NLP Coursework

Jorge Gallego Feliciano
jg2619@ic.ac.uk

Lorenzo Stigliano
ls1121@ic.ac.uk

Adrien Carrel
ac622@ic.ac.uk

1 Introduction

The task is to build a binary classification model to predict whether a text contains patronising or condescending language (PCL). PCL is defined as language that expresses "superior attitude towards others or depicts them in a compassionate way." (Pérez-Almendros et al., 2020). Detecting PCL in text is a challenging task for natural language processing (NLP) models due to the fact that PLC is usually subtle and difficult to detect, resulting in a difficult challenge for language models.

To address this challenge, we utilize an annotated dataset created in the paper "Don't Patronize Me! Patronizing and Condescending Language Towards Vulnerable Communities" (Pérez-Almendros et al., 2020). The dataset aimed at supporting the development of NLP models to identify and categorize PCL language towards vulnerable communities. It consists of paragraph extracts from several news media outlets and articles, categorized into ten groups: Disabled, Homeless, Hopeless, Immigrant, In-need, Migrant, Poor families, Refugee, Vulnerable and Women. The data is collected from 20 different countries, providing a diverse and comprehensive set of examples for model training and evaluation.

2 Data Analysis

As shown in Figure 1, the dataset used in this study exhibits clear class imbalance, with approximately 1k PCL texts and over 9k No PCL texts. We also categorized them into different levels of severity. Appendix A Figure 4 illustrates that the majority of PCL texts fall into either level 3 or level 4 categories.

In Appendix A Figure 5, we present the distribution of class labels across different article types. The results indicate a relatively uniform distribution of both PCL and No PCL texts. Among No PCL texts, the most frequent article types include migrants, immigrants, and women. In contrast, homeless, in-need, and poor families are the most common article types associated with PCL texts. Notably, poor families have the fewest No PCL texts. This finding suggests that certain vulnerable groups are disproportionately represented in media coverage, potentially subjecting them to higher levels of PCL language. The distribution of labels per country code is shown in Figure 6 We observe that we have only between 20 to 80 examples of PCL texts for each country.

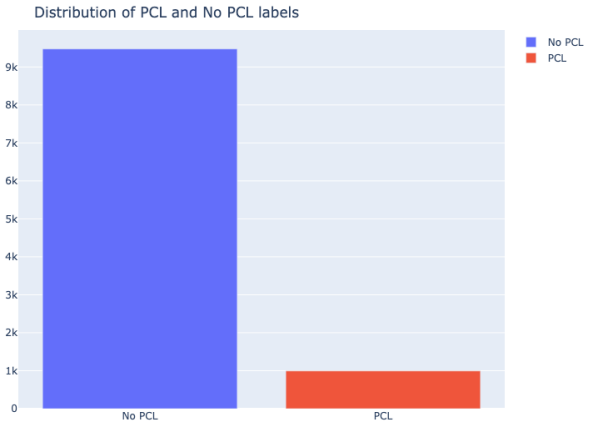

Figure 1: PCL and No PCL distrbution over the dataset.

We also analysed PCL and No PCL texts with respect to the length of the text. In Figure 2 we can see the distribution of the text lengths. The median text length for PCL is slightly greater than for No PCL. Analysis per article type and country can be found in Appendix A Figures 8 and 9. Our results show that the length of PCL text varies across vulnerable groups. For instance, in the case of "Disabled" article type, PCL texts tend to be shorter compared to "Women" where the texts are longer. Similarly, we observed that text length varies across countries, which could affect the model's prediction power for countries with

shorter texts.

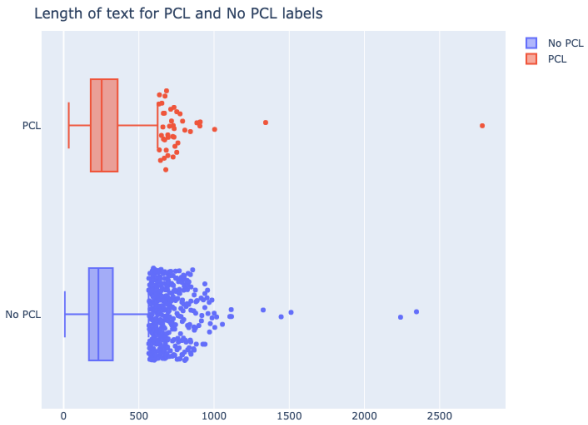

Figure 2: Box plot showing length of text for PCL and No PCL texts, for text length less than 2500 characters.

