Advancing PCL Detection with Custom BERT-base model

Mengyu Rao Kyveli Tsioli Angelos Ragkousis

1 Abstract

In this work, we designed a custom BERT-base model using data augmentation through back translation for the task of predicting Patronizing and Condescending Language (PCL). In comparison with the baseline approachs, our model achieves superior results on the *Don't Patronize Me!* dataset.

2 Introduction

Our goal is to predict whether a given paragraph belongs to Patronizing and Condescending Language (PCL) or not. Patronizing and Condescending Language (PCL) is observed when an entity uses language to portray others in a way that is compassionate or conveys a superior attitude towards them (Pérez-Almendros et al., 2020).

The identification of PCL is challenging for various reasons, such as the subtlety and context-dependency of connotations and the complexity of human expressions and intentions. An novel dataset for the task named *Don't Patronize Me!* was introduced by Perez-Almendros et al., (Pérez-Almendros et al., 2020). This dataset consists of 10,637 article paragraphs associated with potentially susceptible social groups in order to identify instances of PCL. The current baseline for the task is a RoBERTa-base model that recorded an F1 score of 0.48 on the official dev-set. In this work, we effectively develop a custom BERT-base model that achieves a higher F1 score of 0.55.

3 Data Analysis

3.1 Statistics

According to the original paper (Pérez-Almendros et al., 2020), two expert annotators were asked to classify the sentences of the dataset using the levels of 0 (No PCL), 1 (Borderline PCL) and 2 (Overt PCL). Their results were then combined into a 5-point scale, where *level 0* means that both annotators assigned the level 0 (0 + 0), *level 1* means that one annotator assigned the level 0 and the other assigned the level 1 (0 + 1), and so on up to *level 4*, for which both annotators assigned the level 2. For our binary classification task, levels 0 and 1 jointly represent the negative class, whereas levels 2, 3 and 4 represent the positive class.

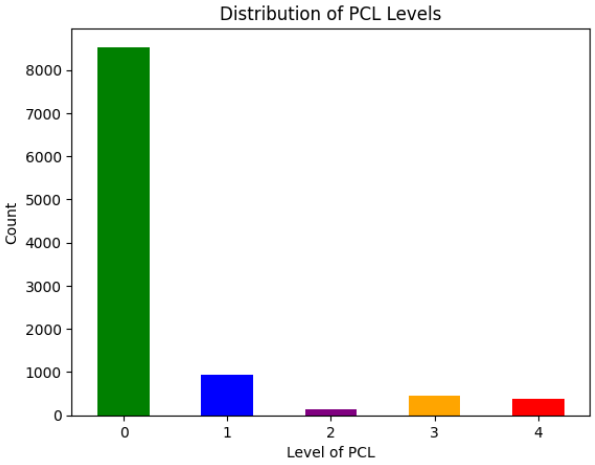

Figure 1: Distribution of PCL levels.

Figure 1 shows the distribution of PCL sentences based on these levels of severity. First, it is evident that the task presents great class imbalance, with approximately 9000 No PCL sentences (levels 0,1) and just 1000 PCL sentences (levels 2,3,4). Along these 1000 positives, only 390 sentences are classified as clearly showing PCL (level 4), which makes the prediction even more difficult.

Additional plots are given in the Appendix. In Figure 2, we present the distribution of class labels across different countries. We see that the sentences are spread uniformly, with around 20-80 positive and 400-500 negative examples for all countries. In Figure 3, we also compare the distribution of sentence lengths per PCL type. The positive and the negative class have very similar average sentence lengths (around 50 words), and the majority of sentences are restricted to 100 words.

However, the positive class presents many more outliers (more than 150 words), which is expected due to the much larger number of examples. Finally, an important analysis that of Figure 4, which shows the distribution of examples across different communities. The figure shows a slight imbalance in the number PCL texts. *Women* and *immigrants* are associated with less PCL sentences compared to *homeless* and *in-need* people. We infer that the media may be biased towards certain groups.

3.2 Task Assessment

Detecting a patronizing attitude in text is inherently challenging. According to the original paper (Pérez-Almendros et al., 2020), the two expert annotators disagreed in 1457 cases, which shows the difficulty of the task. Observing this attitude can be very subjective and is often expressed in indirect and subtle ways. In this section we analyse two factors and provide each with an example.

