Natural Language Processing - Detecting Patronising and Condescending Language with Transformer Models

Alexander Mikheev
am7522@ic.ac.uk

Rahil Pandya
rrp18@ic.ac.uk
Acer Blake
ab222@ic.ac.uk

1 Introduction

The following report details the application of transformer-based models to the natural language processing task of detecting patronising and condescending language (PCL). In particular, we demonstrate that transformer models outperfrom common natural language processing baselines on the 'Don't Patronize Me!' dataset produced by Almendros et al (1), namely multinomial naive Bayes, and support vector machines (SVMs).

2 Data Analysis of the Training Data

The 'Don't Patronise Me' dataset, produced by Almendros et al (1), formed the base of our experimentation. The data set consists of 10,469 examples of potentially condescending and patronising language, each separately ranked on a scale between 0 and 2 by two human annotators, for a possible range of scores in the interval [0, 4]. These, however, we map to a binary classification task with possible labels 0 (non-PCL) and 1 (PCL). The most salient observation regarding the underlying data distribution is that it is significantly imbalanced, with 90.49% of the data being classified as non-PCL and only 9.51% of the examples being classified as PCL. This is immediately suggestive of a need for class balancing, which is an approach explored in Section 2.

The countries and vulnerable groups, or 'communities,' associated with each passage of text are approximately evenly distributed by their counts. However, with respect to their relative likelihood of being the target of PCL, there are three groups that stand out above others, namely the homeless, in-need and poor families, as seen in Figure 1. This was a key finding and motivated the hypothesis that the community labels can be used in conjunction with potentially PCL text to establish a more accurate classification.

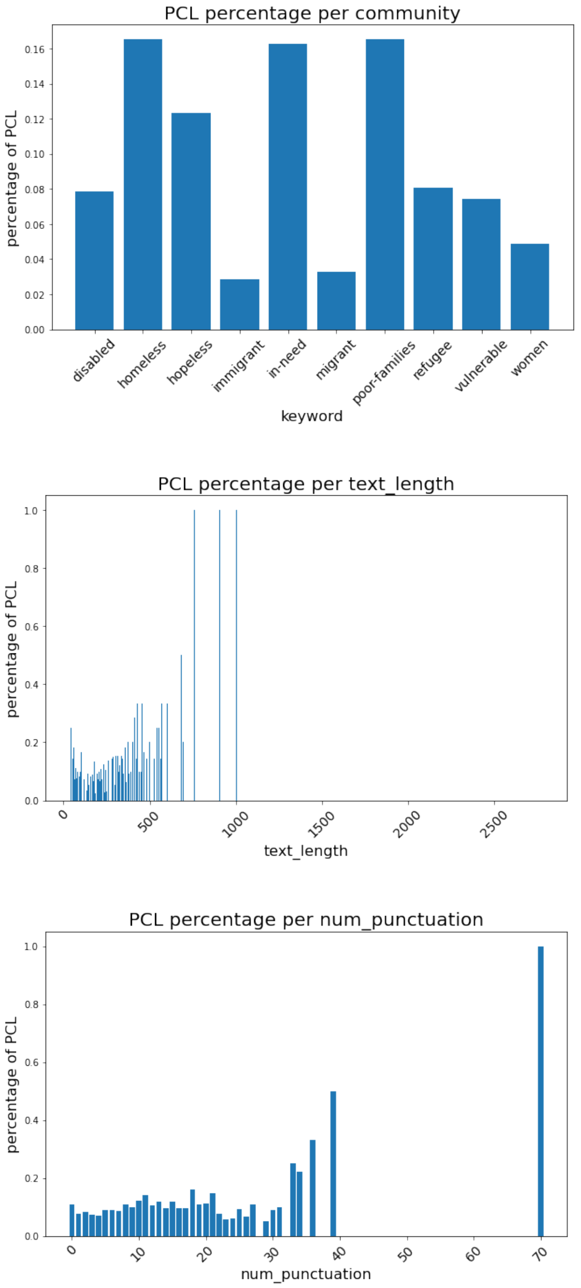

Figure 1: Plots for PCL as a function of various features of the dataset text: (a) PCL percentage by community label, (b) PCL percentage per text length in characters, and (c) PCL percentage by amount of punctuation.

