NLP Coursework

Patrick LEGGETT
MSc Artificial Intelligence
pfl21@ic.ac.uk

Benjamin LEROY
MSc Advanced Computing
bcl21@ic.ac.uk

Tristan OUALID
MSc Advanced Computing
to521@ic.ac.uk

Introduction

The task described herein is to develop a binary classification model to predict whether a text contains patronising or condescending language (PCL). We describe our analysis of the training data, and our improvements to the task's baseline mode (RoBERTa) as well as including analysis of our model based on attributes of the training data.

The *Don't Patronize Me* dataset was introduced by *Pérez-Almendros et al.* and is aimed at supporting development of NLP models to identify language as patronising or condescending to vulnerable communities. The link to our Colab notebook is here.

1 Data analysis of the training data

The dataset is annotated on a five-point scale (0-4) with 0 indicating the language does not contain PCL and 4 indicating it clearly contains PCL (based on scores from two annotators). These are binarised so 0 and 1 maps to 0 (not containing PCL) and all others to 1.

1.1 Quantitative Analysis

After binarisation, the training data contains 9.5% positive labels (paragraphs containing PCL) and 90% negative (not containing PCL).

We analysed the average lengths of the two classes in characters and found positive labels to have a higher mean length (252 characters as regards to 232).

1.2 Qualitative Analysis

Not only is the labelling of this dataset subjective, but the definition of PCL in the original paper contains 5 different traits, and explicitly rules out messages which are openly offensive as PCL. This makes classification on the dataset a difficult task.

As an example of this, one paragraph in the dataset reads as follows:

> 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.

One of the traits cited as an indicator of PCL is ”language [which] raises a feeling of pity towards the vulnerable community”, and one could strongly argue that this passage raises a feeling of pity towards the vulnerable community (vulnerable patients), yet the label in the dataset is of not containing PCL. We would concur with the original authors that this task is ”subtle and subjective”.

As described by Wang and Potts (2019), one characteristic of condescending language is that condescension is highly based on the social roles of the participants (they use an example where something said by one friend to another may be perceived as highly condescending when said to a stranger but not in the former case). This combines with the subjective role of annotators in deciding whether language in a paragraph is condescending or not.

These factors combine to make this task both subjective and difficult.

2 Modelling

After having analyzed the data, we worked on how to outperform the RoBERTa-base model for this specific task. As for any machine learning task, we had to work on 2 specific points to create our model: the pre-processing of the data and the creation of a tuned model.

2.1 Pre-processing

2.1.1 Dealing with imbalanced class using sampling and data augmentation

The dataset is imbalanced, however to make sure that the model performs well, it has to be trained equally on inputs labelled 0 and 1.

The first idea was to down-sample the majority class. This makes the dataset smaller than the initial one but ensuring both classes are equally represented. As expected, this idea reduces the performance of the model: with a smaller dataset, the model doesn't generalize as well. A more intuitive idea was to increase the size of the dataset using up-sampling. Having a bigger dataset makes the model more robust to new inputs. Using an existing method for sampling with replacement from the minority class, this method didn't really improve the performances. Even if the model is trained better on the minority class, it doesn't make it better to new inputs which contains other words. We also tried the SMOTE method for up-sampling but we had several memory issues preventing from fully utilising these ideas.

Having tried different sampling methods with the dataset, we focused on data augmentation for the minority class. We tried to generate new samples using Wordnet to create synonyms of the initial words and then new sentences to add to the dataset. The number of synonyms generated compared to the length of the initial sentence was a hyper-parameter to tune. However, this method didn't improve the F1-score of our model.

2.1.2 Tokenization

The second part of the pre-processing in NLP is modifying the inputs to deal with the myriad representations of words and to create an input usable by our model. We used the class transformers to implement this tokenization because it contains some pretrained tokenizer for each Hugging face model available. Therefore, we used the distilbert-base-uncased as model type to initialize this tokenizer.

2.2 Implementation of a HuggingFace transformer model

2.2.1 Model choice

To perform our classification, we wanted a model that was using the whole paragraph to detect if an input contains or not patronising or condescending language. Furthermore, due to our limitations in GPU usage, we wanted to have a model efficient enough. That's why we picked DistillBERT which is a relativelt small model in comparison to BERT. It uses the Kulback Leiber divergence to optimize its parameters. However, this speed optimization means that we sacrifice on prediction metrics.

Having tried to optimise this model, we also tried to implement the XLNet model which is a bigger model than the previous one. It's a bi-directional transformer so handles dependencies between words better. We decided to try this model because some inputs are long paragraphs (the longest inputs having around 200 words). However once again, due to our limitations we couldn't fully benefit from the advantages of this model.

