	Precision	Recall	F1
Test results:	0.561837456	0.501577287	0.53

GROUP: mdh323		**Max marks:**	**Total Marks**	**Comments**
1. Data analysis				
	1. Class label analysis	5	**3**	Well done for identifying class imbalance. Nice analysis of the distribution per keyword but it would have been nice to also see the distribution per country sentence length and their correlation.
	2. Qualitative assessment of dataset	10	**7**	Very good analysis of the subjective and context dependent nature of the tasks. It would be more indicative to use exact examples instead of paraphrasing them.
2. Modelling				
	1. Successful implementation	10	**10**	Successfully implemented a model that beats the baseline on the test set (0.49) and dev set (0.48). Very good description of the architecture. Well done!
	2. Choice of hyper-parameters	5	**4**	Good set of hyperparameters but it is not mentioned whether it was grid search or other search algorithm. It would also be good to investigate different schedulers.
	3. Further model improvements	10	**10**	Implemented several strategies including data augmentation, re-sampling and preprocessing.These are well explained and justified. Would have been interesting to see how backtranslation work for the data augmentation.
	4. Compare to two simple baselines	10	**10**	Successfully implemented two sensible baselines and provided performance scores. Good error analysis.
	5. Description of results	5	**5**	Good analysis of the different variations and comparison with the baselines.
3. Analysis				
	1. Analysis Q1	5	**5**	Good explanation with some quantitative analysis.
	2. Analysis Q2	5	**5**	Good explanation with quantitative analysis.
	3. Analysis Q3	5	**5**	Good explanation with quantitative analysis. An additional observation is the link between the keyword distribution and the model's performance, as the model is probably underfitting for thos categories.
4. Written Report				
	1. Introduction	5	**5**	Good abstract and concise introduction. The task and dataset are well described and the original work is cited.
	2. Readability	10	**10**	Good report but some information is scattered across sections (hyper parameters, future work,…)
	3. Good use of graphs or tables	10	**10**	Overall good use of graphs and tables and good captions.
	4. Conclusion	5	**4**	Good summary of results and work but missing some concrete future work ideas.
	Total	**100**	**93**	

70016 12

NLP Classification CW

Submitters

mdh323	Michael Hollins
ky523	Kangle Yuan
vm23	Vinayak Modi

NLP Coursework

Michael Hollins
mdh323@ic.ac.uk

Vinayak Modi
vm23@ic.ac.uk

Kangle Yuan
ky523@ic.ac.uk

Abstract

We deploy a Decoding-enhanced BERT (DeBERTa) model to perform binary classification aimed at identifying patronizing and condescending language (PCL) within the ”Don't Patronize Me!” dataset. Our methodology yields an F1 score of 0.58. However, this respectable headline performance exhibits significant variation depending on the length of paragraphs and the identification of vulnerable groups. This variability in performance underscores the empirical nature of natural language processing (NLP), highlighting that the success of specific enhancements and techniques is heavily dependent on the particularities of each task and its context.[1]

1 Introduction

This report summarises our attempt to develop a binary classification model to predict whether a text contains PCL. For our purposes, PCL is defined as language that expresses “a superior attitude towards others or depicts them in a compassionate way” Perez Almendros et al. (2020). Our corpus is the *Don't Patronize Me!* annotated dataset (available here).

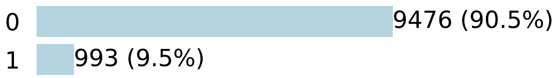
Our report is structured as follows: Section 2 provides a summary of the training data; Section 3 explores the implementation of our classification model; Section 4 discusses our model performance; Section 5 concludes.

2 Data Analysis

The *Don't Patronize Me!* dataset contains over 10,000 paragraphs extracted from English language news sources from 20 different countries. Each news paragraph covers a vulnerable community (e.g. the disabled, migrants, women) in which PCL is either present (1) or not (0). The presence of PCL is determined by a multiclass classification exercise from two expert annotators: each annotates a paragraph either 0 (no PCL), 1 (borderline PCL) or 2 (definite PCL). The annotators' scores are summed such that the combined score ranges from 0 to 4 (see Figure 3 in Appendix A). Finally, values of 0 and 1 are mapped to no PCL (0), and values 2, 3, and 4 map to PCL (1).

Figure 1 shows that over 90 percent of paragraphs in the official training dataset have no PCL, and so there is a clear class imbalance.

