NLP Coursework

Suniyah Minhas
CID / 02157482

Jaime Sabal
CID / 01520988

Shay Divald
CID / 02140952

Abstract

The Don't Patronize Me! dataset is contains 10,637 paragraphs annotated as patronising or non-patronising. [1]. They are from different countries and directed towards different vulnerable groups e.g refugee. In this paper, DeBERTa [2] is used to create a binary classification; given a paragraph, the model should classify if the text is patronizing or not.

1 Data Analysis of the Training Data

There is a large imbalance of the labels (81.5% of the samples are labelled 'non-patronising' and the other 18.5% are distributed across the remaining four labels with only 1.4% corresponding to '2'). This imbalance varies depending on the vulnerable community (the 'homeless' group has much more patronising data relative to 'immigrant'), but is consistent for the different countries. This is an issue that we will address in various ways such as reweighing and random insertion (Section 4.1). Moreover, Figure 1 shows how the distribution of the text lengths for the binary labels of the training dataset. The text lengths are similar, but the mean for patronising texts (288) are slightly longer than for non-patronising samples (265).

Given that the original labels are composed the number of people that thought a given text is patronising (0-4), the task seems quite hard in the sense that the problem is intrinsically subjective. In this manner, it might even seem unreasonable to expect our model to predict this feature of a text, since not even humans are able to do so consistently. For example, the text *'Many refugees do n't want to be resettled anywhere , let alone in the US .'* is labelled with a '2', meaning 2/4 people agreed that it contained patronising content. For a person that grew up in the US, this might seem more condescending than to someone from elsewhere. A less subjective alternative problem could be to treat it as a regression task where the labels are expressed as a percentage for the 'confidence' that a given text contains patronising content.

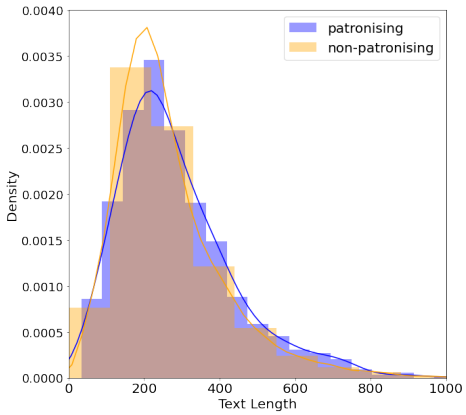

Figure 1: Distribution of text lengths for the two classes (patronising and non-patronising) in the binary Dont-PatronizeMe! dataset.

2 Modelling

2.1 Comparing State-Of-the-Art Models

Table 1 presents the performances using different models. There is clear improvement of the f1-score using DeBerta model. DeBERTa by Microsoft is a BERT-style Self-Attention Transformer version, and it includes two main modifications to improve BERT and RoBERTa. The first technique is the disentangled attention mechanism and the second is the enhanced mask decoder. We found these had a significant impact to performance and therefore have used DeBerta for our final classifier.

2.2 Data Pre-processing

The results for the different preprocessing techniques are summarised in Table 2.

Lemmatization: Lemmatization aims to remove inflectional endings of words and return the base form of a word. This is done for both nouns and

Model	Non-Offensive	Offensive	Macro mean
Bert	0.9442	0.0089	0.4766
RoBerta	0.9532	0.1226	0.5326
DeBerta	0.9593	0.5618	0.7605

Table 1: Comparing f1-scores for Non-Offensive class, Offensive class, and a Macro f1 mean using State-Of-the-Art Models.

verbs in our test data. Using lemmatization significantly reduced the F1 of the patronising class and slightly of the non-patronising class.

Stop Word Removal: Words that do not impact the information in the text (e.g. "the, "a") were removed, however again this resulted in a decreased F1 score for both classes.

Punctuation Removal: Removing the punctuation in the text had minor effect on the results and the f1-score slightly decreased for both classes.

The reason that the performances are not improving using data pre-processing is that the transformer makes an internal embedding of words.[3] The processes of tokenisation is a technique called BPE based WordPiece tokenisation.[4] This technique involves progressively splitting words into subwords. Thus, doing the preprocessing using DeBerta and its tokenisation does not having the same effect we might expect from algorithms such as Word2Vec or Glove.

PreProcessing	**Non-Offensive**	**Offensive**	**Macro**
Base	0.9593	0.5618	0.7605
Lemmatization	0.9591	0.4646	0.7119
Stop Word Removal	0.9562	0.4516	0.7039
Punctuation	0.9582	0.5480	0.7531

Table 2: Non-Patronising (non-offensive), Patronising (Offensive) and macro F1 for different types of preprocessing (all using DeBerta). Note: This was conducted before final hyperparam tuning

2.3 Handling Unbalanced Data

As discussed in section 2, there is a significant imbalance in the dataset towards non-patronising. As our classifier aims to maximise the F1 over the partonising class, dealing with the imbalanced data is important to achieve accurate results. A few methods were tested:

BackTranslation: Back Translation is an augmentation technique that takes an original text written in English, convert it into another language (french in our case) and translates it back to English.[4] 25% of the positive samples were selected to back translate and added as additional positive entries. BackTranslation creates subtle changes in words and word order to generate syntactic data. This however did not produce better results as shown in table 3. This may be because the augmentation did not produce enough differences to the original data, or that taking 25% of the patronising data is not enough to fix the imbalance.