Model	**F1**
Bert-base-cased	0.54269
Bert-base-uncased	0.50264
Albert-v2-base	0.5333
Distilbert-base-uncased	0.47128
Distilbert-base-cased	0.53828
Bart	0.52000
DeBerta-base	**0.55245**

Table 1: Final F1-scores of various Hugging-Face models on the internal development set. These models were tuned independently.

Hyperparameter	**Setting**
Learning rate	10^{-5}
Weight decay factor	10^{-2}
Optimiser	AdamW
Scheduler	None
Batch size	8
Upsample factor	9
Epochs	1

Table 2: A table of hyperparameters and design choices for the transformer architecture.

A sample of plots from the data analysis are visualised in Figure 1. From (b) we can see that the there is a broadly positive correlation between text length in characters and the amount of PCL present. This is an intuitive finding, as verbose prose is a common characteristic of PCL, namely belaboured explanations (1). Further, from (c), we can see that there is again a positive correlation between the amount of punctuation used and the amount of PCL. One interpretation of this result is that excessive punctuation can be used as a means of conveying sarcasm – a common form of PCL.

The subjectivity and difficulty of the task is evidenced by the fact that a number of labels generated by the annotators run counter to the assessments of the authors. For example, in dataset item 1808 the sentence ‘He wants more done now to help those in need’ was labelled as PCL. The authors do not agree with this labelling, and many more such examples exist. Further, there are also instances of language which the authors do regard as condescending that was labelled otherwise by the annotators. One such occurs in dataset item 318, consisting of the sentence ‘The Filipino immigrant’, which was labelled as not condescending by the annotators, but is judged to be so by the opinions of the authors.

3 Modelling

The final model is an adaptation of the Hugging Face ‘DeBerta base’ pre-trained architecture. Being a 100 million parameter cased model with 12 attention heads over 6 layers, each with a hidden layer size of 768, the model was deemed to be a reasonable compromise with respect to considerations of complexity, size, and training time. With the final parameters and meta-level design choices for the model given in Table 1, as well as the improvements outlined in the following sections, DeBerta achieves a final F1-score of 0.55 on the internal development set, and an F1-score of 0.58 on the official development set.

Regarding hyperparameter settings, the Hugging Face documentation provides information regarding the optimal hyperparameters for the task of training the DeBerta base architecture as a language model (2). While the nature of the present task is not language modelling, these suggested settings provided an informative starting point from which to conduct the hyperparameter tuning process. In particular, we conducted experiments by fixing each of the final parameters above, apart from the one being tested. We ran each experiment for 5 epochs and recorded the peak F1, the results for which are given in Table 3. The empirical data justifies the setting for the learning rate, batch size, optimiser, upsample factor of under balanced class, number of epochs and omission of learning rate scheduling, as summarised in Table 2. In cases where an experiment lead to rapid overfitting we would perform manual early-stopping to reduce development time.

A cased model was chosen as the authors’ experiments revealed cased models to generally outperform uncased ones, highlighted by the results of Table 1. We attribute this to the fact that while allowing for multiple cases increases the complexity of building a language model, such models can capture the nuances of language in a way that case insensitive models cannot. This becomes particularly important when dealing with difficult classification tasks, such as detecting PCL.

In total, four improvements were trialled to the model: upsampling the minority class, data augmentation through backtranslation, rephrasing through GPT-3, and the inclusion of categorical labels from the community column to the model.

