# NLP Coursework

**Authors**
Quanlong Li
Charlize Yang
Xuanjia Zhang

## 1 Introduction

This report uses the *Don't Patronize Me!* dataset [1] to identify Patronizing and Condescending Language (PCL) in news stories regarding vulnerable communities. The task is implement a HuggingFace transformer model to perform binary classification on paragraphs (label 0: no PCL; label 1: with PCL) and achieve a F1 score above 0.48.

## 2 Data Analysis

### 2.1 Analysis of the class labels

The dataset consists of 10468 paragraphs, in which 993 paragraphs (9.49%) contain PCL (with label 1), based on the evaluations of two annotators. To make predictions, further data processing techniques (Section 3) are required to rebuild this highly unbalanced dataset.

| | 0 | 1 |
|---|---|---|
| Number of class labels | 9475 | 993 |
| Percentage | 90.51% | 9.49% |

The class labels correlate with data features, such as word lengths of paragraphs, the vulnerable groups and the news article's origin country. Figure 1 shows that PCL tends to be more visible in longer paragraphs. A similar analysis suggests that PCL is more visible to groups of the homeless, those in need, and poor families. For countries, Ghana has the highest probability of PCL (14.35%), while the proportion of PCL usage from Hong Kong is only 5.04%.

### 2.2 Qualitative assessment of the dataset

The original dataset contains labels 0 to 4 with increasing levels of PCL based on the partial disagreement of two annotators' evaluations. The task of identifying PCL content in a paragraph is highly subjective. For example, among all 1457 partial disagreements between two annotators (labels 1 and 3), they have completely opposite opinions on 590 of them (no PCL vs with PCL).

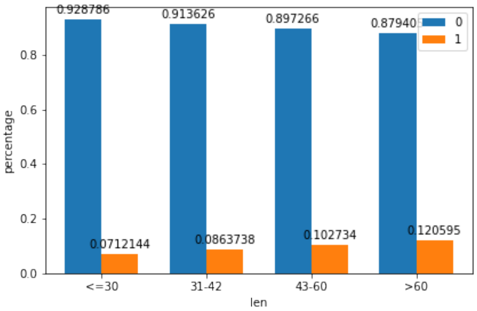

Figure 1: Percentage of class labels with input length

As a further example, both sentences in the table below have label 1, i.e. two main annotators have different opinions on its PCL level. In the first sentence, "should get along well with" and "gainful employment" indicate the author's attitude that the immigrant community should settle themselves well, which seems to be slightly patronising. In contrast, the second sentence is fact-stating and does not contain any PCL. This example shows the subjectivity and potential inaccuracies in the annotator's judgements.

| ID | Text | Label |
|---|---|---|
| 2881 | According to their findings, while Singaporeans do not expect new immigrants to give up their own culture for Singapore's, they feel that immigrants should get along well with their neighbours and colleagues, as well as find gainful employment. | 1 |
| 6733 | From Ghana to Greyhound : One immigrant 's story of getting by in NY. | 1 |

## 3 Data Processing

We experimented with different data processing methods to correct the skewed class proportions in data. Their performances were evaluated by the F1 score on the positive class, using 5-fold cross-validation (CV) on the whole dataset. We only applied data processing to the training sets in CV to avoid enclosing training data information to the validation data.

We tried four data processing techniques: data augmentation, balancing class weights, data downsampling and upsampling, each with a brief explanation below:

**Data augmentation (A)**: For data with label 1 (minority class), we used WordNet (in NLTK library) to replace two words per sentence with synonyms randomly. A new sentence was generated for each original sentence, thus doubling the minority class data size. This setup aims to preserve the patronising level per sentence and maximise the diversity of augmented data.

**Balance class weights (B)**: we leveraged the class weights parameter during the fit model process to give equal attention to both classes in the output. As a result, more loss value was associated with the minority class during the backpropagation.

**Data downsampling (D)**: we downsampled the label 0 data (majority class) by randomly removing records from that category until it doubles the size for the minority class.

**Data Upsampling (U)**: we upsampled the label 1 data (minority class) by randomly copying records from that category until its size doubles.

Results: The combination of augmentation and downsampling is the most effective under the RoBERTa-base baseline model, reaching an average F1 of 0.68 in CV. Adding balanced class weights to this combination actually decreased the F1 score and the precision, as we penalised more on the misclassification made by the minority class with a higher class weight.

| A | B | D | U | Average F1 (Internal CV) |
|---|---|---|---|---|
| - | - | ✓ | - | 0.52 |
| - | - | - | ✓ | 0.49 |
| ✓ | - | ✓ | - | 0.68 |
| ✓ | - | - | ✓ | 0.67 |
| ✓ | ✓ | ✓ | - | 0.51 |
| ✓ | ✓ | - | ✓ | 0.53 |

## 4 Modelling

### 4.1 Hyperparameter tuning

We down-sampled the whole dataset (10637 data points) for hyperparameter tuning. For each model, we trained a maximum of ten epochs and used early stopping with two epochs of patience to reduce overfitting. The best parameters were chosen iteratively based on the averaged F1 score in a 3-fold CV.