First, the most common case of confusion probably arises when the given sentence describes an extreme and harsh situation that leaves the reader with a feeling of sadness and helplessness, while not necessarily being condescending. A relevant example found in the dataset is the following: **"These shocking failures will continue to happen unless the Government tackles the heart of the problem - the chronic underfunding of social care which is piling excruciating pressure on the NHS, leaving vulnerable patients without a lifeline"**. This sentence is not classified as PCL because it simply presents serious societal issues in an objective manner. However, the excess of adjectives and flowery wording, e.g. *shocking failures* and *excruciating pressure* can easily confuse the reader and our classification model.

Moreover, another challenging factor is the difficulty to understand the true intention of a person by a single written phrase, without additional context such as his body language or intonation in a live speech. A relevant example of Level 1 that was marked differently by the two annotators is the following phrase: **"NUEVA ERA , Ilocos Norte - No family shall be homeless under the watch of the municipal government here , said town Mayor Aldrin Garvida ."**. This statement could be perceived by some readers as a bold promise of a town mayor who aims to solve a major problem that a vulnerable community faces, presenting himself as a saviour. However, it can also be perceived as a political statement that simply introduces his future plans to tackle an important issue, without a condescending attitude.

Additionally, there are more reasons that make the task subjective, such as different interpretations based on cultural norms and societal beliefs.

4 Modelling

4.1 Model Description

We implemented a custom model based on BERT. BERT's transformer architecture is very capable of capturing contextual relationships within text, which is crucial for tasks such as PCL detection where not only the literal meaning of words, but most importantly, their connotations and are important (Devlin et al., 2018). Also, BERT's pretrained knowledge serves as a great foundation for further finetuning on the task, and its attention mechanism can effectively handle long-range dependencies. This is crucial for PCL detection since often words in isolation might seem benign, while they contribute to a patronizing tone when considered in the full context of a sentence.

Given the fact that capitalisation in language can alter the tone or emphasis of a particular word or phrase, thereby changing its connotation, we decided to implement a cased version of BERT in order to preserve the linguistic cues and tone in the text. We also use the BERT tokenizer for preprocessing text (operations include: tokenization, addition of special tokens, padding, truncation to a maximum length, creation of attention masks for differentiation between actual tokens and padding for the model). Our custom BERT model builds on top of the Hugging Face pre-trained model, which we adapted to our task. To this end, we used BERT to extract embeddings and we added a linear layer to classify the samples into the PCL and no PCL classes. The projection layer takes the high-dimensional representations learnt by the BERT model and projects them down to a space suitable for a binary classification problem.

We use the following hyper-parameters: learning rate: 0.00001, batch size: 32, number of epochs: 4 (due to computational resources constraints). To perform hyperparameter tuning, we split our training set into training and validation sets with 80-20 ratio and we treated the official dev set as our own internal test set. Last but not least, we are training the model with the default Hugging Face optimizer, AdamW, and we leverage the benefit of decoupling weight decay to introduce

regularisation and prevent overfitting. This is the reason why weight decay is not explicitly set in our training arguments. Please refer to Table 1 for a comparison of our customised BERT versions. These versions are described in the next section.

4.2 Model Improvements

4.2.1 Data Augmentation

After experimenting with classic approaches to data augmentation such as synonym/antonym replacement, we designed a more sophisticated data augmentation method through back translation. More specifically, we performed back translation from English to French and then back to English. This method inherently preserves the semantic content while introducing syntactic and linguistic variability, which is crucial for recognising nuanced and subtle expressions of condescension. In addition, back translation can generate more naturally varied sentences than simple synonym or antonym replacement, providing a more rich set of training text examples that better "mimic" the variability and diversity found in real-world language. It is worth mentioning that given the class imbalance in the training set, we decided to perform data augmentation via back-translation only to the training samples of the PCL class. This is backed up by the need to achieve better f1 score on the PCL class, which is the class of interest and for which the model is less able to perform well. It should be highlighted that when performing our data augmentation method, we remove the downsampling of the majority class step which was implemented in the legacy code. This is because we want to avoid "double skewing" of the training data distribution, which could introduce bias to the training set.