Hyperparameter	Experiments	Results (F1-scores)
Learning rate	$10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$	0.536, 0.552, 0,475,0.392
Scheduler	Linear, cosine, polynomial, none	0.536, 0.532, 0.533, 0.552
Batch size	4, 8, 16, 32	0.514, 0.552, 0.541, memory error
Optimiser	Adam, Adamax, AdamW	0.528, 0.532, 0.552
Upsample factor	1, 4, 9	0.512, 0.530, 0.552

Table 3: Hyperparameter tuning experiments for the final DeBerta model.

As noted in Section 1, upsampling the minority class was suggested by the significant imbalance between the positive and negative classes. Upsampling the positive class such that the ratio between the two classes is approximately 50:50 proved to be effective in raising the F1-score to be competitive with the RoBERTa baseline of 0.48. While upsampling is sufficient to address issues with class imbalances, simply repeating data for the minority class is less desirable than creating new, diverse, and representative data through augmentation techniques. This motivated the development of a backtranslation pipeline.

Backtranslation is a technique in which new text is generated from a dataset by translating the source text to a different language and then back again. Since the translations are not one-to-one mappings, this introduces diversity into the backtranslated text, ideally producing new sentences which retain the semantic meaning of the original text. The motivation here is that this would address the problem of class imbalances while also allowing the model to learn robust representations of condescending and patronising language. Backtranslation was implemented using pre-trained Hugging Face translation models. Research revealed that of the more common European languages for which such translation models exist, German is the most appropriate language, being closest to English. Backtranslation proved unsuccessful as a data augmentation technique due to the fact that in the best case the backtranslated texts were identical to the originals, and in the worst case several words had been mistranslated and the text was rendered lower quality or outright unintelligible. The disconnect between the authors' expectations and the reality of backtranslation arises from the fact that the Hugging Face models used do not implement rephrasing. Rather, they attempt a one-to-one mapping between the source and target text, favouring structural similarity over fluidity and retention of sentiment.

With backtranslation experiments yielding poor results, rephrasing the source text directly through a publicly available language model was the next logical extension. For this the authors leveraged OpenAI's GPT-3 API (3) and prompted the model to rephrase each of the 637 positive class examples 9 times to bring the dataset balance to a more reasonable split of an approximately 50:50 ratio. An example of the nature of the rephrasings is given in Figure 2. Note that the version generated by GPT is gramatically correct and free of erroneous punctuation, whereas the original is not. The authors found this to be a common occurrence. Counterintuitively, however, while the quality of the GPT-generated rephrasings was very high, the use of the model's output did not improve the performance of the transformer. In fact, regular upsampling without any data augmentation proved to perform better.

The fourth and final improvement was to incorporate the categorical data in the community column when training the transformer. Per Figure 1 (a) we can observe that certain groups are significantly more likely to be the subject of PCL, and as such the authors suspected that passing this information to the model when training should improve predictive accuracy. A natural method of achieving this was to prepend the community label to the front of the text strings for each example and allowing the transformer to learn that the first token functions as a separate indicator. It was found that performance with this change was identical or marginally worse, and was not implemented in the final model.

Multinomial naive Bayes and a support vector machine (SVM) were used as baselines against which to compare the transformer model. Naive Bayes and SVMs are both standard baseline models for a variety of NLP tasks, and are used in the original 'Don't Patronise Me' paper, making these models a natural choice. Regarding performance, multinomial naive Bayes was able to achieve

Original: ”””” I feel it is the duty of us as humans to be compassionate to others in need and not treat them as vermin , ”” story-sharing website Upworthy quoted Furzer as saying .”

Augmented: Upworthy, a story-sharing website, quoted Furzer as saying, ”It is our responsibility as humans to be kind and understanding to those in need, not to regard them as inferior.”