Figure 1: Distribution of binary PCL labels



Moreover, while the dataset is broadly balanced by vulnerable group with around 1,000 examples for each of the ten groups (see Table 1 in Perez Almendros et al. (2020)), the proportion of PCL classification per group varies significantly. Figure 2 charts how, for example, PCL is present in around twenty percent of paragraphs discussing poor families and those in need, but is very rarely found in paragraphs discussing migrants.[2]

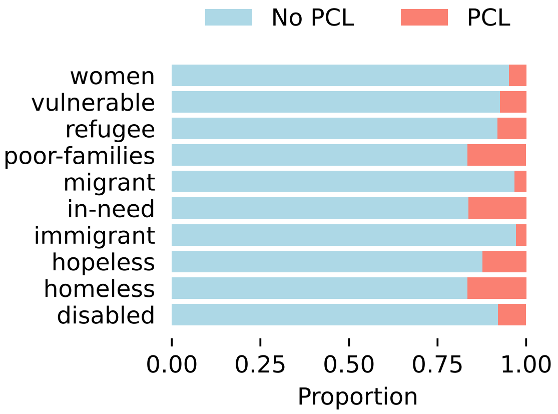
Class imbalance can be problematic in machine learning contexts due to the associated impacts on the model's ability to learn from the data effectively.[3] Issues may include bias towards the majority class, poor generalisation to the minority class, ineffective error signalling and misleading evaluation metrics. Efforts to tackle this imbalance are discussed in Section 3.2.2.

[1] For further details and access to our research framework, the code repository is available here.

[2] Fortunately however, PCL classification does not appear to differ significantly between longer and shorter paragraphs (see Figure 4 in Appendix A).

[3] For a survey of issues see (Abd Elrahman and Abraham, 2013).

Figure 2: PCL flags by paragraph keyword



In addition to inferential challenges from the imbalanced labels, it is important to highlight some of the key empirical challenges in using NLP to detect PCL.

First, the vocabulary employed in PCL can vary widely due to disparate *cultural and societal norms*. As the dataset is sourced from twenty different countries, what is considered PCL by members of one country could differ from those in another country. For example, one PCL passage (para ID:10324) from Nigeria discusses hopelessness in the context of the encounter between the prophet Elijah and God.[4] To a Nigerian audience, this episode might highlight the importance of material care alongside mental anguish with no PCL, but to another reader it might read as a simplistic and patronising solution to despair.

Second, the *intent* behind PCL is often benevolent and the use of PCL can be *unconscious*. This is something underlined by Perez Almendros et al. (2020) for why detecting PCL is much more challenging than other overt language, such as hate-speech. For example, paragraph ID 9850 speaks about the efforts Pope Francis has made to assist the poor and needy; despite these good intentions, the language used in the article still exhibits PCL.

Third, there are perceived *variations of degree* of PCL which are not easily captured by a neat binary label. As Perez Almendros et al. (2020) concede, the annotators don't always agree and sometimes a third annotator must referee these disagreements. To give a sense of the magnitude of the issue, the authors report that the annotators agreed in 9,182 paragraphs and disagreed in 1,457. Of these disagreements, 590 were total disagreements (0 vs 2) and 867 were borderline cases. Therefore we must acknowledge the extent to which the binary labels are reductive for the nuances of language.

3 Modelling

Our aim is to train a model to correctly classify PCL which outperforms the task's RoBERTa-base baseline model, which achieved an F1 score of 0.48 on the official dev-set and 0.49 on the official test-set.

3.1 Model description and motivation

We take our inspiration from the DeBERTa (Decoding-enhanced BERT with disentangled attention) from He et al. (2021), which the authors designed to improve on the performance of both BERT and RoBERTa. There are two innovations to model architecture that the authors highlight.

Disentangled attention mechanism: traditional models like BERT and RoBERTa treat all words equally for context, blending content and position importance. By contrast, DeBERTa separates (or disentangles) the attention mechanism into content and position, enabling a deeper understanding of words and their relationships.

Enhanced mask decoder: DeBERTa, like its predecessors, uses masked language modeling for pre-training, where certain words are hidden, and the model predicts them from the context. DeBERTa builds on this by factoring in both the relative position of words through its attention mechanism and their absolute positions before the softmax layer, enhancing its predictive accuracy.