An example of an original training data sentence and its respective augmented version generated through the back-translation is provided:
Original text: *"ASWS CEO Di Gipey said the Coroner 's report into the deaths of two women after long histories of domestic violence* ***again*** *highlighted the* ***pressing*** *need to make real changes that will make women and children safe."*
Generated text: "ASWS CEO Di Gipey said the coroner's report on the death of two women after many years of domestic violence, **once again** stressed the **urgent** need for real changes that will make women and children safe."

4.2.2 Further Experimentation and Engineering Pipeline

As a first step in the engineering experimentation pipeline, we freezed the weights of BERT in order to use the pre-trained model's parameters and then we evaluated the performance of the model on our dev set. This approach didn't boost the generalisation performance of the model on our dev set, which lead to our decision to adjust the pre-learnt word embeddings and weights during training, effectively performing fine-tuning on our task.

As far as the network architecture is concerned, we experimented with two versions of the projection layer, namely a single linear layer and a more complex set of linear layers. We decided to experiment and explore these two variants of the projection layer to assess whether the model benefits from a gradual, sequential transformation of the features before the final classification. Our experiments with both customised BERT versions revealed signs of overfitting on the training set, for which reason we introduced dropout in the projection layer, in our efforts to boost the generalisation ability of our model.

As a next step, we trained both model architectures with weighted Cross Entropy Loss, taking into account the class imbalance problem. We trained the model with weighting based on the inverse proportion of the classes in order to put more importance on the minority class (i.e. PCL class), thus penalising misclassifications of this class more than those of the majority class (i.e. non PCL). After conducting several experiments with varying hyperparameter settings, we concluded that the model was still failing to classify the PCL instances, leading to a poor generalisation performance on the dev set reflected on the f1 score of the PCL class which was in the range of [0.45, 0.46]. This lead us to more drastic changes in the model setup, especially in the context of exploring different data augmentation strategies, which were described in detail in section 3.2.1.

4.2.3 Final Experiments

Thus, we conducted our final experiments using a grid search strategy that combines different levels of data augmentation with all the different network architectures described above. All our final experiments are summarized in Table 1. To this end, we explored how augmenting the original training set by 10%, 20% and 30% respectively affects the performance. To augment the PCL class,

we randomly sampled PCL class texts which were then augmented by back-translation. We tested the two architectural variants, namely the single layer and the three layer projection head, and explored dropout rates of 0.1 and 0.2 respectively.

Our analysis revealed that the model that yielded the best f1 score on the PCL class on the dev set was achieved with 30% augmentation and with the single linear layer architecture component. One possible explanation for this is that adding more layers increases then model's capacity, which can lead to overfitting. Indeed, in Table 1, we can see a considerable discrepancy between the train set f1 and dev set f1 scores. In addition, a single linear layer can effectively map the rich contextual embeddings provided by BERT, which might indicate that more layers are not necessarily leading to improved performance.

Once we concluded on our final configuration, we performed hyper-parameter tuning, exploring the batch size and the learning rate. The results are shown in Table 2. After experimenting with different learning rate values, we found out that having a small learning rate of 0.00001 improves stability during fine-tuning. Since we are not freezing the weights of the pre-trained BERT, we mostly experimented with small learning rate values to prevent "catastrophic forgetting" (losing the previously learnt knowledge), by allowing the model to adjust its weights gradually. Also, the smaller batch size of 32 proved more effective than that of 64, since smaller batches often lead to better generalization on new data and assist in regularisation.

Table 1: Summary of F1 Scores for PCL with Various Data Augmentation and MLP Configurations

Aug. (%)	Configuration	Train F1	Dev F1
10%	One linear layer	0.870	0.523
10%	MLP, dropout p=0.1	0.784	0.536
10%	MLP, dropout p=0.2	0.621	0.504
20%	One linear layer	0.906	0.527
20%	MLP, dropout p=0.1	0.907	0.528
20%	MLP, dropout p=0.2	0.718	0.542
30%	**One linear layer**	**0.887**	**0.552**
30%	MLP, dropout p=0.1	0.817	0.549
30%	MLP, dropout p=0.2	0.745	0.523

Table 2: Hyperparameter Tuning

Configuration	Train F1	Dev F1
($batch = 32, \quad lr = 1 * 10^{-5}$)	**0.887**	**0.552**
($batch = 64, \quad lr = 1 * 10^{-5}$)	0.833	0.532
($batch = 32, \quad lr = 2 * 10^{-5}$)	0.845	0.541
($batch = 64, \quad lr = 2 * 10^{-5}$)	0.828	0.527
($batch = 32, \quad lr = 5 * 10^{-5}$)	0.780	0.529
($batch = 64, \quad lr = 5 * 10^{-5}$)	0.764	0.515

4.3 Comparison with Baselines

We implemented Support Vector Machine (SVM) and Bag of Words (BoW) as baselines for comparison with our model. The following section introduces the basic principles of the two baselines and compares their performance. We also highlight an example of miss-classification.