Figure 2: A rephrasing generated by the GPT-3 API.

an F1-score of 0.37, while the SVM was able to achieve an F1-score of 0.31. Pre-processing for the baseline models entailed removing non-alphabetic characters, converting words to lowercase, removing stop words, stemming, and finally tokenisation. The need for such extensive pre-processing for the baseline models was motivated by the fact that they do not come pre-trained for language processing tasks, nor do they provide their own sophisticated word-embeddings and tokenizers. As such, computing a bag-of-words data structure is required to provide a numerical representation of the training set text to the models. This requires that the corpus be pre-processed to handle case sensitivity, words with common prefixes, and punctuation. This makes the vocabulary size of the bag-of-words smaller by stripping away redundancies and enriches the feature space.

We can glean insight into why the baseline models are significantly less powerful than transformer models at classifying PCL by consulting both theory and experiments. Theory informs us that purely statistical models, a category to which multinomial naive Bayes and SVMs belong, take no note of the structure of a sentence. That is, a bag-of-words representation of a sentence is invariant to the order of words, whereas PCL is highly dependent on context, which in turn is a function of sentence structure and word ordering. This is in contrast to the attention mechanism underlying transformers, as well as their use of contextual word embeddings (5). Further, simple pre-processing techniques, such as lower casing words, can lead to significant loss of information. An example of this can be seen in Figure 3, which shows the acronym ‘US’ being reduced to ‘us’, losing critical information about the subject of the sentence, potentially contributing to a misclassification.

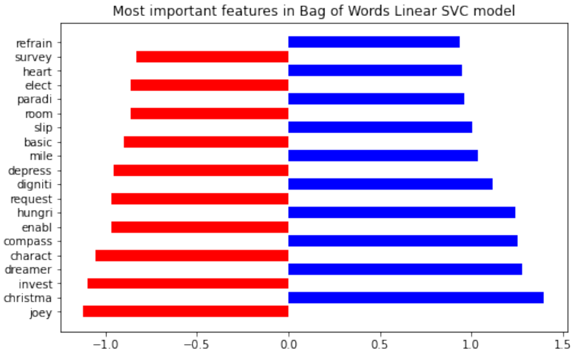

Figure 4: Relative feature importance for a subset of the dataset vocabulary for the SVM baseline.

Original: Many refugees do n’t want to be resettled anywhere , let alone in the US .

Stemmed: mani refuge n want resettl anywher let alon us

Prediction: Non-PCL

Actual: PCL

Figure 3: An example of the effects of pre-processing for the baseline model on the dataset text.

In computing and visualising the most important stemmed words for the bag of words SVM model in Figure 4, we can see that these can be hard to interpret. Being stemmed, it is difficult to tell which words corresponding to the stemmed features contribute most to the bag-of-words representation. We can observe, however, that stems corresponding to injunctions (‘refrain’) and emotionally charged words (‘depress’) appear among the list of important features, which is intuitive, as these are indicative of PCL.

4 Analysis

To analyse the performance of the model we consider three aspects of the transformer’s predictions: the extent to which the network is better at predicting examples with a higher level of PCL, how the length of the input sequence impacts the model performance, and finally the extent to which the classifier’s performance depends on the data categories.

Observing Figure 7, we see that the model is significantly better at detecting PCL from examples with a total lower scores for PCL, namely scores in the set $\{0, 1\}$. This is intuitive, as examples with PCL scores in this set constitute 90.49% of the original training data, providing the model with more examples to learn from than the 9.51%

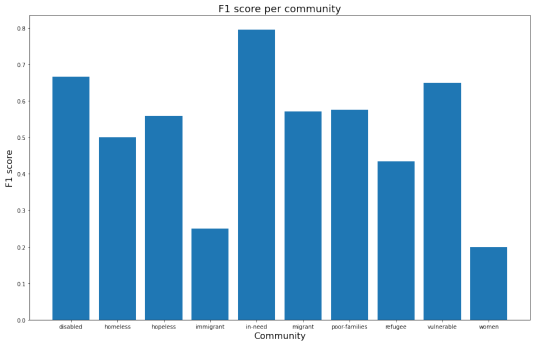

Figure 5: F1-score charted against community labels. Observe that the predictive accuracy of the classifier favours certain groups.

of examples classified with scores between 2 to 4. Here we also find the model performs the worst where the PCL score for a text is 2, which is promising, as this is where the potentially patronising content is most suitable.

