Classifying patronising and condescending language using NLP

Megan Howard
mh2919@ic.ac.uk

Benedict Newton-Ingham
bn120@ic.ac.uk

Anjali Ramesh
ar4323@ic.ac.uk

Abstract

This report provides an account of identifying patronizing and condescending language using a language model. In order to achieve this classification aim, the necessary exploration and discussion of the 'Don't Patronize Me!' annotated dataset are presented. Next, a detailed account of experimentation performed to identify techniques that enable the model to perform better than a baseline RoBERTa model is given. The performance of the model found during experimentation is then analysed against features of the dataset. Finally recommendations are provided as to future experimentation with this approach.

1 Introduction

In this report we tackle the challenge of assessing patronising and condescending language (PCL) using the 'Don't Patronize Me!' dataset (Pérez et al., 2020). While the machine learning community has made significant progress in identifying overtly harmful text (Zampieri et al., 2019), PCL content has proven more challenging to identify and classify, though still significantly harmful to vulnerable communities. The creation of the 'Don't Patronize Me!' dataset aimed to enable the NLP community to improve the modelling and detection of PCL and was done collecting textual data containing categorical keywords (e.g. disabled, immigrant, women, etc) from English language in web news sources (Davies, 2013). This text was then evaluated to three expert annotators. Two annotations provided a score of zero (indicating no PCL), one (borderline case) and two (definite use of PCL) for the entire dataset. Where the scores from the two annotators completely differed(i.e. scores of zero and two), the third annotator provided a verdict on the degree of PCL used. These scores were summed and all instances with a total score greater than 1 would be categorised as PCL (Pérez et al., 2020).

This paper first contains initial data analysis and discussion of the nature of the problem. Then the experimental methodology and model exploration details how we found the best performing model. Subsequently, this model is analysed against a baseline model, and also across different text distributions (e.g. varying PCL scores, sequence lengths and categories). Finally, a conclusion is presented with some proposals for future experimentation.

2 Data Exploration

The train set contains 10469 entries, each with seven features, including the keywords from the text's paragraph, the article's country of origin, the text itself, a binary PCL classifier label and the original label between zero and four.

The first and most stark observation was the large discrepancy between the number of negative to positive cases in the binary classification labels. The positive (patronising) texts make up only around 9.5% of this dataset. This disparity in size is further pronounced in the distribution of the original labels. Naturally the samples labelled either group 0 or 4 are the most distinct and expressive examples. However, the relative scarcity of the distinct PCL cases compared to those that are not PCL cases is clear when considering that 0 labeled text makes up 81.4% of sample while 4 labeled text makes up 3.7%.

Comparing the feature distributions of both binary classes, in Figure 8, there is no apparent predictive difference in varying string lengths between the two classifications. Since they both have the same distribution of string lengths, this feature doesn't correlate with the inclusion of PCL. When further evaluating the text characteristics, such as the average word length, sentence length and number of words per sentence, again no significant differences could be identified. This lack of difference informed us that the model must focus on the content rather than any immediately quantifiable property of the text.

Then the relationship between the categorical features and the proportion of binary labels was explored. The most pronounced variations were seen in the proportions of labels relative to the keyword feature, as seen in Figure 9. This difference in the distribution is likely due to the context in which the word appears. For example, the migrant is the correct technical term for a person who moves from one place to another; however, keywords such as 'poor families' have a much more emotional connotation when compared to 'lower-income families' for example. Investigating the country of origin showed only minor deviations across the binary labels.

Identifying subtle and often unintended PCL is complex by its very nature; evidenced by the many samples whereby the first and second annotators didn't agree. There were 1457 disagreements overall with 590 stark disagreements (i.e., one annotator scored the text as zero, and the second scored it as two) (Pérez et al., 2020). The scale of this problem is especially striking when it is considered that there were only 391 samples classed by both annotators as clearly PCL. Hence, most of the text classified as PCL in the binary system was text that one of the annotators classified as borderline, if not both (602/993). This highlights how highly subjective this task is, even for humans. Hence, with the small number of annotators and the binary nature of this classification problem, it is an inherently difficult task with noisy data. This makes the task more challenging than in other more overt domains for machine learning systems to be able to classify effectively.

