PiCkLe at SemEval-2022 Task 4: Boosting Pre-trained Language Models with Task Specific Metadata and Cost Sensitive Learning

Manan Suri
Netaji Subhas University of Technology
New Delhi, India
manan.suri.ug20@nsut.ac.in

Abstract

This paper describes our system for Task 4 of SemEval 2022: Patronizing and Condescending Language Detection. Patronizing and Condescending Language (PCL) refers to language used with respect to vulnerable communities that portrays them pitifully and is reflective of a sense of superiority. Task 4 involved binary classification (Subtask 1) and multi-label classification (Subtask 2) of Patronizing and Condescending Language (PCL). For our system, we experimented with fine-tuning different transformer-based pre-trained models including BERT, DistilBERT, RoBERTa and ALBERT. Further, we have used token separated metadata to improve our model by helping it contextualize different communities with respect to PCL. We faced the challenge of class imbalance, which we solved by experimenting with different class weighting schemes. Our models were effective in both subtasks, with the best performance coming out of models with Effective Number of Samples (ENS) class weighting and token separated metadata in both subtasks. For subtask 1 and subtask 2, our best models were finetuned BERT and RoBERTa models respectively.

1 Introduction

Patronizing and Condescending Language (PCL) in the context of vulnerable communities refers to language that portrays a sense of superiority or has a tendency to view communities with a pitiful and passionate lens. PCL works subtly and is intricately associated with the way that words and phrases are used. This makes it difficult to classify PCL as compared to overtly offensive language where the nature of words and phrases is clearly negative. Since the harms associated with PCL are not always evident, it is often used inadvertently by actors trying to help these communities. Recognising PCL is important because the expression of PCL leads to a paradigm where discrimination and injustices are routinised and made less visible (Ng, 2007). Use of PCL also feeds into stereotypes (Fiske, 1993), reinforces societal power dynamics and avoids deep-rooted societal problems, providing surface-level solutions for the same (Chouliaraki, 2010). Task 4 of SemEval 2022 (Pérez-Almendros et al., 2022) aims to identify PCL with Subtask 1 working towards binary classification and Subtask 2 working towards the multi-label classification of PCL.

Our strategy involves using state-of-the-art Pre-Trained Language Models (PLMs) and finetuning them for our specific task. We work with BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al., 2019), ALBERT (Lan et al., 2019) as our models. Additionally, we design effective and simple approaches to optimize our models, by experimenting with different cost-sensitive class weighting methods and working with token separated metadata to enhance performance. With an increasing number of PLMs, each having millions of parameters and being computationally expensive to train, it is essential to make the right model choice. This paper provides a comprehensive analysis of the performance of different models. This can help determine baselines for similar text classification tasks. For model replicability, our code is available online.[1]

2 Background

2.1 Task Description

Patronizing and Condescending Language (PCL) refers to language which may seem kind or helpful but is reflective of a sense of superiority. SemEval 2020 Task 4: Patronizing and Condescending Language Detection (Pérez-Almendros et al., 2022) had two subtasks that dealt with the identification of PCL and the categories used to express it. These were seen specifically in reference to communities

[1] https://github.com/MananSuri27/PatronisingAndCondescendingLanguage

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 464 - 472

	PCL	Non PCL	unb	sha	pre	aut	met	com	the
Training Set	794	7581	574	160	162	192	363	145	29
Development Set	199	1895	142	36	62	38	106	52	11

Table 1: Distribution of categories across the training and development set. The labels 'unb', 'sha', 'pre', 'aut', 'met', 'com' and 'the' refer to 'unbalanced power relations', 'shallow solution', 'presupposition', 'authority voice', 'metaphor', 'compassion', and 'the-poorer-the-merrier' respectively.

being identified as being vulnerable and having unfair treatment in the media.