To summarise, a model like BERT or RoBERTa tries to understand and predict words based on their joint content and relative position. By contrast, DeBERTa considers a word's content and relative position in their own right, as well as its absolute position in a sentence. This richer approach is especially promising in the context of PCL where the structure and content of language can be more subtle and implied.

To test our intuition that DeBERTa could be an improvement, we ran ten simple random experiments to compare the performance of DeBERTa against the baseline RoBERTa. The results, summarised in Figure 5, show that DeBERTa on average performs better. This motivates our adoption of this transformer architecture.

To begin with, we use 4 epochs and batch sizes of 32 (train) and 16 (evaluation). Our learning rate

[4] See 1 Kings 19.

is 2e-5 which is consistent with the implementation in He et al. (2021)[5] This gave us an **F1 score of 0.58**. We did not change our labels from the binary classification as we felt that the authors' design of deriving labels from multiclass classification was reasonable.

In our quest to improve on this, we implemented a cosine learning rate scheduler, anticipating improved training dynamics and model generalization. Theoretically, varying the learning rate could enhance performance by escaping local minima. However, despite these efforts, the results post-training and evaluation yielded an F1 score of 0.515 on our test dataset. This result prompted a return to simpler learning rate management for future experiments.

3.2 Further model improvements

Though we achieve improved performance compared to the baseline, it is clear that using more powerful models will likely yield better results. While the original RoBERTa model has around 120M number of parameters, our DeBERTa model has slightly more with over 140M parameters.[6] Therefore, to evidence meaningful improvement beyond this change in model architecture, we next turn to consider pre-processing, data sampling, and data augmentation.

3.2.1 Pre-processing

There are many pre-processing techniques in NLP (see for example the survey by Tabassum and Patil (2020)). These are sometimes used to help sharpen the model's focus on what the model will need to predict. We explore several strategies for pre-processing, including: punctuation normalization (A), strategic replacement of stop-words with a uniform token (B), and numeric token substitution (C).

A brief summary of our pre-processing results is that they improved performance relative to the RoBERTa base benchmark, but lowered performance relative to our DeBERTa model. Stop-word replacement (B) significantly reduced effectiveness, with F1 plummeting to 0.25, emphasizing the need for judicious stop-word filtering to avoid losing essential context.[7]

Table 1: Impact of Pre-processing Techniques on Model Performance

Pre-processing Technique	**F1-Score**
A	0.503
B	0.25
C	0.503
A + B	0.49
B + C	0.533
A + C	0.503
A + B + C	0.494

3.2.2 Data sampling

Earlier in Section 2 we showed how imbalanced the dataset is and touched on the potential issues that might arise as a consequence. In response to this, we use data sampling techniques in order to bring more balance to our labels with the goal of improving the predictive success of our model, trying both downsampling and upsampling.

Two balanced datasets are created: `balanced_df_down` through downsampling and `balanced_df_up` through upsampling, each followed by shuffling and index resetting for randomness.For the benefit of having more data points, we chose the upsampled dataset with 12,150 data points, evenly distributed across both classes, for subsequent training.

Performance fell from 0.58 to 0.51, potentially due to overfitting from minority class upsampling. Despite high accuracy (0.92), the low recall (0.42) indicates better negative class identification, showing a bias towards conservative positive class predictions.

Future approaches may include SMOTE for synthetic sample creation, enhancing class variability, and employing different models or regularization methods to curb overfitting and boost data generalization.

3.2.3 Data augmentation

Data augmentation aims to enhance training dataset size and diversity through modifications, seeking to improve model generalization, robustness, and reduce bias.

[5] We found that changes from this resulted in poor learning.

[6] Specifically, 124,647,170 and 139,193,858 parameters respectively.

[7] Our approach to refining the conventional stop-word list provided by the NLTK library was both unique and deliberate. We chose to retain pronouns (e.g., 'I', 'their', 'ourselves', 'we','they','us','them') and specific verbs (e.g., 'need', 'should', 'must'), hypothesizing their potential significance in conveying the author's stance or intentions, which could be pivotal in PCL classification. However, given the results, unfortunately our approach did not pay off.

Random Deletion dropped the F1 score to 0.483 by omitting words randomly. Random Insertion, adding synonyms, yielded an F1 of 0.43. Synonym Replacement also led to an F1 of 0.483, and Random Swap, changing word order, resulted in an F1 of 0.49.