3 Modelling

3.1 Model Setup

The RoBERTaForSequenceClassification model is chosen for the task of classifying text as patronising or not due to its strengths in sentiment analysis and high accuracy in text classification. Benefiting from training on a larger corpus and longer durations than BERT, RoBERTa captures language nuances and context more effectively. This is crucial for handling datasets like Don't Patronize Me!, where RoBERTa excels in identifying minority classes, further enhanced by data augmentation and ensembling strategies. It employs a cased Byte-Pair Encoding tokenizer, preserving text case to distinguish meanings (e.g., "US" vs. "us"), hence the decision against preprocessing text into lowercase. Hyperparameter tuning was conducted with an internal dev set, keeping the official dev set untouched. The model showed a tendency to overfit, as indicated by the divergence of training and validation loss with increased epochs, suggesting memorization over learning - Figure 5.

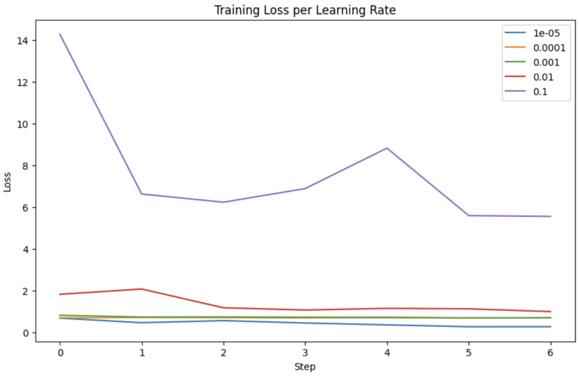

Figure 1: Training loss for varying learning rates

To combat overfitting, various learning rates were tested, with 10^{-5} showing lower training loss and stable validation loss and F1 scores, indicating less overfitting compared to higher rates - Figure 1. A learning rate scheduler, which increased linearly after 500 warmup steps, and early stopping were implemented to address overfitting, with early stopping reducing necessary epochs to 3-4 for optimal performance - Figures Figure 6 and Figure 7. Validation loss guided early stopping to ensure the model generalises well to unseen data. For simplicity and to mitigate overfitting, training labels were mapped to binary classifications by considering original labels of '0' or '1' as not patronising, and labels of '2', '3' and '4' as patronising.

3.2 Model Improvements

The 'roberta-base' model was improved upon using data preprocessing, upsampling, and data augmentation for class balance and diversity, and by using a weighted loss function to account for any remaining imbalance. All F1 score and accuracy metrics in this report that were used testing are evaluated on an internal dev set.

Firstly, preprocessing was applied to the input data in which stopwords and punctuation were removed, contractions were expanded, and words were lemmatised. Although the removal of stopwords and punctuation can help reduce noise in the data and allow it to focus on more meaningful context, we were mindful that over-cleaning can lead to the loss of contextual nuance that may be important for detecting patronising content, such as tonal emphasis conveyed by the use of exclamation marks. One of the strengths of RoBERTa is understanding the context of each word within a sentence while considering the entire sentence structure, including punctuation and the order of words. Lemmatisation and preprocessing can disrupt this context and lead to a significant loss of meaning. In this case specifically, stopwords and punctuation can provide valuable cues about the sentence's tone, intention, connotation, and emotion; all of which are strong indicators of patronising content.

Input Training Data	Accuracy	F1-Score
No preprocessing, translation	0.92	0.60
Preprocessing, translation	0.92	0.49
No preprocessing, translation, synonyms	0.88	0.45
Preprocessing, translation, synonyms	0.88	0.43

Table 1: Effect of varying preprocessing and data augmentation on model performance, tested on internal dev set

Table 1 shows that when using F1 score as a metric, the best perforing model was trained on un-preprocessed data.

To account for the disparity in negative and positive labels in the training dataset, the minority class was up-sampled, to mitigate class imbalance and improve the model's ability to learn those instances. By providing more examples of the positive class, the model is less likely to miss patronising instances which improves recall. To do this, backtranslation was used on just the positive class to introduce linguistic diversity and provide varied ways of expressing the same idea, and this was appended to the entire original preprocessed dataset. This is a robust method for generating new samples, as the semantic meaning and sentiment of the original text is preserved; especially crucial for this task. To further augment the positive class, synonym replacement was implemented on adjectives and adverbs in each text input. The synonym-replacement method used WordNet through NLTK to find all synonyms for a given word based on its part-of-speech tag. While this technique can, in theory, increase the diversity of the training data, it resulted in poorer model performance. Although the replaced words were limited to adjectives and adverbs (which are less likely to impact the meaning of a sentence than verbs or proper nouns) in hopes of preserving the meaning of the text, there was still loss of specificity that disrupted the subtle nuances crucial for classifying this text. Inappropriate synonyms might not have fit the sentence well or had rarely been used in the given context, thereby reducing the quality of the augmented data.