The SVM is based on the principle of finding a hyperplane that optimally separates two categories in a dataset. To optimally separate these two categories, the goal of SVM is to maximize the margin between the two categories, which is the distance of the support vectors to the separating plane. For the Bag of Words (BoW) model, we implemented the Unigram version. Unigram model means that the text is treated as a collection of words, ignoring the order between them, and each word is treated as an independent feature.

As shown in Table 3, the SVM model has an F1 score of 0.413 on the training set and 0.420 on the development set, while the BoW model has an F1 score of 0.379 on the training set and 0.343 on the development set. Compared to these two baseline models and the original RoBERTa baseline, our model shows great improvement in performance on the training set, and also an increased ability to generalise to new data in the validation set.

Table 3: Comparison with baseline methods

Method	Train F1	Dev F1
RoBERTa (baseline)	0.490	0.480
Support Vector Machine (SVM)	0.413	0.420
Bag of Words (BoW)	0.379	0.343
Model (ours)	**0.887**	**0.552**

An example of miss-classification by our BoW model is the following: *The memo said that refugee processing centres abroad would not be able to request new SAOs for refugees until there*

was further guidance from government. The real label of this sentence is 0, while the model misclassifies it as label 1. Some possible reasons are: a) The sentence contains a complex structure and the conditional expression: *would not be able to request new SAOs for refugees until there was further guidance from government*. Its meaning and intonation are highly dependent on the order and context of the words, so that the model cannot accurately capture this complexity and subtle semantic differences. b) The sentence contains domain-specific terms or acronyms (e.g., SAOs) which may have fewer occurrences in the training data, and the model may not be able to fully comprehend their significance or contextual relevance.

In contrast, our transformer model addresses these challenges and captures contextual relationships due to its attention mechanism and our data enrichment after back-translate augmentation.

5 Analysis

Is the model better at predicting examples with a higher level of patronising content?

The predicted F1 score per level of patronising content is shown in Figure 5. The results are calculated on the official validation dataset (2093 sentences). From the results it is clear that the F1 score increases for higher levels of patronising content, which is expected. The score is around 0.8 for level 4 sentences, 0.6 for level 3 sentences, and drastically drops to 0.35 for level 2 sentences. Just like the human annotators, our model can more easily distinguish overt PCL sentences of level 4 rather than sentences with ambiguous meanings in level 2.

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

The predicted F1 score per text length is shown in Figure 6 in the Appendix. The maximum sentence length of the official validation dataset is 272 and the average sentence length is 48. Thus, to facilitate the analysis, we split the sentences into 5 bins based on their length, such that all bins contain over 100 sentences. The figure shows that in all bins, the f1 score is in the same range, between 50-60. This shows that the text length does not seem to affect our predictions. This could be expected because the essence of patronizing language lies primarily in its choice of words and context rather than the sheer amount of text. A short sentence can be just as patronizing as a longer one if it implies superiority or condescension. Also, the architecture of transformers, based on multi-head attention, explains why the predictions do not rely too much on sentence length. BERT's architecture allows it to consider the entire context of a sentence, which includes the semantics, the interplay between words, and the potential connotations of phrases, rather than relying on surface-level features like sentence length.

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

The predicted F1 score per community is shown in Figure 7 in the Appendix. It is clear that most community types show comparable F1 scores, in the range of 0.4-0.5, with the exception of 2 communities. First, predicting PCL for those *in-need* shows a much larger F1 score of around 0.7. Looking again at the distribution in the training set of Figure 4, we observe that this category also has the largest number of PCL examples, which may have assisted our model to find more subtle meanings for this category and generalise better in the validation set. On the opposite side, we observe a poor F1 score of just 0.10 for the community of *women*, which also has the fewest training PCL examples in Figure 4. Thus, the model did not have enough information to properly associate PCL attitude towards this group.

