraphs among keywords and countries.

The dataset was annotated by 3 experts with backgrounds in communication, media, and data science to overcome the subjective nature of PCL. Each annotator had three labels per annotator: 0 indicates the paragraph contains no PCL, 1 indicates the borderline case, and 2 indicates that it clearly contains PCL. The labels were then aggregated into a 5-point scale, where total disagreements i.e. 0 vs 2 were resolved by the third annotator. Points 0 and 1 are treated as negative examples, and points 2,3 and 4 are treated as positive.

This paper focuses on the first task proposed on the task paper, which is a paragraph-level binary classification PCL.

Although the binary classification task does not require it, it is still important to recognise a more fine-grained categories of PCL, as it is useful for recognising and improving upon the strengths and limitations of the model. The task paper created 7 categories of PCL, which can be found in the original task paper.

2 Analysis of the Training Data

2.1 Quantitative Analysis

We performed basic analysis on the training data, with our focus around the correlation of features with the class labels (0-4). We look at many features but we will be focusing on the major correlations we have found, that may cause issues or influence our model decision.

Figure 1 shows the frequency of each point in our training dataset. Interestingly, the frequency of point 0 is significantly higher than any of the other class labels, with almost 7,000 values out of 8,375 total rows in the training data. This means we have to be careful to not weight our model towards this point, meaning we would weight the model towards pieces of text that do not contain

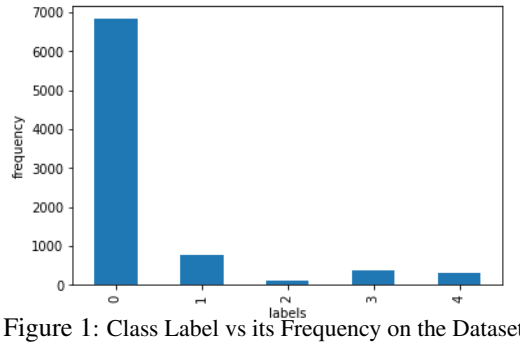

Figure 1: Class Label vs its Frequency on the Dataset

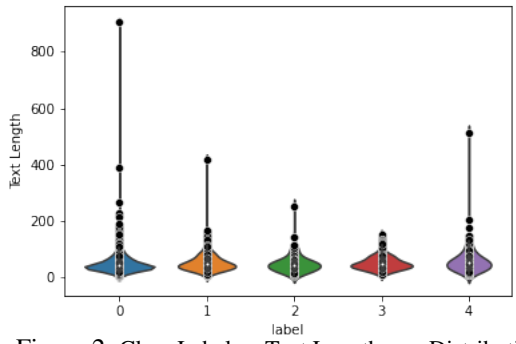

Figure 2: Class Label vs Text Length as a Distribution

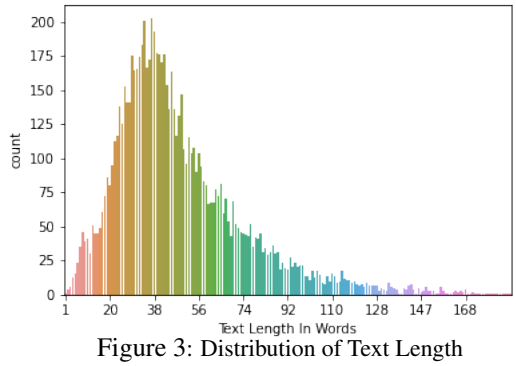

Figure 3: Distribution of Text Length

PCL, based on the label meanings from the task paper.

Figure 2 shows us the text length distribution within each point. Interestingly, we see point 0 has the biggest distribution, which as we know from 1 may be due to the high frequency of point 0 samples as well as a few outliers skewing the distribution, as shown by the frequency of long texts as per 3. This that larger pieces of text are less likely to be annotated as containing PCL, however the sample size of larger text isn't big enough to be significant. Interestingly however, point 4, which is designated as both annotators saying it clearly contains PCL, has the second highest distribution of text length, however this appears to be caused due to a couple of outliers, with the main distribution being in a tight range.

2.2 Qualitative Analysis

PCL language is subtle and subjective, which makes it harder to detect, a fact when accounted for creating the dataset through aggregating the decision of the three annotators. However even the expert annotators may pose some personal biases that may affect the classification results.

News articles (NoW Dataset (Davies, 2018)) were chosen as the datset for investigating the prevalence of PCL in media. It is unclear whether this dataset draws news from outlets of all political spectrums. As such, the decision to use this dataset may introduce some bias. However, the dataset might naturally contain text from a diverse set simply to increase the number of samples.