#### 4.1.1 Search for best architecture

We experimented with three transformer models in HuggingFace with the following initial setting: *lr*: 1e-4, *categorical*: True, *tokenizer_max_len*: 256, *gradient_clip_val*: 0.1, *target_label*: original, *batch_size*: 32, *dropout*: 0.1.

| Architecture | Average F1 |
|---|---|
| distilbert-base-uncased | 0.256 |
| distilbert-base-cased | 0.160 |
| roberta-base | 0.019 |

Best model: distilbert-base-uncased. This relatively small model is faster to train and less prone to diverge. The small training set size may cause the poor performance of the roberta-base model in CV (as shown in the table). The uncased version of the distilbert model performed much better than the cased model. Thus we used **distilbert-base-uncased** model in all later experiments.

#### 4.1.2 Original label vs Binary label

| Target label | Average F1 |
|---|---|
| Binary label | 0.403 |
| Original label | 0.344 |

Trying to predict the original labels 0-4 first and then classify them as positive or negative examples gave a worse result than predicting the binary label directly.

#### 4.1.3 Add categorical features

| Add categorical | Average F1 |
|---|---|
| False | 0.403 |
| True | 0.403 |

We concatenated the one-hot encoded categorical features ”country” and ”keyword” to the output of the encoder layer, then passed the whole vector to a final linear layer for classification. The result shows that adding categorical features do not influence model performance. So we excluded categorical features in later experiments to simplify the model.

#### 4.1.4 Other parameters

We found that adding dropout in the final classification layer or changing gradient clip values did not significantly affect model performance. In addition, we tuned the learning rate, batch size, max token length to optimise model performance. The full hyperparameter tuning results are available in the GitLab repository (https://gitlab.doc.ic.ac.uk/ql5318/nlp_cw), with the final model results presented in Section 5.

## 5 Final results

Our final model has *architecture*: distilbert-base-uncased, *lr*: 1e-4, *target_label*: binary, *categorical*: False, *tokenizer_max_len*: 128, *batch_size*: 32, *gradient_clip_val*: 100, *dropout*: 0.3. The final model was trained on the augmented and downsampled **official training set** to obtain the best results.

### 5.1 Single model

The model performance on the **official dev set** is:

| Precision | Recall | F1 |
|---|---|---|
| 0.46 | 0.58 | 0.51 |

The model performance on the **official test set** is:

| Precision | Recall | F1 |
|---|---|---|
| 0.42 | 0.61 | 0.50 |

### 5.2 Ensemble model

We also tried to ensemble the predictions from the three models in Section 4.1.1 by majority voting. The performance of the ensemble model on the **official test set** is given below. As we observe, the ensemble model outperformed any single model in terms of F1 score and precision, showing its capability to capture more minority class labels. This is our best-performing model.

| Precision | Recall | F1 |
|---|---|---|
| 0.49 | 0.58 | 0.53 |

## 6 Analysis of model

### 6.1 Influence of patronising level

| Label | Correct classif. | Wrong classif. | Acc |
|---|---|---|---|
| 0 | 1610 | 94 | 0.945 |
| 1 | 147 | 44 | 0.770 |
| 2 | 4 | 14 | 0.222 |
| 3 | 46 | 43 | 0.517 |
| 4 | 66 | 26 | 0.717 |

The table above shows our best single model performances in paragraphs with different patronising levels (with original labels 0-4). Label 2 instances, classified by both annotators as borderline cases, were frequently misclassified as negative examples. On the other hand, the clear-cut cases with labels 0 and 4 are rarely misclassified. Overall, the predictive performance of our model is much lower for the positive examples than the negative examples (accuracy: 0.583 vs 0.927), as we have a highly imbalanced dataset.

### 6.2 Influence of input paragraph length

| Word length | Precision | Recall | F1 |
|---|---|---|---|
| ≤ 30 | 0.4000 | 0.7568 | 0.5233 |
| 31-42 | 0.4918 | 0.5769 | 0.5310 |
| 43-60 | 0.4815 | 0.6047 | 0.5361 |
| >60 | 0.4637 | 0.4776 | 0.4705 |

In general, the F1 score and precision increase with input length up to 60 words and then drop. The potential reason is: for moderately long sentences, we could extract more information about PCL as it gets longer. But when the paragraph gets too long, the prediction quality may drop due to too many noisy and unclear expressions.

## 7 Conclusion

To summarise, we adopted data augmentation and downsampling to balance the unequal classes. Our best model ensembled three transformer models and achieved an F1 score of 0.53 on the official test set. The analysis showed that the model better identifies examples with either no PCL or very strong PCL, and it best predicts moderately long paragraphs (30-60 words). As next steps, we could experiment with alternative data upsampling methods such as SMOTE and back-translation, implement a learning rate scheduler, adapt to a weighted sum ensemble model and consider grouping countries by continent to provide more information!

## References

[1] Carla Perez-Almendros, Luis Espinosa-Anke, and Steven Schockaert. 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, 2020.