Table 2: F1-Scores for Different Data Augmentation Techniques

Data Augmentation Technique	F1-Score
Random Deletion (Del)	0.483
Random Insertion (Ins)	0.43
Synonym Replacement (Syn)	0.483
Random Swap (Swap)	0.49

In another experiment, we integrated these four techniques, applying one at random to each piece of text. This led to a slight decrease in performance, with the F1-score dropping from 0.58 to 0.57 compared to the initial model. Despite this minor drop, the performance is significantly higher than when applying a single technique across the entire dataset as shown in Table 2. By merging these strategies, the augmentation not only doubles the dataset size but also significantly enhances its quality, laying a solid foundation for training more adaptable and accurate NLP models. This approach demonstrates that using a combination of techniques can improve on using any single technique alone.

In conclusion, we hypothesise that these techniques make our DeBERTa model perform worse because they could inhibit both the disentangled attention mechanism and the enhanced mask decoder.

3.2.4 Further hyperparameter tuning

Our choice of some of the hyperparameters has already been set out in Section 3. Here, we briefly cover our hyperparameter search process. Due to the time and computational expense, we focused on what we perceived as the key hyperparameters such as the learning rate, number of training epochs, batch size, and early stopping patience, which we varied across multiple trials to identify the most effective configuration. The learning rates considered were 2e-5, 3e-5, and 5e-5; the model was trained for 2, 3, 4, and 5 epochs in different runs; batch sizes of 16, 32, and 64 were tested; and the model was set to cease training after 3 epochs without improvement in evaluation loss.

These parameters were evaluated against several metrics, including the Matthews correlation coefficient (MCC), accuracy, F1 score, AUC-ROC, and AUPRC, to gauge their influence on the model's predictive capability. Our results indicated that the model achieved F1 scores around 0.49 for the validation set and 0.58 for the test set, suggesting a well-tuned model that performs robustly on unseen data. The optimal set of hyperparameters was found to be a learning rate of 2e-5, a training duration of 4 epochs, a batch size of 32, and an early stopping patience of 3, which was then chosen for the final model configuration.

3.3 Comparison to two baselines

We next measure our model against two simple baselines (BoW and TF-IDF):

3.3.1 Bag of Words (BoW)

The features utilized by the BoW model were unigrams.These features were further refined by capping them at 23,000 to avoid overfitting. In evaluation, the BoW model achieved a validation set F1 score of approximately 0.287 and an accuracy of 0.884, whereas for the test set, the F1 score was around 0.278 with a similar accuracy level.

By way of a misclassification example, a sentence portraying Syrian homelessness as non-patronizing reflects the model's limitations in contextually interpreting text.[8] It suggests an over-reliance on word frequencies, where nuanced terms such as 'millions' and 'homeless' may be underrepresented in training data for patronizing content. This incident highlights the critical need for models to incorporate full sentence contexts to accurately discern complex tones, a capability beyond the reach of frequency-based approaches like BoW.

3.3.2 Term Frequency-Inverse Document Frequency(TF-IDF)

Performance metrics from the validation set indicated an F1 score of approximately 0.302, an accuracy of about 0.865, a precision of 0.316, and a recall of 0.290. The test set results were slightly lower, with an F1 score around 0.275, demonstrating the challenges of achieving high precision and recall in this setting. Confusion matrices showed more false negatives, indicating the model, like the BoW baseline, better identified non-patronizing content but struggled with patronizing content.

[8] See relevant notebook in code repository.

A notable misclassification of a sentence describing Syrians' homelessness by the TF-IDF model as non-patronizing underscores its limitations in understanding context. This reflects the model's focus on word significance without fully grasping contextual nuances, illustrating a fundamental challenge in accurately identifying patronizing content through TF-IDF's lens.

The DeBERTa model surpasses traditional BoW and TF-IDF approaches by adeptly capturing linguistic nuances through its deep contextual comprehension, highlighting the effectiveness of advanced models in accurately classifying textual data beyond simple word counts.

4 Analysing model performance

To what extent is the model better at predicting examples with a higher level of patronising content? Though we have trained a binary classification model, it is interesting to consider how this performs using the multiclass categories discussed earlier in Section 2. We consider the model's F1 scores on paragraphs with high patronizing content (originally scoring 4), medium (3) and low(2).