The dataset only contains English text, which may increase the prevalence of biases for certain non-English speaking fringe groups, such as refugees and migrants. However, we have not found any significant pattern towards those keywords. We also recognise that it is a practical limitation, since creating multilingual models would be significantly harder. This is further offset by taking articles from 20 countries, including English-speaking third-world countries to provide a more diverse set of viewpoints and cultures.

Overall, we believe the quality of the dataset is good as from our manual analysis, we find it classifies PCL language correctly. For example, the phrase *"[...] 'The plight of homeless people should be on our minds all year round - not just at Christmas.' "* is point 4 and obviously contains PCL, and the phrase *"[...] the focus of immigration actions would be illegal immigrants 'who have also otherwise violated our laws' "* clearly contains no PCL and is point 0.

However, some examples, especially in points 1-3, can become a lot less clear-cut. For example, paragraph 6947 reads: *"[...] we try to use delivery flights of new aircraft to our customers to ship medical or humanitarian donations to countries or regions in need"*. One could argue that this is an example of Shallow Solution PCL due to the vague nature of the solution presented. However, it could also be argued as just a PR strategy rather than actually presenting a solution. As such, judging the quality of the solution is complicated and will remain subjective.

3 The Model

Our model is a fine-tuned version of DistilBERT (Sanh et al., 2019), a 'distilled' version of the *BERT-base-uncased* model, with 40% less parameters and runs 60% faster, while preserving over 95% of BERT's performance in the GLUE language benchmark. The main reason why we chose DistilBERT is because of its relative training and inference speed being faster compared to other BERT-based models retaining most of its performance. The task paper (Pérez-Almendros et al., 2020) also mentions how models with too

many parameters tend to overfit without more data points, which implies that smaller models are able to generalise better. Our model is uncased as the original model is uncased and the text is insensitive to capitals letters.

The best model involved no text augmentation, instead used tokenization with padding and text truncation to 512 characters inputs. We perform batched tokenization for faster vectorized operations and reduced RAM usage. We also dropped any rows missing text values since transformers do not extract patterns well with empty strings. We then shuffle the training dataset in order to improve generalization of the transformer model to mitigate learning patterns on the data order. It also leads to more stable learning as well as data variability. We trained overall for 10 epochs.

We performed hyperparameter search using Ray Tune (more in §3.1). We trained the model for 10 epochs with a batch size of 32 for both train and validation set. We used an AdamW optimizer during learning with a 500 step warmup period before weight decay starts. We set the learning rate to `1e-5` with linear weight decay to value of 0. We found this learning schedule to be more effective compared to no learning schedule. We don't use early-stopping during training, but did in §3.1 using ASHAScheduler. We train the model using text column only for multi-label classification $0, 1, 2, 3, 4$, and then post-process/normalize the classes to $0, 1$.

We then evaluated the model trained on the internal training set on the official dev dataset, then retrained the model on both training and official dev set for inference on the test dataset. The model used for predicting the official dev set produced an F1 score of 0.52. Full results for this inference is provided on the repository.

3.1 Hyperparameter Tuning

For hyperparameter tuning, we decided to use Ray Tune to perform a hyperparameter search. We used a uniform log range between 1e-6 and 1e-2 for the learning rate, as well as a fixed-set of batch sizes (16, 32, 64, 128) for both training and evaluation. We found that 32 is the best batch size and 1e-5 for learning rate to start with. We used a FIFO Scheduler.

We then performed the hyperparameter search using HyperOptSearch (Bergstra et al., 2013) with an ASHAScheduler (Asynchronous Successive Halving Algorithm) (Li et al., 2018).

HyperOptSearch with TPE algorithm efficiently explores hyperparameter space by balancing exploration-exploitation trade-off. It prioritizes exploration of hyperparameters likely to yield better performance, while exploiting regions that have already shown promising results. TPE iteratively selects hyperparameters based on a probabilistic model, and converges to a set of high-performing hyperparameters for validation set.

The ASHA scheduler efficiently explores large hyperparameter spaces and can find good hyperparameters in fewer iterations than random or grid search. It trains candidate models for a fixed number of epochs and selects the best performers based on the objective metric, discarding the rest. The process is repeated until only one model remains, resulting in optimal parameters. This approach also incorporates early stopping to speed up the search process by discarding unpromising trials. We used this method to find the optimal hyperparameters for our best model.

3.2 Further Improvements

We improved our text classification model by using two effective data augmentation techniques. The first technique involves randomly masking words and filling them in using the Roberta base model, which expands dataset diversity and improves generalization ability. The second technique is back translation using t5-base and a reverse model to translate the text to French and back to English, introducing variations in word choice and structure while preserving semantic meaning. These techniques enhance dataset variability, resulting in a more robust and generalized model.