Detecting PCL is a challenging task due to its subtle nature and reliance on contextual factors. We have identified several factors that could affect the difficulty of identifying PCL, such as; the degree of PCL, the differentiation between intent and impact, social norms and non-verbal cues.

PCL can *vary in degree* from subtle to overt. To illustrate this, we presented two PCL texts at level 2 and 4. "She reiterated her ministry's commitment to put in place the necessary legal and policy framework to address all issues that affect women's rights and gave a strong indication of hope for Ghanaian women." (par id:6249) and "Women are generally not as competitive as men, and not as motivated by job prestige..." (par id:1194). Level 2 might be challenging to identify, while a text at level 4 is more apparent.

It is also crucial to differentiate between *intent and impact*. Oftentimes individuals may not have intended to be patronizing, yet their words convey such an attitude. For example: "Mombasa county team manager Anisa Abdala called on the corporate community to sponsor various teams as a way of showing solidarity with the disabled." (par id:715). While the intention is positive, the language used still exhibits PCL undertones.

It is also important to consider *cultural and social norms*. Communication styles vary across cultures, and what one person perceives as PCL may be seen as perfectly acceptable by someone from a different culture. Cultural sensitivity is essential when identifying PCL language. The quote from New Zealand, "Instead of passively paying a sickness benefit for 40 years, for example, we want to take steps to intervene now to help vulnerable New Zealanders get a job, lead a better life, and save the Government money in the long run." (par id:438), may be seen as patronizing by some, while others view it as a positive approach to aid the vulnerable population. In a similar vein, in India, the statement, "such crimes against women and weaker sections, especially minorities, It is time people, police and government agencies stood up to discharge their duties to protect the weak and vulnerable sections of society." (par id:2803) acknowledges the need to safeguard vulnerable groups but employs the term "weak" which could be considered condescending in other cultures. These instances of PCL in New Zealand and India highlight how societal and cultural norms can shape the use of PCL.

Nonverbal cues such as tone of voice, facial expressions, and body language also play a role in the perception of PCL. However, these are not available when only using the text.

In conclusion, detecting PCL can be challenging due to various factors such as different degrees of patronizing or condescending language, differentiation between intent and impact, cultural and social norms and nonverbal cues.

3 Modelling

3.1 Model Description

We decided to use a cased model, PCL language often involves the use of capitalization and punctuation, which can convey important nuances in meaning. By using RoBERTa-cased, the model is better able to capture these nuances and improve its performance. On top of the transformer architecture, we added one fully connected layer with a probability output (768 to 1) for classification. We have frozen the weights of RoBERTa, and we clip the norm gradient while training.

We use the following hyper-parameters; learning rate: 5×10^{-5}, batch size: 32, and number of epochs 30, with results displayed in table 3. In order to tune the number of epochs we used early-stopping by manually interviewing when the F1-score stagnated. We found 30 epochs to be a good early-stopping criteria. Furthermore, adding a step learning rate scheduler improved the performance ($\gamma = 0.9$, step of 5 epochs, Adam optimizer with $\epsilon = 1 \times 10^{-8}$).

We're using the class labels for the loss and to calculate class weights. Since class imbalanced still posed to be a problem we decided to solve this

by introducing a weighted loss, in particular, Cross Entropy Loss weighted by class, this encourages the model to pay more attention to the minority classes during training, leading to better overall performance on the dataset (Lin et al., 2017).

3.2 Model Improvements

3.2.1 Pre-processing

We focused on several pre-processing methods: white-space normalization (A), punctuation normalization (B), replacement of stop-words with a standardized token (C) and substitution of numbers with a numeric token (D). We based our list of stop words in the NLTK library, but edited the list by removing pronouns such as 'I/their/ourselves', which express the authors' position with respect to a person or group. Also, verbs such as 'need/should/must' can be PCL, as they express knowing better than the addressed person. We experimented with various combinations of these pre-processing methods to analyse their impact on the models' performance.

Pre-processing technique	F1-Score	
	Train Set	Dev Set
A	0.483	0.495
A + B	0.485	0.501
A + C	0.469	0.486
A + B + C	0.483	0.493
A + B + D	**0.501**	**0.518**

Table 1: Results from different pre-processing techniques. Learning rate and batch size defined in the pre-trained model (4×10^{-5} and 8). Trained for 10 epochs.