Our model demonstrates a higher F1 performance with increased levels of patronizing content. While we achieve an F1 score of approximately 0.82 for highly patronizing content, this decreases for moderate levels to 0.68 and drops further for low levels of patronizing content (0.29).[9]

To summarise, the model is better at detecting higher levels of patronizing content. This characteristic could make the model particularly useful for flagging highly patronizing articles that may require further review by a human editor, which is a common practice in publishing houses. The lower performance in detecting low levels of patronizing content might be due to a variety of factors, including the subtlety of language cues at this level, which are inherently more challenging for the model to discern. Nevertheless, this mimics the challenges that humans face when categorising borderline cases of PCL.

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

The distribution of paragraph lengths, as indicated by a histogram, shows a significant right skew, leading to categorization of text lengths into very short (100-200 characters), short (200-500), medium (500-800), and long (¿ 800 chars).

Our model exhibits variable performance across different text lengths, summarised below and in Figure 7 in the Appendix A. The optimal F1 score of 0.72 for Medium texts suggests this length allows the model to utilize its attention mechanisms effectively without the confounding effects of extraneous content. Very Short texts are handled with a surprising degree of accuracy, as indicated by an F1 score of 0.62, pointing towards the model's proficiency with concise data. A moderate decline in accuracy is seen with Short texts (F1 score of 0.58), hinting at a lower limit of text length for effective prediction. The slight dip for Long texts with an F1 score of 0.67 may be due to an overload of context or noise in the data, suggesting there's a threshold for usefulness in additional text data.

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

The model's performance exhibits considerable variability across different data categories, which is likely indicative of how the language's intrinsic characteristics vary within each context (see Figure 8 in Appendix A). For instance, the model achieves a high F1 score of 0.61 in the disabled category, suggesting a strong alignment with the specific linguistic cues prevalent in this sector. By contrast, categories such as immigrant and women register notably lower F1 scores of 0.44 and 0.45 respectively, highlighting challenges the model faces in capturing the subtleties of language that characterize these contexts.

The variance in F1 scores across categories may stem from the differing proportions of PCL examples in the training data, with higher scores in categories like poor families and those in need, and lower scores for immigrants and migrants.

5 Conclusion

We successfully improved on the baseline RoBERTa performance by implementing a DeBERTa architecture. With carefully selected hyperparameters we achieve a meaningful F1 improvement to 0.58. Performance remains largely improved with the inclusion of other techniques, such as pre-processing, but these are lower than the headline achievement of 0.58. The evidence suggests that the class imbalance and subtle nature of PCL comprise significant hurdles to further model performance improvement.

[9] See Figure 6 in Appendix A.

References

Shaza M Abd Elrahman and Ajith Abraham. 2013. A review of class imbalance problem. *Journal of Network and Innovative Computing*, 1(2013):332–340.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Carla Perez Almendros, Luis Espinosa Anke, and Steven Schockaert. 2020. Don't patronize me! an annotated dataset with patronizing and condescending language towards vulnerable communities. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5891–5902, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ayisha Tabassum and Rajendra R Patil. 2020. A survey on text pre-processing & feature extraction techniques in natural language processing. *International Research Journal of Engineering and Technology (IRJET)*, 7(06):4864–4867.

A Appendices

Figure 3: Distribution of original labels

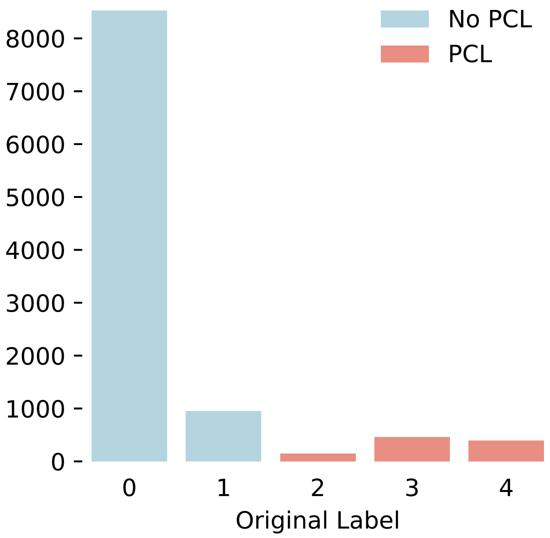
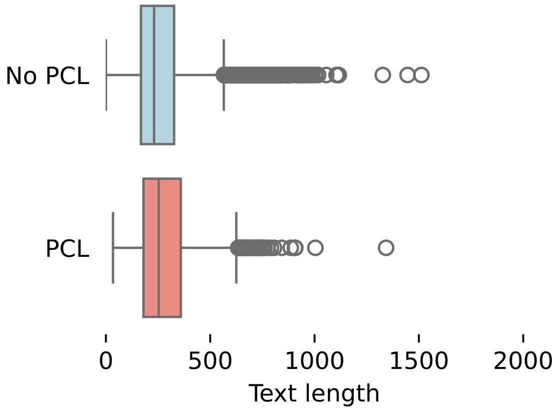