Other improvements we made were tweaking the hyperparameters, as well as sampling method. We tried to sample equal amounts of each class, which led to underfitting the data as there was a limiting factor of 192 rows for class 2 only, out of a total training size of over 8000. We also tried data replacement, so generating almost 500 rows per class of training data for balancing the dataset. This led to a slightly improved model, but still not sufficiently close to the best model we derived using the entire dataset. This may be again due to lower data size, as well as unable to learn intricate word patterns across the dataset.

With the data augmentation techniques, we did

not receive better F1 scores as much as we did by focusing more on the hyperparameter search, which is more effective than the data augmentation techniques.

3.3 Comparison with Baseline

The first baseline model used was a SVM Classifier model with a TF-IDF weighted BoW word embedding. The hyperparameters used are the same as ones given in the original task paper and no further hyperparameter tuning was performed. After training on the training dataset, the F1-score on the positive class was 0.102. Looking at precision and recall values (0.667 and 0.055) it is evident the model classifies most positive PCL texts as negative, which is to be expected as there is a heavy imbalance towards positive cases. This is exacerbated with the second baseline model, which uses GloVe embedding (Pennington et al., 2014) instead. Both SVM and CNN were used for the classifier, and both models produced an F1 score of 0.0, as it classified every text as negative.

This results make sense, as detecting PCL requires a macro overview of the paragraph, including some forms of world knowledge, rather than detecting the use of specific words or phrases. This explains why the TF-IDF model performed better than the GloVe model, because Bag-of-Words with TF-IDF weighting takes account of the entire document rather than just each words or sentences, which is the case with models that use the GloVe embeddings directly.

For instance, the TF-IDF model classified paragraph `1358` as not PCL, when it actually is classified as PCL with label of 3 with the category presupposition. The paragraph is as follows: *"[...] this could be devastating for the disabled and elderly, who lack the agility to cope with anything sudden and unexpected"*. Humans can immediately figure out that this is a clear example of Presupposition, as it shows an inaccurate assumption that every single elderly and the disabled. However, the model classified this as not PCL, which is potentially because there are no words or phrases that clearly indicate simpler forms of PCL like Unbalanced Power Relations and Compassion, which means that the model fails to detect PCL altogether as it performs poorly on more complex forms of PCL like Presupposition.

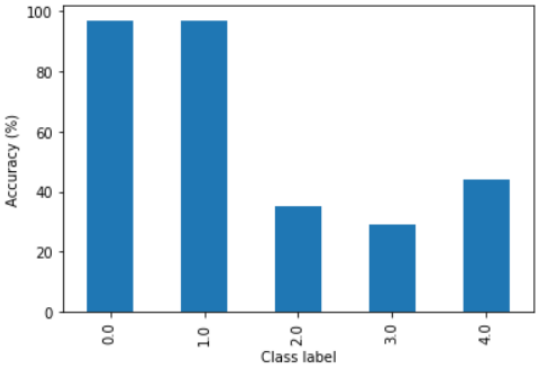

Figure 4: Accuracy of model vs class label

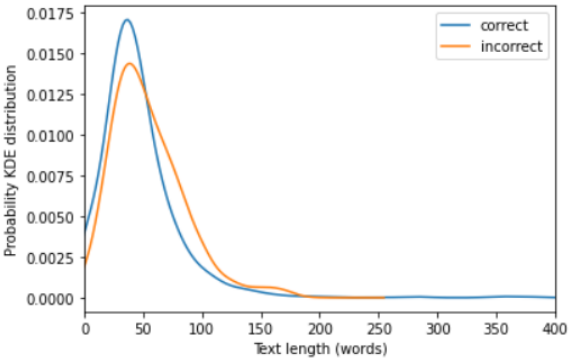

Figure 5: KDE Gaussian Of Correct Labelling Text vs Length In Words

4 Analysis

1. To what extent is the model better at predicting examples with a higher level of patronising content? Justify your answer.

As we can see from 4, we can see our model is good at identifying texts that do not contain PCL, with classes 0 and 1 having high accuracy. Accuracy drops signficiantly for classes 2 and 3,as expected, because for these classes, the PCL is disputed between annotators. We see our model rise to just above 50% accuracy of correctly labelling pieces of texts containing PCL (class 4). This means our model is prone to false negatives when disputed (class 2 and 3), but is okay at predicting examples with high (class 4) and low / no patronising content (class 1 and 0).