6 Conclusion

Compared with the baselines, our model increased the F1 score on the training set to 0.887 and the F1 score on the dev-set to 0.552. We do that by augmenting our dataset using back-translation and finetuning a BERT-base model with a projection layer. We find that the model's ability to predict patronizing content improves with higher levels of such content, achieving an F1 score of 0.8 for the most overt cases. Moreover, the length of the input sequence does not significantly impact model performance, and the performance varies across community categories, with notably higher accuracy in detecting PCL towards communities having more training examples, such as those in-need.

We note that incorporating the level of PCL of each sentence in the weighted loss function can be a future work. Forcing the model to assign larger weights to level 4 PCL sentences than lower levels could further tackle class imbalance.

References

[Devlin et al.2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. October.

[Pérez-Almendros et al.2020] 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.

Appendix A Additional Plots

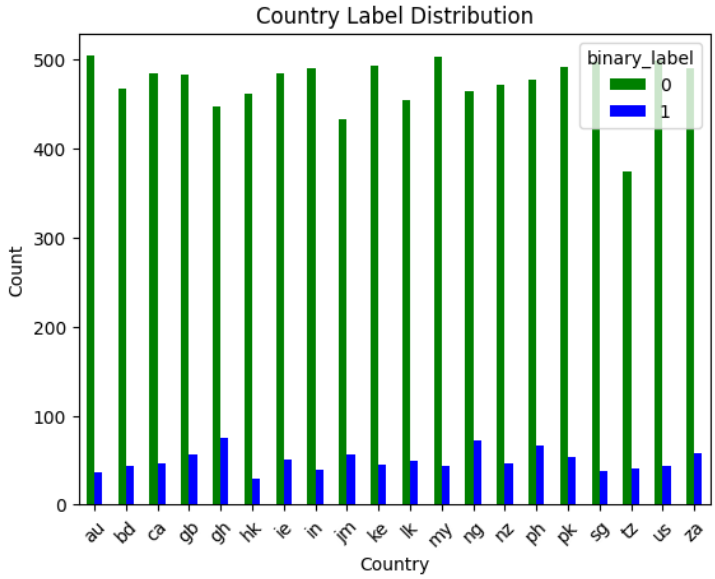

Figure 2: Distribution of PCL (1) and No PCL (0) texts per country.

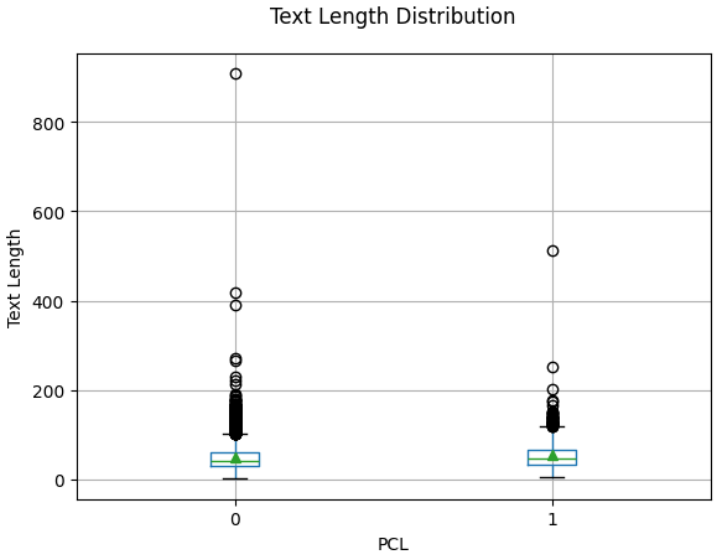

Figure 3: Distribution of PCL (1) and No PCL (0) texts per text length.