The 'keyword' and 'country' categorical labels are appended to the 'text' column, as they provide contextual cues that may influence the interpretation of the text and lead to better classification. Certain keywords might be associated with patronising language, or the way patronising language is used may vary by country. By adding these to the analyzed text, the feature space of the model is effectively increased without any manual engineering, and the model can learn associations between these additional context words.

Lastly, a weighted loss function helps the model to focus on the minority class, effectively addressing the remaining imbalance. Cross entropy loss is used with computed weights (calculated based on the distribution of the labels in the training set, where the 'y' values are used to automatically adjust weights inversely proportional to class frequencies in the input data) to ensure that the loss calculation considers the imbalance in the classes. This led to a significantly better performing model.

3.3 Baseline Model Comparison

3 baseline models were implemented: logistic regression, TF-IDF and Bag of Words (BoW), each using the raw training set and a downsampled training set, and the base RoBERTa model to establish a baseline F1 score. In Table 2, all 4 versions of the logistic regression models showed high accuracy but poor F1 scores, indicating that

Model	Training Data Used	Accuracy	F1-Score
TF-IDF	raw	0.905	0.019
BoW	raw	0.898	0.219
TF-IDF	down-sampled	0.886	0.279
BoW	down-sampled	0.825	0.319
RoBERTa base	down-sampled	0.843	0.481

Table 2: Performance of baseline models with varying training data, tested on internal dev set

they are biased towards predicting the majority class well and struggle with the patronising class. BoW with downsampled data seemed to strike the best balance between identifying both classes. It had the highest F1 score among the models, suggesting that while downsampling reduces overall accuracy, it improves the model's ability to correctly classify the minority class.

The BoW model creates features based on the occurrence of words within the training data. Commonly used words in the dataset become features, with their counts in each document representing their value. This approach is straightforward but does not account for any word context or significance beyond this as the improved RoBERTa model does. This model could misclassify a text as non-patronising when it does actually contain subtle patronising content in the case that the text used uncommon words or phrases to convey patronising sentiment that was not well-represented in the training data. Downsampling also improves performance compared to the raw data, as there is less class imbalance. In a high-dimensional feature space such as this, many features may provide very little value and lead to a model that is not as discriminatory as it needs to be for this task. Compared to these baseline models, the final tuned model increases the F1 score by around 0.12, most significantly by accounting for the imbalanced classes and mitigating any biases in the data.

3.4 Model Results and Hyperparameter Tuning

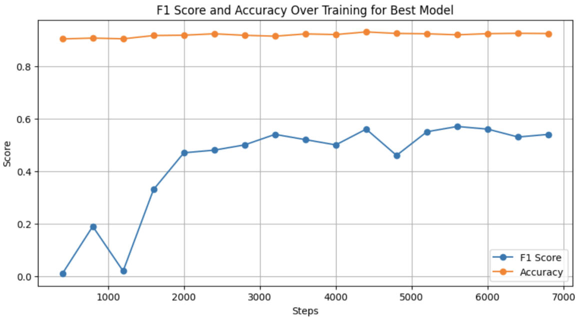

Figure 2: Accuracy and F1-scores over steps for best performing model

Figure 2 shows the F1 score and accuracy scores of the final model evaluated on the internal dev set. The F1 score increases through training and then levels out, suggesting that over epochs, the model becomes better at balancing precision and recall, particularly for the more challenging class (patronising text). Plateauing indicates that the model has reached a point where further training will not significantly improve this balance. In the context of this imbalanced dataset, where the cost of false positives and negatives is high, the increasing F1 score implies improvements in handling the minority class. The model's accuracy also remains stable at around 0.9, suggesting that the model is generally well-suited to the task learns effectively. This model, optimised with hyperparameters from part (b) —notably learning rate, learning rate scheduler, early stopping, and epoch number—demonstrated best performance on the official dev set with a learning rate of 1e-5, weight decay of 0.01, and a 3-epoch training duration. This learning rate, typical for transformer-based models, ensures effective learning with appropriately sized updates. Early stopping identified 3 epochs as ideal, allowing quick adaptation and preventing overfitting. A weight decay of 0.01 for L2 regularisation, also curbs overfitting by penalising large weights. Despite minor performance gains from the learning rate scheduler, reducing epochs addressed its issues without significantly impacting the F1 score.