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

5 shows us the probability estimate for text being correctly labelled based on it's length, and as we can see, from the data, it appears that our model is likely correctly predicting texts of up to 75 words approximately in length correctly, however interestingly, we see that texts from 75 words onwards are labelled incorrectly with higher or equal probability to being labelled correctly in our

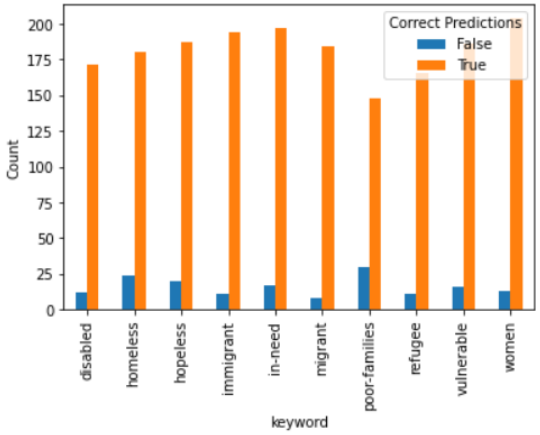

Figure 6: Correct and incorrect labelled of texts vs keyword

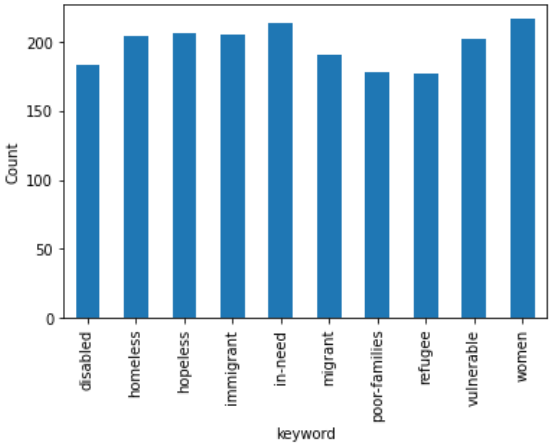

Figure 7: A graph the frequency of total texts per keyword

model. If we look at 3 we can see that the frequency of texts at this length are very small as a proportion and as such, it may be our model does not have sufficient pieces of longer texts to train on. Using 2, we can see classes 2 and 3 have a high density of samples in this range, so this is consistent from the above analysis that our model is worse at predicting texts with disputed PCL levels and as such, we can suggest that text length isn't actually a factor in our model, as this change can be attributed to the frequency of class 2 and 3 texts in that length range.

3. To what extent does model performance depend on data categories? E.g. Observations for homeless vs poor-families, etc.

Model performance appears to be especially good at analysing pieces of text related to 'migrants', 'refugees' and 'women' based on 6, and less good at analysing for PCL in texts related to 'poor-families' and 'homeless'. However, whilst there is a noticeable performance difference between these categories, they are not drastic, as we can see the ratio of false predictions to true predictions in each category are near consistent, with the worst ratio of correct predictions being on poor families, with a lower true prediction count but a highest false prediction count. This however, is still not huge in difference, as the shape of the graph in 6 is very similar to that of 7 with the general frequency of each keyword. Therefore, whilst there appears to minor correlation, it is not significant in our model, perhaps due to us not taking into account the categories column in our final model.

5 Conclusion

Based on our analysis above, we can conclude our model is good labelling pieces of texts with more extreme amounts of PCL at both ends of the spectrum, as we've seen correlations with class labels and the accuracy of our model label. However, whilst our model appears to have correlations between accuracy and text length, this is unlikely to be causal and likely due to our dataset distribution. There is small correlation, perhaps due to causation with keywords and the accuracy of our model. In order to explore this further, we could perhaps experiment to see if there is an affect, in our model of the different category labels and the accuracy of our labelling, with category labelling being the seven types of PCL, which has a many to many relationship of keywords to these categories. This might provide specific insights to the tone of the text and we could potentially feed into a new model.

5.1 Future Work

Further improvements for the model can be made, by significantly increasing the size of the input embeddings, as well as data augmentation techniques such as the masking rate. This all requires a significantly larger GPU RAM which is a limitation of our approach and abilities.

Data can be augmented from a wider range of sources that includes more countries and wider range of identities so the model can learn a more robust pattern. This is especially because the data is unbalanced class-wise as well as country-wise. Furthermore, a sentence-level text inference can be created and annotated so that large pieces of text in each example do not create unnecessary noise.

There are further methods of exploration, such as a one-vs-all approach which can be experimented to see if the binary classification for each class works better, due to the class imbalance.

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 54–63.

James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR.

Mark Davies. 2018. Corpus of news on the web (now): 3+ billion words from 20 countries, updated every day.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. 2018. Massively parallel hyperparameter tuning. *arXiv preprint arXiv:1810.05934*, 5.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation.

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. *arXiv preprint arXiv:2011.08320*.

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

Zijian Wang and Christopher Potts. 2019. Talkdown: A corpus for condescension detection in context. *arXiv preprint arXiv:1909.11272*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.