The results presented in Table 1 clearly demonstrate the significant impact of pre-processing on the performance of the model. The F1-scores sets improved noticeably with (A)+(B)+(D). Thus, in order to ensure the best performance of the model, we apply pre-processing to all future analyses, excluding (C).

3.2.2 Data Augmentation

The class imbalance, as seen in Figure 1, motivated us to explore data augmentation techniques to increase the number of PCL examples. We employed the "nlpaug" library (Ma, 2019), focusing on word-level techniques. These included synonym and antonym substitutions, random word deletion (Del) and substitution (Sub), and contextual word embedding (CWE) substitution using the pre-trained model, RoBERTa. For chained augmentations, we either substituted or inserted words with equal probability into the text with CWE. Additionally, we explored character-level augmentation techniques such as keyboard augmentation, which involves adding characters based on keyboard distance. At the sentence level, we used CWE to add a sentence at the end of the text. For all combinations, we used the best pre-processing techniques described in the previous section. The model was trained for 10 epochs. More importantly, we up-sampled and down-sampled such that the split of No PCL to PCL was $1:2$. Thanks to data augmentation we do not remove as many No PCL samples.

Data Augmentation technique	F1-Score	
	Train Set	Dev Set
Keyboard Character	0.472	0.473
Random Word (Sub)	0.479	0.474
Random Word (Del)	0.473	0.470
Synonym	0.504	0.521
Antonym	0.482	0.476
CWE (Sub)	0.512	0.532
Chained CWE	**0.513**	**0.536**
Sentence CWE	0.483	0.487

Table 2: Results from different data augmentation techniques. Learning rate and batch size defined in the pre-trained model (4×10^{-5} and 8). Trained for 10 epochs.

Table 2 illustrates the varying performance of different data augmentation techniques, with some techniques proving to be more effective than others. Notably, keyboard character insertion resulted in a drastic decrease in model performance. This is not surprising, as it involves creating words that do not exist, which is irrelevant for articles that typically have accurate spelling. Random word substitution, deletion and antonyms was also detrimental, as it can disrupt the context. Deletion of words was particularly problematic as the relative distance between words can affect the transformer's understanding. In contrast, CWE and synonyms in combination with word substitution or insertion improved performance as it maintained the meaning of the augmented texts, making them effective synthetic samples. As a result, chained CWE were used for the final model.

3.2.3 Other Techniques

Another technique which we used in order to improve model performance was to tune the learning rate and batch size. The results in Table 3,

are achieved by using the pre-processing and augmentation techniques described in Sections 3.2.1 and 3.2.2. We can see that a larger learning rate and a lower batch size resulted in the best results for both train and dev set.

Hyper-parameter combination	F1-Score	
	Train Set	Dev Set
$[2 \times 10^{-5}, 32]$	0.499	0.520
$[2 \times 10^{-5}, 64]$	0.493	0.521
$[5 \times 10^{-5}, 64]$	0.516	0.539
$[5 \times 10^{-5}, 32]$	**0.516**	**0.557**

Table 3: Hyper-parameter tuning for model. The hyper-parameters are defined as [learning rate, batch size], trained for 30 epochs.

We also tried using the extended PCL labels to train the model. For this purpose, we changed the output of the model to 5 classes, corresponding to the 5 PCL levels. We applied a Softmax activation function to turn the logits into probabilities for each class and calculated a weighted cross entropy. We weighted the classes on their proportion in the train set. Finally, we computed backward propagation with an error that had two terms: the multiclass cross entropy loss, and the original binary entropy loss corresponding to the binary label PCL vs no PCL. The idea is that in multi classification, we penalize the model by the same amount when it guesses 1 instead of 0 vs 1 instead of 2, but our original goal is binary classification, so to balance the two objectives, we add the losses. Also, we weight term proportions with a constant α. We couldn't experiment much with choosing an appropriate α and other tunings of this model, and so it didn't achieve great performance metrics, with a final F1 score of 0.45 after 10 epochs.

3.3 Model Performance

We implemented a couple of baseline models to compare with. For a quick comparison, refer to Table 4 to compare F1 scores. No class rebalancing and minimum pre-processing was done for these models.

The most basic baseline we implemented is a model that always outputs the label 0 (no PCL), independent of the input. As the data is quite unbalanced, the results showed the F1 score of this model was quite high. It made us realize that this is a difficult classification task, as complex models such as the RoBERTa model only achieves to improve the baseline by 0.015 on the dev set.