- **Subtask 1:** Subtask 1 was binary classification, where given a paragraph the system was supposed to predict whether it contains any form of PCL. The basis of evaluation was F1 score on the positive class, PCL.
- **Subtask 2:** Subtask 2 was a multi-label classification task where given a paragraph, we were supposed to predict what categories of PCL the paragraph subscribes to. Pérez-Almendros et al. (2020) determined these categories based on previous works and their research on PCL. The 7 categories considered are 'unbalanced power relations' (unb), 'shallow solution' (sha), 'presupposition' (pre), 'authority voice' (auth), 'metaphor' (met), 'compassion' (com), and 'the-poorer-the-merrier' (the). The basis of evaluation was average F1 score across the given classes.

2.2 Data Description

This task is based on the Don't Patronize Me!(Pérez-Almendros et al., 2020) dataset by the task organizers. For this paper, we have considered the practice split offered by the organizers as the split between train and development set. The train, development and test set contain 8375, 2094 and 3831 rows of data respectively.

The paragraphs in this dataset have been selected from news stories and have been annotated with labels specifying whether they qualify as PCL and the categories of PCL that they belong to. The dataset includes additional information about the paragraphs— including the country of reference and the keyword. The keywords clarify the context of the paragraph. The included keywords are: 'disabled' (dis), 'homeless' (hom), 'hopeless' (hop), 'immigrant' (imm), 'in-need' (need), 'migrant' (mig), 'poor families' (poor), 'refugees' (ref), 'vulnerable' (vul) and 'women' (wom). The dataset includes reports from 20 countries.

Table 1 shows the distribution of labels in the train and dev set. We can observe that the distribution of classes is heavily imbalanced. In the training set, only 9.5% of the samples belong to the PCL class. Similarly, in the multi-label category, 72% of all samples with PCL have the class label of 'unb' for the training set.

3 System Overview

3.1 Finetuning Pre-trained Language Models (PLMs)

Pre-training in NLP is a technique that involves training general-purpose language representations through a large set of unannotated text data. It is beneficial for downstream tasks and avoids training a new model from scratch. Pre-training leads to a better generalization performance and helps in convergence on downstream tasks because it provides a better model initialisation. Most NLP datasets contain limited human-annotated samples, due to which there is a tendency to cause overfitting. Pre-training can be regarded as a kind of regularization, preventing overfitting on smaller datasets (Erhan et al., 2010). Pre-training models followed by fine-tuning them for downstream tasks has shown good performance on many NLP tasks (Dai and Le, 2015; Radford and Narasimhan, 2018; Peters et al., 2018).

Briefly discussing the PLMs we have used for this task:

BERT: BERT refers to Bidirectional Encoder Representations (Devlin et al., 2019). It uses bidirectional transformers (Vaswani et al., 2017) pretrained using a combination of Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). It learns deep bidirectional representations by jointly conditioning on both left and right context layers.

RoBERTa: RoBERTa refers to A Robustly Optimized BERT Pretraining Approach (Liu et al., 2019). It builds on BERT and modifies key hyperparameters, such as training with larger mini-batches and learning rates. RoBERTa also removes

BERT			
Class Weighting	**precision**	**recall**	**F1**
None	0.672	0.402	0.503
INS	0.456	0.698	0.552
ISNS	0.515	0.598	0.553
ENS	0.541	0.658	0.594

RoBERTa			
Class Weighting	**precision**	**recall**	**F1**
None	0.667	0.462	0.546
INS	0.370	0.779	0.502
ISNS	0.510	0.648	0.571
ENS	0.455	0.678	0.544

DistilBERT			
Class Weighting	**precision**	**recall**	**F1**
None	0.703	0.392	0.503
INS	0.476	0.492	0.484
ISNS	0.564	0.508	0.542
ENS	0.494	0.593	0.539

ALBERT			
Class Weighting	**precision**	**recall**	**F1**
None	0.513	0.296	0.376
INS	0.213	0.764	0.333
ISNS	0.377	0.638	0.474
ENS	0.389	0.739	0.510

Table 2: Subtask1: Binary Classification; The performance of the PLMs we have considered: BERT, RoBERTa, DistilBERT and ALBERT on the dev set, with different class weighting techniques.These systems also included token separated metadata. The class weighting strategies (INS- Inverse Number of Samples, ISNS- Inverse of Square Root of Number of Samples, ENS- Effective Number of Samples) are discussed in section 3.2 .