2.2.2 Hyper parameter tuning

Once the model had been implemented we focused on the hyper parameter tuning. The hugging face models have a function called hyper-parameter search to get the best F1-score possible. However once again we didn't manage to use it due to memory issues (even when we reduced the size of the batch).

Therefore we implemented a grid search to find the satisfying hyper-parameters. This has lead us to take a learning rate = $2 \times 10^{-5}$and a number of training epochs = 15 .

These parameters allow the model to get, on the test set, the results synthetised in the following confusion matrix :

	Not labelled PCL	Labelled PCL
Not predicted PCL	1814	81
Predicted PCL	102	97

Table 1: Confusion matrix of our model

Our model has an F1-score equals to 52 % which is slightly better than the RoBERTa-base baseline of 48%.

Nevertheless, those results aren't fully satisfying because the model isn't good at predicted paragraphs which contains PCL (precision of 54% and recall of 49%).

3 Analysis

3.1 Link between the quality of the model and the level of patronising content

Our model is better at predicting examples with a higher level of patronising level.

PCL level	Samples	True predictions	Accuracy
Level 0	1704	1666	97.8%
Level 1	191	176	92.1%
Level 2	18	3	16.7%
Level 3	89	29	32.6%
Level 4	92	45	48.9%

Table 2: Accuracy of the model on the testing set in function of the level of patronizing content.

Indeed, the model, such as the human, struggles to detect when tweets have patronising content but in a low level (level 2 and 3). Furthermore it also isn't really good at detecting high level of patronising content (level 4).

3.2 Link between the length of the inputs and the model performance

The impact of the length of the input sequence on the model performance is not very important.

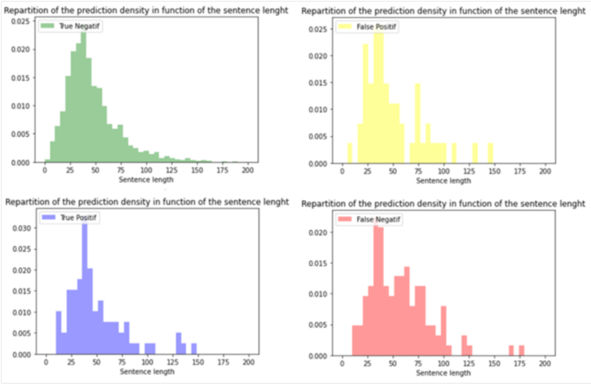

Figure 1: Distributions of the sequence length for TP, TN, FP and FN predicted by our model on the test set.

The small difference that we can see (on the false negative distribution) is that the model has a slightly lower performance on long sequences with patronizing content. It might be because for a long sequence, the language model has difficulty to distinguish between some part of the paragraph that could look as patronizing and some part that are not patronizing.

3.3 Influence of the categorical data on the model predictions

The categorical data provided seems to have a light influence on our model. Indeed, each paragraph that contains patronizing language can be classify in different type of PC speech. There are 7 different categories: Unbalanced power relations (Category 0); Shallow solution (Category 1); Presupposition (Category 2); Authority voice (Category 3); Metaphor (Category 4); Compassion (Category 5) and The poorer, the merrier (Category 6).

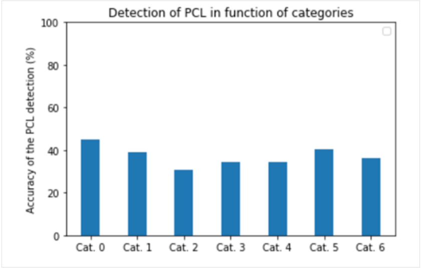

Figure 2: Accuracy of the PCL detection of our model for different categories.

We can see that our model is slightly better to detect some of the categories (0, 1 and 5 for example).

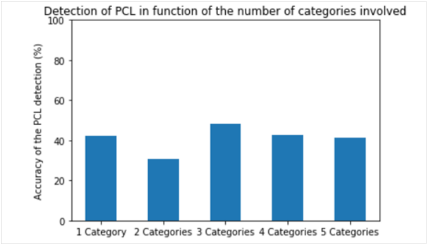

Figure 3: Accuracy of the PCL detection of our model in function of the number of simultaneous categories.

However, we can't see a clear pattern between the number of simultaneous categories of a paragraph and the detection of PCL by our model.

Conclusion

We were slightly disappointed with the performance of our model (52% of F1-score vs. 48 % for the baseline), and would look to improve on this by using either a different base model, or improving the dataset using more examples having PCL or perform other pre-processing methods. Nevertheless, this is a very difficult task as explained in the introduction, and also very subjective so large efforts at model performance may lead to marginal improvements in performance.