NLP

Amrit Diggavi Seshadri
CID 01796915
asd20@ic.ac.uk

Guillem Garrofé Montoliu
CID 02163491
gg921@ic.ac.uk

Matteo Cuccorese
CID 02158568
mc1721@ic.ac.uk

1 Introduction

NLP Transformer's applications range from text generation to machine translation. In this project, we present an NLP model to perform text classification. Specifically, we use a fine-tuned ELECTRA-based model to predict whether or not text contains Patronising or Condescending Language (PLC). We show that word-context has great importance for this purpose and that balancing the length of sentences in the training data set through data augmentation proves to improve model performance. Experiment results demonstrate that our fine-tuned ELECTRA model makes an improvement of 13.56% over the RoBERTa's baseline's f1 score, on the test set - better identifying paragraphs with PLC content.

Code: https://tinyurl.com/nlp-cw-code

CodaLab Username: ggarrofe

2 Data analysis

In this section, we analyse the data set's properties. We study the distribution of sentence lengths for the two labels (PLC and non-PLC) and examine how distinguishable the two labels text content are in the "Don't patronize me!" dataset. (Pérez-Almendros et al., 2020).

2.1 Analysis of the class labels

As shown in Figure 1, the dataset is unbalanced (see the number of samples in the peak). There are 7581 non-patronising text samples and just 794 patronising text samples in the training set.

In terms of length, the non-patronising examples' have a mean of 46.72 words, whereas the patronising texts have 54.02 words on average. We also analyse the length distribution for the two classes, and find that both classes are distributed similarly. However, we can see that long texts are far less frequent and hypothesise that the model will probably struggle more when classifying longer sentences/paragraphs.

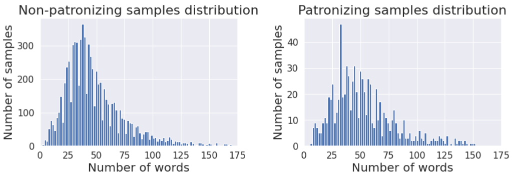
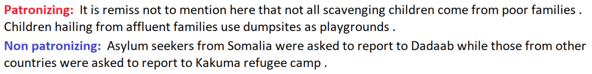


Figure 1: Distribution of text lengths for the two class labels in the training set.

2.1.1 Qualitative assessment of the dataset

The task of identifying PLC content is not simple. As an experiment, the project's authors tried to classify ten random paragraphs from the training set, and we found that we just could not agree on the PLC classification of more than three of them. Hence, we conclude that this classification task is difficult to perform objectively by humans, and it can only be done correctly by experts. Therefore, when manipulating the data set in our experiments, we are very careful not to alter paragraph meanings.



Patronizing: It is remiss not to mention here that not all scavenging children come from poor families . Children hailing from affluent families use dumpsites as playgrounds .

Non patronizing: Asylum seekers from Somalia were asked to report to Dadaab while those from other countries were asked to report to Kakuma refugee camp .

Figure 2: Sample sentences for the two class labels from the training set.

3 ELECTRA model

Given the limitations of resources imposed by Google Colab, we wanted to train a transformer-based model with a reasonable number of parameters that learns efficiently and achieves high accuracies on classification benchmarks.

In this vein, we chose the ELECTRA-base (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) (Clark et al., 2020) model as a backbone network for fine tuning. The ELECTRA-base model requires relatively less computation and despite its smaller size, substantially outperforms other MLM-based methods such as BERT (Devlin et al., 2018) ALBERT (Lan et al., 2020), XLNet (Yang et al., 2020) and RoBERTa (Liu et al., 2019) when evaluated on the GLUE natural language understanding benchmark (Wang et al., 2018). Our work reaffirmed the superiority of ELECTRA's architecure as we demonstrate improvement over the RoBERTa's baseline for the PLC classification task.

ELECTRA is based on masked language modelling (MLM) principles. These models replace some tokens of a sentence with masks and train a model to reconstruct the original sentence. Until the appearance of ELECTRA in 2020, MLM models were able to learn only with 15% of the tokens per example. However, with the proposed *replaced token detection* approach, the model learns from all input positions, leading to much faster learning. Specifically, this approach replaces some tokens with samples proposed by a small MLM instead of masking them; then, trains a discriminator (i.e., the model that we will use as a classifier later) to predict which tokens are original and which are not.

3.1 Latent representations to class labels

The pre-trained ELECTRA model that we use outputs a single contextualized hidden representation for each word in the input sentence. We average these word-level representations to obtain a sentence-level representation of the given text sample and feed this sentence-level representation to a single fully connected layer to get a probability for each class label. We train our model to minimize the cross-entropy loss between our predicted labels and the ground truth.

We also experimented with combining word representations by concatenating min, mean and max functions over the output sequence of word representations but found better performance by using the mean of word representations directly.

3.2 Hyper-parameter tuning

To choose the best configuration of hyper-parameters for our dataset, we evaluate the model's performance on data held out during training (i.e., the test set). We evaluate model performance on a wide range of batch sizes and learning rates (i.e. learning rates from $5 \cdot 10^{-05}$ to $1 \cdot 10^{-04}$ and batch sizes from 16 to 64 samples), and select the configuration that provided best f1 scores. As can be seen in figure 3, the best configuration was achieved for *learning rate* $= 5 \cdot 10^{-05}$ *and batch size of* 16 *samples*.