Next we implemented a logistic regression model with the following features: length of the document, the targeted group label (categories in the original dataset i.e. 'homeless' or 'disabled'), the country code, both one-hot encoded, and finally, an average of the word embeddings using a pretrained model (Mikolov et al., 2013). The scores of the logistic model outperform the RoBERTa baseline by 0.05. Our final model in comparison outperforms the logistic regression model by 0.023.

Finally, we implemented a Bag of Words (BoW) model as a naive Bayes classifier. This model takes around 10 seconds to compute all the count tables, and can start predicting immediately after. The unigram version does very poorly, similar to the dummy model, and worse in the dev set. The bigram version does surprisingly well, with an F1 score of 0.53. Simple heuristics can do very well.

We analysed cases of misclassification by looking at the highest scoring bigrams in false positives, and found that bigrams such as "poor families" and "homeless people" were very common, as these words make an explicit reference to a vulnerable group and tend to have a high score in the BoW. Other common bigrams are "he would", and "they will", which contain pronouns and verbs like "would" or "need" that places the writer in a position of power as it expresses a strong opinion of what's best for someone else.

For a list of the most indicative bigrams and unigrams, check Table 5.

Model	F1-Score	
	Train Set	Dev Set
RoBERTa	0.480	0.490
Dummy (always 0)	0.475	0.475
Logistic Regression	0.527	0.534
BoW (Unigram)	0.495	0.483
BoW (Bigram)	**0.916**	0.531
Final Model	0.516	**0.557**

Table 4: Comparison of final model with different baselines.

Furthermore, Figure 3 shows the learning curve of our final model on the dev set, we can see that the loss is decreasing and that it manages to surpass the RoBERTa's performance of after 5 epochs.

In conclusion, our final model exhibits the highest F1-score on the dev set. It is not surprising that

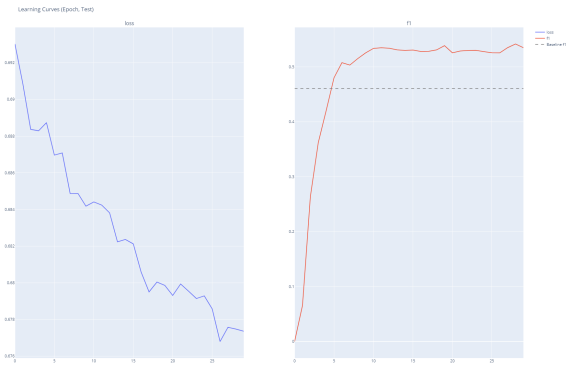

Figure 3: Learning curves for final model on dev set.

the BoW model outperforms other models on the training set since it has the tendency to overfit to the training data, resulting in inflated scores.

4 Analysis

1. Is the model better at predicting examples with a higher level of patronizing content?

As we can see in figure 10, our model has a very good score in the PCL levels, with an F1 score of around 0.8 for all categories. We have trained a model to identify PCL texts very easily, with a slight tendency to compute false positives, lowering the total F1 score. This behaviour is desired as this could be used as a tool to flag articles to be rechecked by a human, for example in a publishing company. There is no indication that the model is better at detecting strongly PCL language, as the level with the highest score is actually level 2.

Although level 2 is the group with the least number of examples, it still performs quite well. This could be due to the data augmentation and class balancing techniques. It also provides a reason why the multiclass prediction didn't work for our model. If our model was much better at predicting stronger PCL levels to lower ones, a multiclass loss term addition could help smooth the gradient and move the model in the right direction, but it is not the case of our model.

2. How does the length of the input sequence impact the model performance?

The model is better at predicting longer text sequences, and has a very bad performance with lower length texts. After a length of 500, the F1 score stops increasing, and so we conclude that the model requires at least a text of this length to perform accurate predictions. Below 500, the F1 scores are below 0.5, getting to as little as 0.05 for short texts.

This is due to the self-attention mechanism of our transformer model, that allows the model to assign different weights to different positions in the input sequence, based on their relevance. Moreover, multi-head attention is utilized, enabling the model to attend to various segments of the input sequence simultaneously. This feature can be especially beneficial for capturing long-range dependencies, as it permits the model to attend to multiple relevant positions in the sequence simultaneously.