4 Analysis

4.1 Model Performance on Higher Levels of Patronising Content

Figure 3 shows that the model performs much better in predicting labels for text at the more polarised ends of the spectrum, with higher accuracy and F1 scores for patronising levels 0 and 4. This suggests that texts with a more unambiguous indication of being either highly patronising or not are

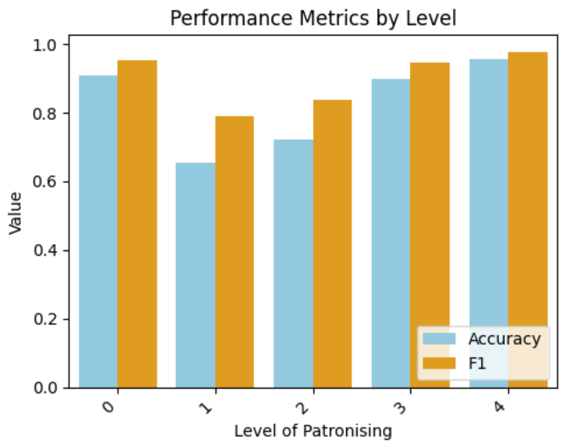

Figure 3: Accuracy and F1-scores over different levels of patronising content

more straightforward for the model to classify correctly. Texts falling under the middle degrees of patronising (1, 2, and 3) performed poorer in the metrics, reflecting more challenges in their classification. These results imply that more nuanced or moderately patronising texts introduce ambiguity, leading to reduced predictive performance and that the model is stronger at distinguishing texts with clear-cut characteristics of patronising content.

4.2 Length of Input Sequence

The length of the input sequence also impacts the model's performance, as we see variations in accuracy and quite significantly in F1 scores across different text length ranges - figure **??**. There is not a very pronounced trend in accuracy; it fluctuates in the range of about 82-90% up until around 800 characters, where it starts to become unreliable. Similarly, F1 score initially improves with increasing text length, suggesting that longer texts provide more contextual information, aiding accurate predictions. However, at a similar text length to that seen in accuracy performance, the F1 score peaks before becoming erratic at extreme text lengths. This is probably due to sparse data in those ranges or overfitting to specific text patterns. In some of the longest text ranges, perfect scores are observed, which could be due to very few examples in those bins making it easier for the model to predict correctly, but doesn't seem to be a reliable or generalisable trend. Therefore, while longer texts seem to improve model performance by providing more context, extremely long or short texts seem to lead to inconsistent results due to sparsity or overfitting in the entries used for training in these ranges.

4.3 Model Dependence on Data Categories

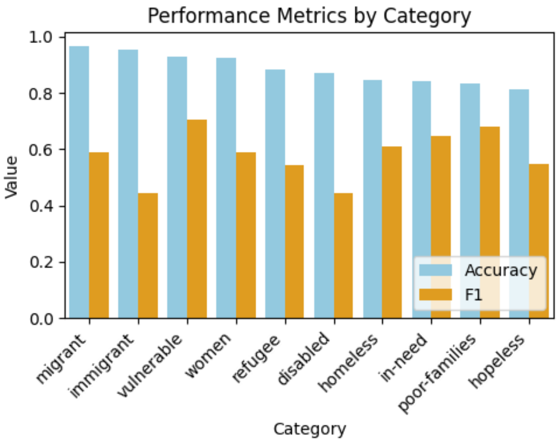

Figure 4: Accuracy and F1-scores over different text categories

We see from figure 4 that both the F1 score and accuracy vary notably across different categories. The model performs more accurately in the ”migrant,” ”vulnerable,” and ”poor-families” categories specifically, though their F1 scores are relatively average. The categories with the lowest accuracy, such as ”hopeless” and ”homeless,” show more challenges in the label predictions and hence performance metrics. This variation could perhaps be due to differences in the data distribution across categories in training relative to testing, as well as perhaps more distinct language used against the categories with better performance metrics. This would cause varying levels of complexity in classifying certain categories one way or another and may explain the trends seen.

References

Mark Davies. 2013. English-corpora: Now. Corpus of news on the web.

Carla Pérez, Almendros Luis, and Espinosa-Anke Schockaert. 2020. *Don't Patronize Me! An Annotated Dataset with Patronizing and Condescending Language towards Vulnerable Communities*.

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)*.