BERT								
Class Weighting	**unb**	**sha**	**pre**	**aut**	**met**	**com**	**the**	**avg**
None	0.339	0.070	0.270	0.190	0.191	0.314	0.000	0.196
INS	0.409	0.190	0.278	0.218	0.245	0.386	0.098	0.261
ISNS	0.440	0.148	0.279	0.156	0.182	0.384	0.098	0.241
ENS	0.427	0.197	0.277	0.223	0.256	0.396	0.129	0.272

RoBERTa								
Class Weighting	**unb**	**sha**	**pre**	**aut**	**met**	**com**	**the**	**avg**
None	0.838	0.287	0.209	0.085	0.029	0.642	0.000	0.299
INS	0.838	0.324	0.358	0.370	0.328	0.642	0.062	0.417
ISNS	0.834	0.310	0.315	0.317	0.312	0.642	0.109	0.405
ENS	0.838	0.313	0.367	0.379	0.323	0.642	0.079	0.420

DistilBERT								
Class Weighting	**unb**	**sha**	**pre**	**aut**	**met**	**com**	**the**	**avg**
None	0.451	0.092	0.162	0.137	0.121	0.335	0.000	0.185
INS	0.484	0.157	0.255	0.205	0.216	0.394	0.091	0.257
ISNS	0.377	0.147	0.244	0.168	0.184	0.306	0.043	0.210
ENS	0.480	0.217	0.288	0.205	0.212	0.400	0.080	0.269

ALBERT								
Class Weighting	**unb**	**sha**	**pre**	**aut**	**met**	**com**	**the**	**avg**
None	0.826	0.000	0.009	0.000	0.010	0.490	0.000	0.191
INS	0.828	0.100	0.258	0.088	0.079	0.449	0.000	0.257
ISNS	0.548	0.177	0.223	0.147	0.083	0.456	0.000	0.233
ENS	0.436	0.217	0.294	0.238	0.228	0.418	0.000	0.262

Table 3: Subtask2: Category Classification; The performance of the PLMs we have considered: BERT, RoBERTa, DistilBERT and ALBERT on the dev set, with different class weighting techniques.These systems also included token separated metadata. The columns represent the F1 score for different classes and the average F1 score across all classes. The class weighting strategies (INS- Inverse Number of Samples, ISNS- Inverse of Square Root of Number of Samples, ENS- Effective Number of Samples) are discussed in section 3.2 .

the Next Sentence Training (NSP) objective in BERT.

DistilBERT: DistilBERT (Sanh et al., 2019) is a small, fast, cheap and light transformer based model trained by distilling BERT base. It has lesser parameters, and runs faster but still conserves a large proportion of BERT's performance. DistilBERT uses a triple loss combining language model, distillation and cosine distance losses to leverage the advantage gained by larger models during pre-training.

ALBERT: A Lite BERT for Self-supervised Learning of Language Representations (Lan et al., 2019) is a modification of BERT which efficiently allocates model capacity, reducing the training time and memory consumption. It reduces parameters by factorised embedding parameterization- where the embedding matrix is decomposed to a lower dimension and projected. Layer sharing across layers reduced redundancy. Inter-sentence coherence loss based on Sentence Order Prediction (SOP) is also employed by the model.

We finetuned the PLMs for both subtasks by stacking a dropout layer followed by a dense layer on top of the PLM model. The dropout layer before the dense classification layer is added for regularization. In Subtask 1 we use Sigmoid activation to predict binary labels. In Subtask 2 we use sigmoid activation in the final layer rather than softmax as it allows us to deal with non-exclusive labels. For BERT, DistilBERT and ALBERT we use the features of the [CLS] token and for RoBERTa the <s> token. The performance of the models along with the other strategies we have used is present in Table 2 and Table 3 for Subtask 1 and Subtask 2 respectively.