3. To what extent does model performance depend on the data categories?

For this section, we found it appropriate to visualize the final F1 scores with the proportion of PCL documents in each category. The migrant and immigrant categories had the lowest scores in the data set, with a score close to 0.2, which coincide with the two classes with the least proportion of PCL examples, so this explains why it scored so low on those cases. The rest of the categories are all very close to 0.57, with varying proportions of PCL examples. The class "poor families" is an outlier with a score above 0.6, and the class "women" is below the average but not by much. The baseline models we implemented had a similar score weighting for the classes, but our model does much better in the "women" category, which is close to 0.2 in the BoW baselines.

5 Conclusion

Our experiments have shown that detecting PCL is indeed a difficult task, for the many reasons exhibited in section 2. Models for natural language processing are not very explainable, and are very sensitive to perturbations in the dataset, such as removal of stop words. As an example, for the BoW model, removing stop words had an impact of reducing the F1 score by 0.2. A human is still capable of understanding texts without stop words, but the inner workings of transformer models are slightly opaque and difficult to predict.

A further improvement we suggest is to add a loss term of prediction of the targeted group label, as it will push the model to identify key information for PCL detection. We could also improve the classification of extended labels. Another improvement would be to down-sample while respecting the article and country distributions.

References

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

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*.

A Appendices

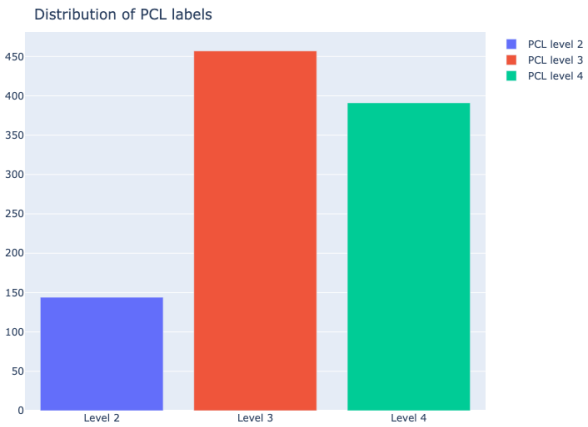

Figure 4: PCL distrbution by level of severity.

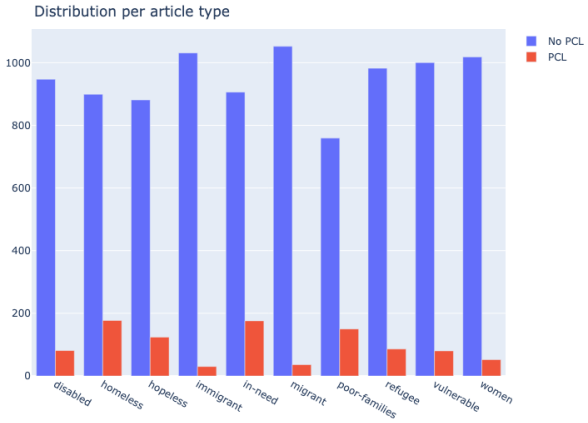

Figure 5: PCL and No PCL texts separated by article type.

Figure 6: PCL and No PCL texts separated by country code.

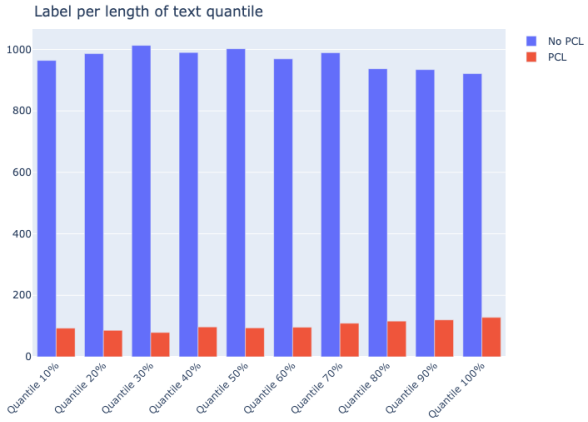

Figure 7: Quantiles of text length per label.