Concerning sequence length, there appears to be no discernible relationship between the length of a text and the model's ability to predict whether it contains PCL. This finding is intuitive, since theory informs us that transformers are designed to be able to model long-term dependencies through the attention mechanism (4). As such, with the maximum text length in the dataset being no more than several hundred words long, we would not expect to see a degradation in the performance as the sequence length increases. This is precisely what we observe in Figure 6.

As noted in Section 1, different community labels are subject to varying degrees of PCL. As such, it is a reasonable assumption that the model's ability to predict PCL will be contingent on the community to which any given sample text refers. We visualise this relationship in Figure 5, and indeed see that the transformer's predictions yield the highest F1-scores when predicting PCL for the in need, disabled and vulnerable, reasonable F1-scores for the homeless, hopeless, migrants and poor families, and finally poor F1 for immigrants and refugees. There is some overlap here with respect to the distributions of Figure 1, in that generally, communities with higher distributions of PCL are better predicted by the model, with the in-need standing out as a prime example. There are, however, exceptions, for example the vulnerable, which are underrepresented as targets of PCL in the dataset, but predicted especially well by the classifier.

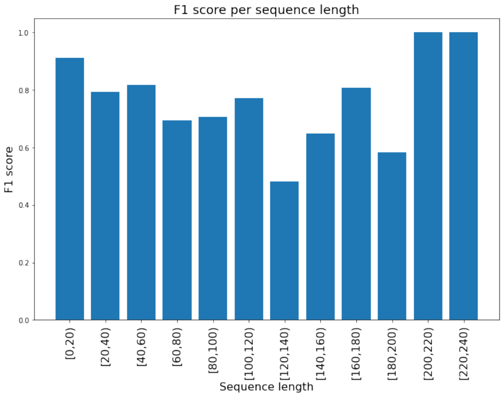

Figure 6: F1-score charted against input sequence length, where sequence length is discretized into bins.

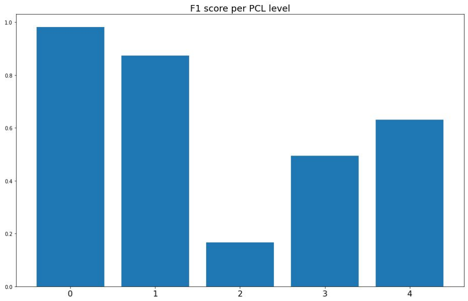

Figure 7: F1-score charted against degree of PCL. Note that the model performs best where the scores indicate consensus between annotators.

5 Conclusion and summary of results

Coupled with upsampling techniques, the final transformer model was able to achieve an F1-score of 0.55 on the internal development set and 0.58 on the official development set. This comfortably exceeds the RoBerta baseline F1-scores of 0.48 and 0.49, and significantly outperforms the baseline multinomial naive Bayes and SVM models, which achieved F1-scores of 0.37 and 0.31 on the internal development set respectively. This demonstrates a respectable ability to detect PCL, which the authors find to be a satisfactory result. In future work we would explore how stronger neural baselines such as LSTMs compare against our model, as this would provide a more realistic comparison as to how suitable transformers are for the present task.

References

[1] Carla Perez-Almendros et al (2020). 'Don't Patronize Me! An Annotated Dataset with Patronizing and Condescending Language towards Vulnerable Communities'.

[2] DeBerta Base. https://huggingface.co/microsoft/deberta-base.

[3] OpenAI API: https://openai.com/blog/openai-api accessed 07-03-2023

[4] Ashish Vaswani et al (2017). Attention Is All You Need: https://arxiv.org/pdf/1706.03762.pdf

[5] Alessio Miaschi et al (2020). Contextual and Non-Contextual Word Embeddings: an in-depth Linguistic Investigation.