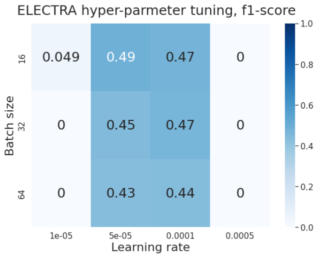


Figure 3: F1-score for different learning rates and batch sizes using the ELECTRA transformer, evaluated using the test set.

4 Data processing

4.1 Pre-processing

Unlike conventional non-contextual language models such as rule-based methods, bag of words methods or CNNs, in contextual models, each encoded word-representation is computed with attention over all other tokens in the sequence. As a consequence, stopwords and word capitalisation (which are normally removed or filtered for non-contextual language models) provide important language-cues and help our model understand text meaning.

As a check, we filtered and removed stop-words for the PLC detection problem, and observed that an ELECTRA model fine-tuned over this stopword-free text failed to classify any text as patronising - emphasizing how important it is that we include stop-words in sentences when training these deep contextual models. We further note that recent works (Qiao et al., 2019), have shown that stop-words receive as much attention as non-stop words for transformer based models.

All further models disccused make use of stop-words in the input text sequence.

Augment. Method	Short Sent.	Long Sent.
None	84.91%	80.19%
Soft Augment.	87.75%	82.62%
Hard Augment.	87.61%	83.92%

Table 1: Overall accuracies for shorter sentences (fewer than 54 words) and longer sentences (more than 54 words) on the test set using different data augmentation techniques.

4.2 Data Augmentation

As shown in Table 1, we observe that the **length of the input sentences impact the prediction accuracy of our model** as our model tends to achieve lower overall accuracy for longer sentences than it does for shorter sentence sentences. To remedy this, we explore two alternative data augmentation techniques to increase the frequency of longer samples in our training set:

1. **Soft Augmentation**: We sample sentences randomly from our training set according to a weighting of sentence length - so that the longer a sentence, the more likely we are to sample it. Then, we randomly replace up to ten words in each sample with their synonym words to get new sentences. Finally, we add these new long sentences to our training set.
2. **Hard Augmentation**: In this case, the same synonym procedure is followed as above, except that we select long sentences deterministically - directly selecting those sentences with more than 54 words.

As shown in Table 1, both soft data augmentation and hard data augmentation improve the accuracy of our model for longer sentences in the test set. However, we observe best f1-scores for the task of identifying patronizing text by using the hard data augmentation technique (see Table 2).

Model	f1-score
RoBERTa	46.80 %
ELECTRA	49.16 %
ELECTRA with soft augmentation	49.18 %
ELECTRA with hard augmentation	50.87 %

Table 2: F1-score for the RoBERTa baseline and variations of our Electra based-model after 1 epoch of fine-tuning.

5 Results

5.1 Early Stopping

As depicted in Figure 4, the performance of our final model on the test set deteriorates for fine-tuning after three epochs. Therefore, we stop training at three epochs to achieve the best performance on the test set and report a final f1-score of 53.15% (i.e., 13,56% better f1-score than RoBERTa) on the test set.

5.2 Category influence on predictions

From Figure 4 (b), we can see that those **samples that contain a higher level of PLC content are correctly identified by our model more often than samples with less PLC content**.

Given a text sample, we observe that different text origin and text topic influences whether our model correctly classifies it as PLC or not. (Figure 4) We interpret this as text originating from some countries are more prone to use a higher level of patronising content than others. Moreover, some keywords specific to patronizing prone topics seem to be used more often alongside patronising language; making text that use these keywords more easy to classify (Figure 4) (e.g. sentences containing words: migrant, hopeless and in-need are more often correctly identified as PLC or non-PLC). **We conclude that the provided categorical data may influence the model predictions to a great extent.**

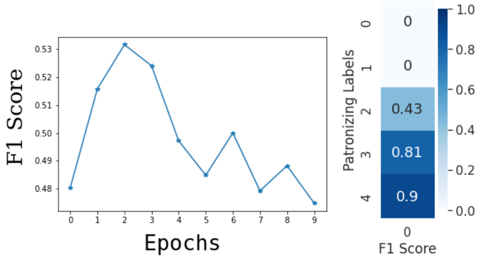


Figure 4: (a) F1 score on the test set during fine-tuning with a hard data-augmented training set. (b) Predictions for different levels of patronizing content.

f1 <0.6	0.6 <f1-score <0.8	0.8 <f1
Country influence		
us, bd	pk, nz, tz, ph, gb, hk, my, ca, gh, sg, in, au	ng, za, ke, lk, jm, ie
Keyword influence		
refugee, disabled	homeless, vulnerable poor-families women, immigrant	hopeless, in-need, migrant

Table 3: f1-scores for the predictions of patronising examples that belong to different categories.

6 Conclusion and directions for further experimentation

In this paper, we have presented a model that outperforms the RoBERTa baseline in identifying PLC content. From the results (see Image 4(b)), we can conclude that our model correctly perceives higher levels of patronising content. Moreover, some countries may use a more patronising language, and some keywords may be related to condescending language more often than others; in these cases, our model may better identify which texts contain patronising language. We expect transformer models using more sophisticated architecures (e.g. GPT-3 (Brown et al., 2020) or DeepNet (Wang et al., 2022)) to improve performance for this task and suggest experiments with these models as an avenue for future experimentation.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

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.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. 2022. Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.