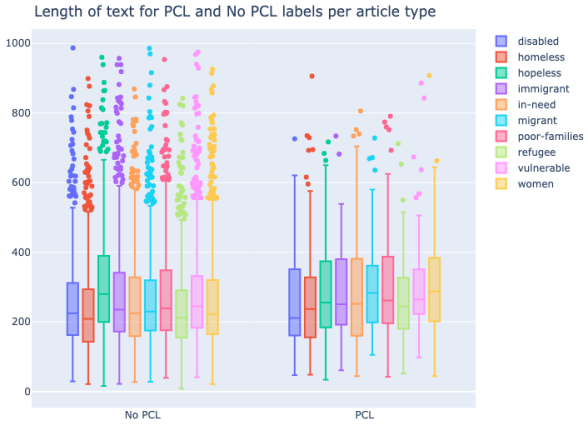

Figure 8: Box plot showing length of text for PCL and No PCL texts per article type.

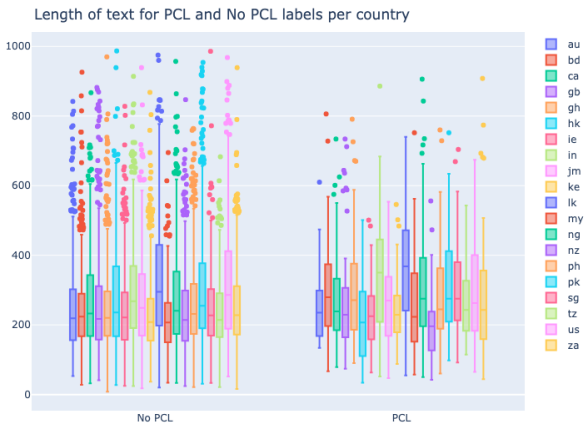

Figure 9: Box plot showing length of text for PCL and No PCL texts per country.

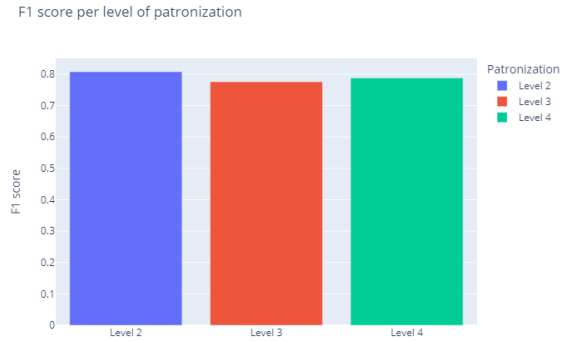

Figure 10: Box plot showing F1 score per level of PCL severity.

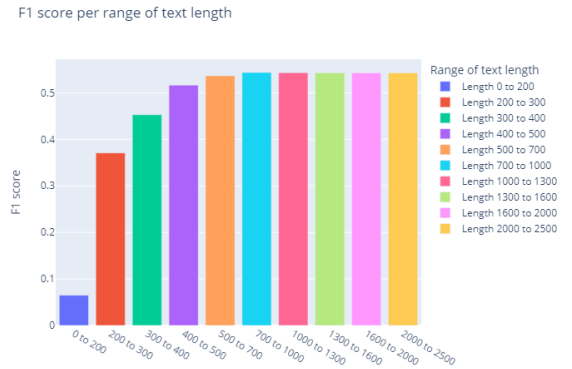

Figure 11: Box plot showing F1 score per length of text.

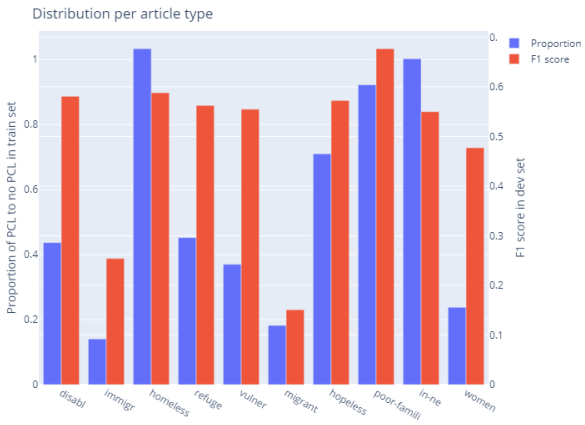

Figure 12: Box plot showing F1 score per text category.

Rank	Unigram	Bigram
1	diaper	donat blood
2	volunteer	our street
3	yong	desper need
4	warmth	dark of
5	waysid	that come
6	poverty-stricken	you educ
7	huddl	feed homeless
8	indig	the digniti
9	evan	restor hope
10	sabc	fm in
11	papal	give hope
12	dinu	children orphan
13	akka	the recognit
14	teresa	the helpless
15	brigg	the hungri

Table 5: List of the most indicative unigrams and bigrams (tokenized).