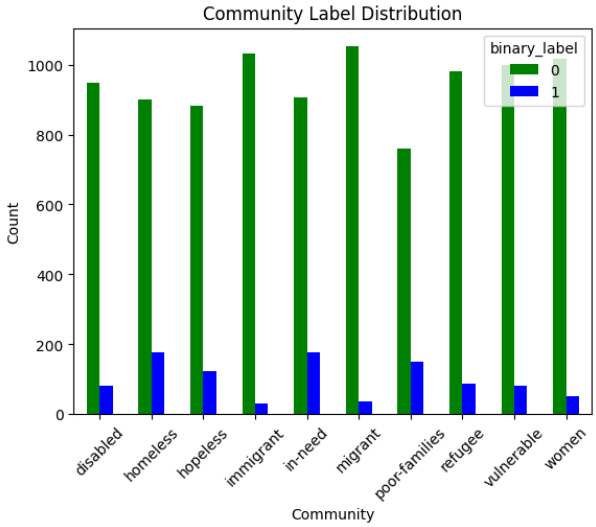

Figure 4: Distribution of PCL (1) and No PCL (0) texts per community type.

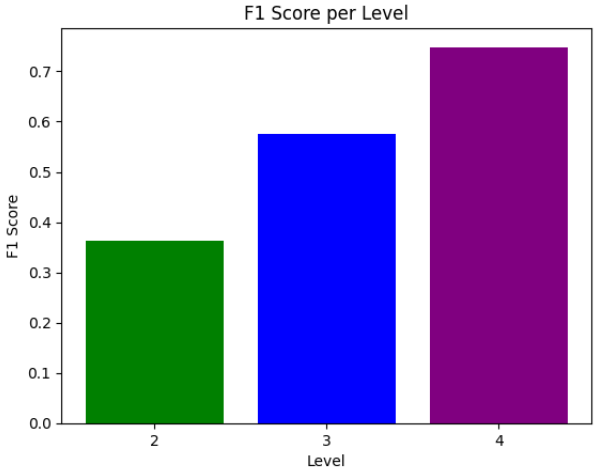

Figure 5: F1 score per level of PCL.

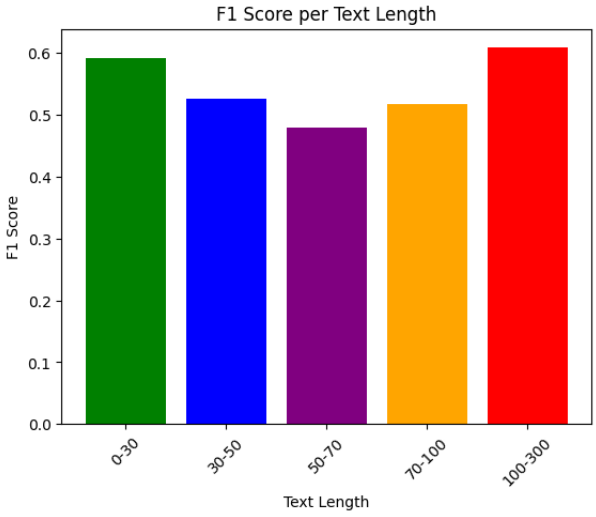

Figure 6: F1 score per text length.

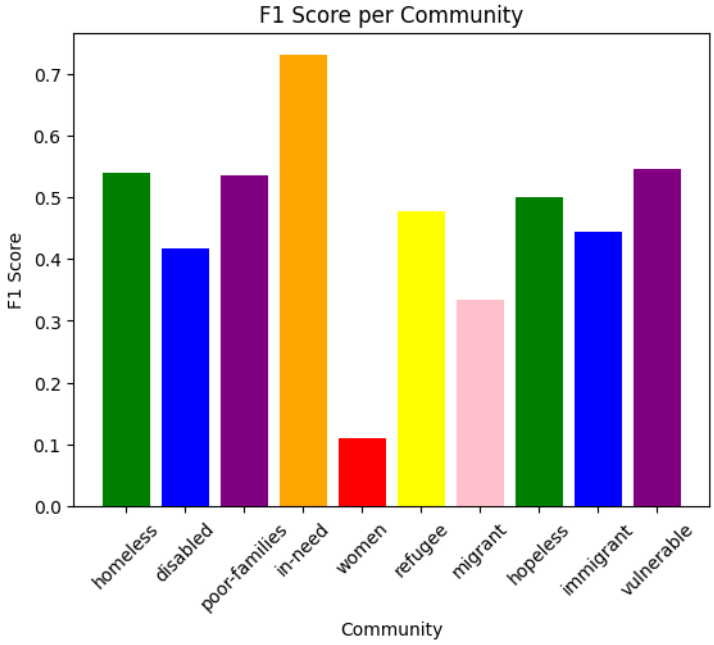

Figure 7: F1 score per community type.