A Appendix

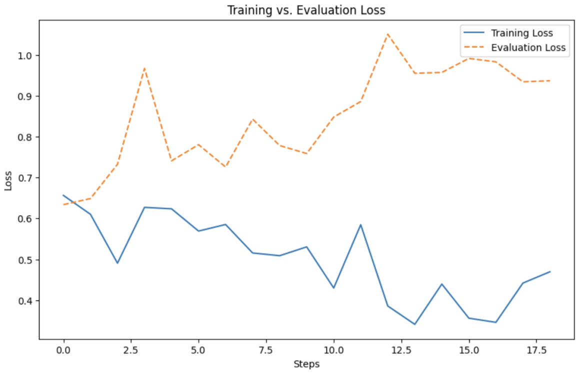

Figure 5: Divergence in training vs.validation loss due to overfitting

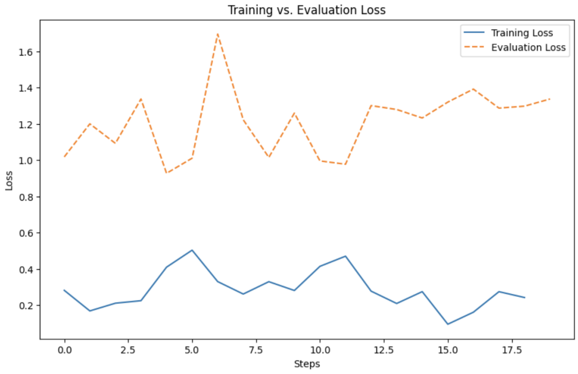

Figure 6: Training and validation loss using learning rate scheduler

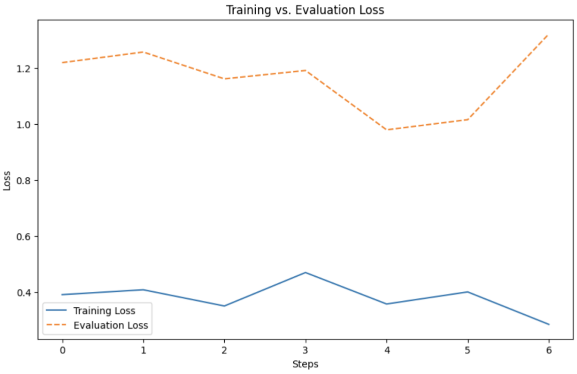

Figure 7: Training and validation loss after early stopping

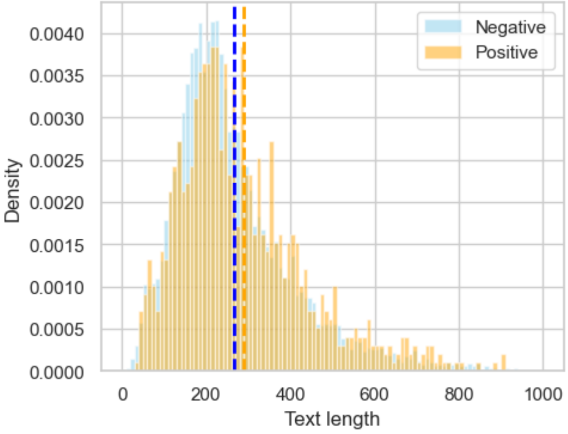

Figure 8: A histogram to show the density of the text lengths of the positive (PCL) and negative (non-PCL) class. This histogram has been clipped to remove a very small number of extremely long texts. The blue and orange dashed lines on the figure shows the mean values.

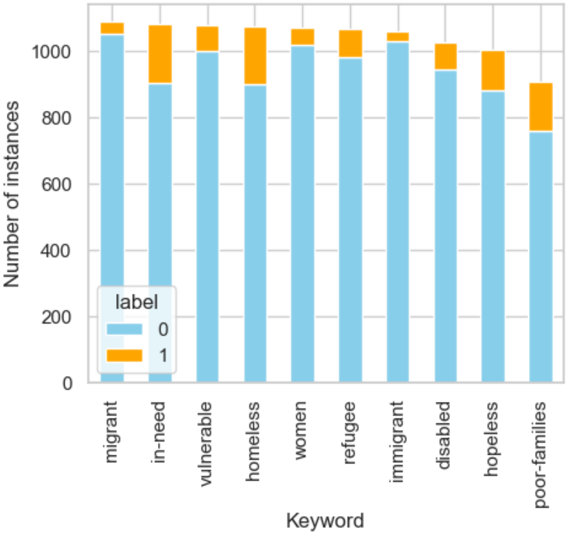

Figure 9: A stacked bar chart to show the relative split of each class, 0 (non-PCL) and 1 (PCL), by the keyword used for extraction.

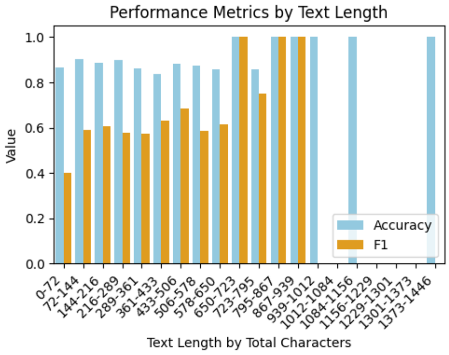

Figure 10: Accuracy and F1-scores over different text length ranges