3.2 Utilising metadata

We have attempted to enrich the PLMs with additional metadata provided in context to the paragraphs in the task. In this setup, more data is available to the model. This is based on the idea that more meaningful data leads to better performance on classification. The same has been observed by other researchers who have experimented with including task-specific data in NLP (Ostendorff et al., 2019; Zhang et al., 2019).

Pérez-Almendros et al. (2020) included ten keywords related to potentially vulnerable communities that are widely covered in media and have had the propensity of receiving condescending treatment, namely: disabled, homeless, hopeless, immigrant, in need, migrant, poor families, refugee, vulnerable and women. Since PCL involves a subtle use of language, we believe that contextualizing whether a phrase is PCL also depends on the context of which community or situation is being referred to.

496 @@26214070 refugee hk 3
Hundreds of thousands of Rohingya refugees living in sprawling camps in Bangladesh are celebrating the Muslim holiday of Eid al-Adha, praying for better lives as they wonder if they'll ever again celebrate at their homes in Myanmar. People streamed into makeshift mosques in the camps, the children dressed in new clothing . Those who could afford it feasted on buffalo meat. Muslims often...

350 @@21894186 homeless lk 4
It can not be right to allow homes to sit empty while many struggle to find somewhere to live, others having to sleep rough on pavements during Christmas, hoping against hope, for some charity to provide shelter . The number left homeless and destitute is alarming not necessarily at Christmas ?

Table 4: Two examples from the dataset to get an intuitive sense of the advantage that using keyword might add to contextualise the paragraph. Both paragraphs describe *home*, but one in the context of *refugees* and the other in context of *homelessness*. The data included in the first row of each paragraph includes the serial number, paragraph ID, keyword, country and annotation.

To understand this, let us consider two examples (Table 4) from the dataset, where paragraphs with ID @@26214070 and @@21894186 are tagged with the keywords "refugee" and "homeless" respectively, and use the word "home" in different contexts. With the "refugee" tag, we are given to understand that "home" refers in a very specific way to a place in the actor's country of origin whereas in the "homeless" context, "home" refers to accommodation or the lack there-of. The purpose of including additional metadata thus was to add to the contextualizing abilities of the model.

We include the metadata by adding it to the input string itself as another sentence in itself; separating

the metadata from the text with the [SEP] token in case of BERT, DistilBERT and ALBERT and the </s> token for RoBERTA.

Therefore, the input in case of BERT, DistilBERT and ALBERT looks like:

[CLS] keyword [SEP] paragraph [SEP],

and in the case of RoBERTa, looks like:

<s> keyword </s> paragraph </s>.

We make this system design choice based on the following ideas:

- The chosen PLMs are strong enough to learn how the metadata interacts with the input sequence, considering that we have enough training samples available.
- Using token separated metadata rather than concatenating another model reduces the number of additional parameters to be trained.
- Using the [SEP] and </s> tokens help us utilize the power of pre-training which wouldn't have been convenient if we defined new special tokens to separate the metadata instead of using the predefined special tokens.

	Subtask 1	
Model	**F1 with**	**F1 without**
BERT	0.595	0.556
RoBERTa	0.571	0.566
DistilBERT	0.534	0.510
ALBERT	0.474	0.450
	Subtask 2	
Model	**Avg F1 with**	**Avg F1 without**
BERT	0.272	0.229
RoBERTa	0.420	0.387
DistilBERT	0.269	0.249
ALBERT	0.262	0.238

Table 5: The performance of the chosen models with and without use the of token separated metadata on the development set. For each model, the same parameters including class weights are used to ensure comparability.

Table 5 includes a comparison of the different models we have used, with and without the token separated metadata. For comparability, the same parameters including class weights are used for each model.