Figure 4: PCL distribution by length of text



For length less than 2000 characters

Figure 5: Model performance over 10 experiments

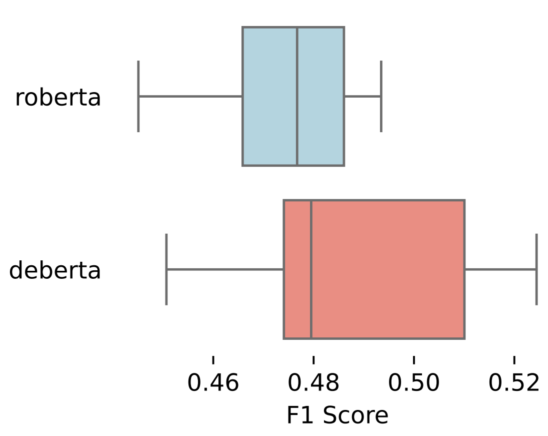
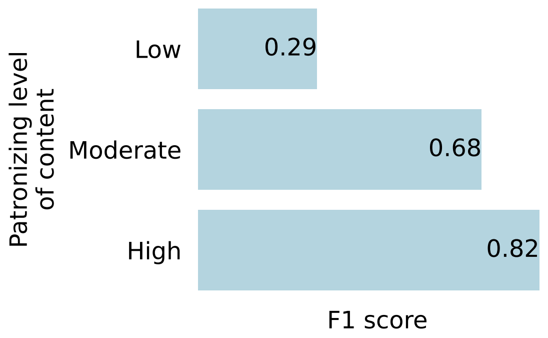
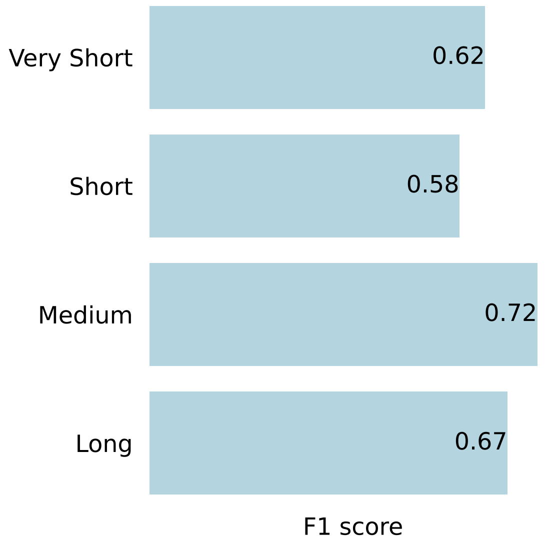


Figure 6: F1 Score by level of Patronising Content



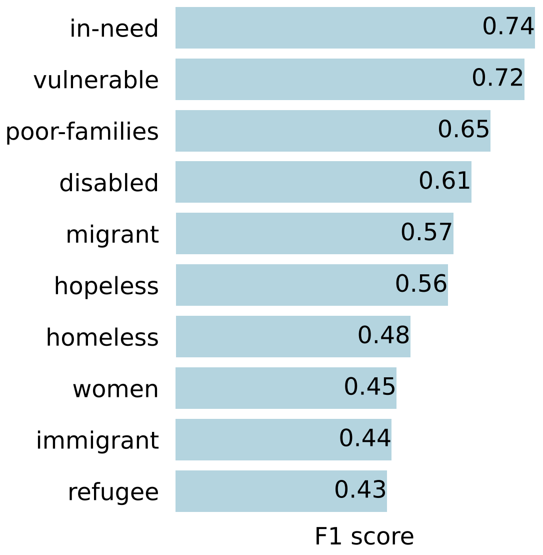
Categorised from Low, Moderate and High

Figure 7: F1 Score by Text length



Very short (100-200 characters), short (200-500), medium (500-800), long (over 800)

Figure 8: F1 score by vulnerable group



Categorised from Low, Moderate and High