Random Insertion: 25% of the positive samples underwent a random insertion procedure, which inserted random synonyms next to similar words. This technique was shown to have a relatively significant positive impact on the offensive class F1 score and chosen to be used on our final model (see Table 3).

DownSampling: This method was used to simply decrease the amount of data used for the non-patronising class. This had the most significant decrease in both the offensive and non-offensive class, this is likely because less data could lead to the model missing out on learning certain important concepts. It may have prevented the classifier from distinguishing between positive and negative classes. This was investigated with different scales of down-sampling, but none helped to improve the model.

Reweighing: Reweighing gives a different emphasis to the loss computed for different samples based on whether they belong to the majority or the minority classes. Putting a higher emphasis in the loss of patronising data did decrease the F1 of the non-patronising, however the patronising increased as expected. As this is our primary interest reweighing was determined as an improvement feature.

3 Hyper-Parameter Tuning

In order to tune the hyper-parameters, we used the `wandb` library and API to run sweeps through different combinations of hyper-parameters; learning rate, batch size, and warm-up steps. *He et al* [2] provides a range of values to tune these hyper-parameters for the DeBERTa pre-trained model on a downstream task like classification. We found that the best learning rate was 5×10^{-6}, batch size 16 and warm-up steps 50.

Balancing Technique	Non-Offensive	Offensive	Macro mean
Base	0.9593	0.5618	0.7605
Backtranslation	0.9575	0.5330	0.7452
Insert words	0.9612	0.5706	0.7659
Downsampling	0.9263	0.5274	0.7268
Reweighing	0.9575	0.5668	0.7622

Table 3: Comparing f1-scores for Non-Offensive class, Offensive class, and a Macro f1 mean using different balancing techniques (all using DeBerta). Note: This was conducted before final hyperparam tuning

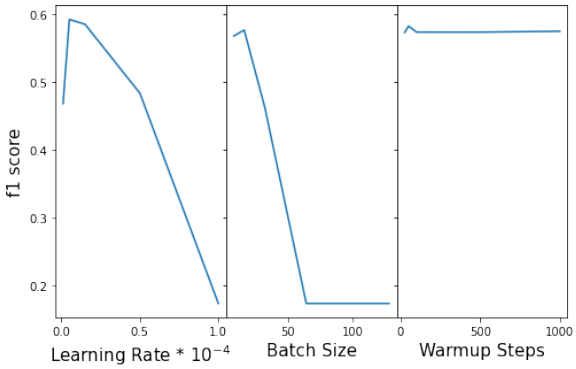

Figure 2: F1 scores for different values of learning rate

4 Analysis

4.1 Correlation Between Model Performance and Extent of Patronising Content

There is a large correlation between higher levels of patronising content and the performance of the classifier (with f1-score of second level:0.19, third:0.66, fourth:0.84). More patronizing content is correlated to higher f1-score. This is likely because of the ambiguity of the text within lower patronising levels (as even the human marks were split).

4.2 Impact of Input Sequence Length and Model Performance

The text length did have an impact on the level of patronising content.The general trend (see Fig 3) until length of 700 is that the F1 score decreases for longer texts. This is likely because longer sentences have more variability and are more difficult to interpret and to extract the specific words that make it patronising. Beyond length of 700 we get extreme f1-scores with very high/low values. However, the frequency is very low, so the f1-score for these sentences could be regarded as outliers, dependent on only a few examples. To reduce this ambiguity with longer texts, we could split longer sentences or create data augmentation by combining the ones with the same patronising levels.

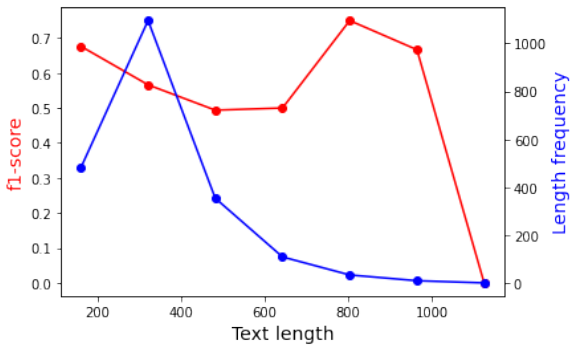

Figure 3: Model performance for different length of the input sequence.

4.3 Influence of Categorical Data on Model Predictions

The categorical data did have some extent of influence. However the trend is not as clear as one might have expected; for example Tanzania has a smaller amount of training data, but this did not have a large impact on it's performance, Most of the categories seem to have a fairly similar relative performance. The slight deviations could be country and culture specific e.g. gb has a lower f1 perhaps because more subtle patronisation was harder for the classifier to detect. The Keywords exhibited a similar trend, some groups e.g. "in-need" had higher F1 (0.7) and some groups (perhaps with a lack of testing examples) had closer to 0, but in general there was not that much variation.

5 Conclusion and final results

To conclude it was found DeBerta was the best Transformer model for our dataset. Using data preprocessing did not help this model due to it's already functioning tokenizer. However data augmentation and reweighing did help to alleviate some of the biases in the dataset. Other techniques such as downsampling or back-translation were not as effective. Hyperparameters were tuned in order to get a final patronising F1 score of 0.5931 on the official dev set and a score of 0.5331 on the official CodaLab page (user is nlp_imperial on CodaLab). The notebook is accessible through https://colab.research.google.com/drive/1J3zBbUxtdotnuXMg3yDzeq80VVK0aWDk?usp=sharing.

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, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[2] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[4] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.