3.3 Cost Sensitive Learning

One of the challenges in the task was a heavy imbalance in the number of samples in the given classes in the training data. The positive class for the binary classification task (Subtask 1) was underrepresented where the number of samples with PCL was only around 9.5% of the training set. Similarly, subtask 2 which included multi-label classification had a large proportion of samples from only 2 classes- 'unb' and 'met' (72% and 45% of all samples with PCL respectively), and some classes such as 'the' were heavily underrepresented(3% of training samples with PCL). This varying distribution is a significant issue while training because it becomes a challenge for us to ensure that our model learns the characteristics of the minority classes as well.

Class imbalance is a common issue in many real-world datasets, and many techniques have been developed to mitigate this problem: changing the data (undersampling the majority class, oversampling the minority class, data augmentation by using synonyms or other such methods) or adjusting the model (like changing the performance metric). We found in our experiments that data manipulation techniques only provide a marginal performance boost, and the same has been observed by other researchers working on transformer-based models in text classification tasks (Tayyar Madabushi et al., 2019).

The technique that we have used is cost-sensitive learning (Elkan, 2001), which is based on the statistical concept of importance sampling. Importance sampling refers to weights being assigned to samples in a way that matches the given distribution of data. Mathematically, the loss function is modified to account for a multiplier that represents the weight of the class. This method doesn't modify the distribution of the imbalanced data directly.

For a single prediction x with a gold label of a given class, the loss function is then modified as:

$$loss(x, class) = weight[class]\Theta \qquad (1)$$

where Θ represents the original loss function.

This can be interpreted as adjusting the cost of misclassification of the given classes, practically increasing the cost for misclassification of an important class such as a minority class by assigning a larger weight to it.

We explored different cost weighting strategies which are discussed below. Table 6 describes the count as well as class weights of different cate-

	PCL	Non PCL	unb	sha	pre	aut	met	com	the
Count	794	7581	574	160	162	192	363	145	29
INS	10.55	1.10	3.65	13.09	12.93	10.91	5.77	14.44	72.21
ISNS	297.22	96.19	1.91	165.55	164.52	151.12	109.91	173.90	388.85
ENS	11.85	2.80	16.30	20.32	20.22	19.01	16.68	21.18	64.27

Table 6: Count of different categories in the training set and the calculated weights according to the Inverse of Number of Samples (INS), Inverse of Square-root of Number of Samples(ISNS) and Effective Number of Samples (ENS) schemes.

gories for subtask 1(PCL, No PCL) and subtask 2 (unb, sha, pre, aut, met, com, the) according to the different weighting schemes we discuss below.

1. Inverse of Number of Samples (INS)

For finding the class weight for a given class, we simply take the inverse of the number of samples in the class. It is a simplistic way of ensuring that under-represented classes have a higher weight and classes with a large number of samples have a lower weight. INS class weighting can be described by the following equation:

$$weight[class] \propto \frac{1}{n_{class}} \tag{2}$$

where n_{class} is the number of samples in that class.

2. Inverse of Square Root of Number of Samples (ISNS)

The INS method increases the recall by decreasing the number of false negatives, we observed that because the weight of the majority class had been highly diminished, the precision is still low because of a higher number of false positives. The ISNS method mathematically is the inverse of the root of class frequency. Mathematically, this means that the class weights are larger numeric quantities here than in the INS method and more importantly, the problem of the weights of the majority class being highly diminished for our dataset is mitigated by this method. The ISNS class weighting strategy can be summarised by the following equation:

$$weight[class] \propto \frac{1}{\sqrt{n_{class}}} \tag{3}$$

where n_{class} is the number of samples in that class.

3. Effective Number of Samples (ENS)

A third class weighting strategy that we consider is the Effective Number of Samples (ENS) method which was described by Cui et al. (2019). The authors argue that as the number of samples of a class increases, the benefit added by each new sample will diminish. Instead of considering individual rows of data as singular points in the space, this model considers them as small neighbouring regions, the effective number of samples is then calculated mathematically as the effective volume of samples, given by the simple formula:

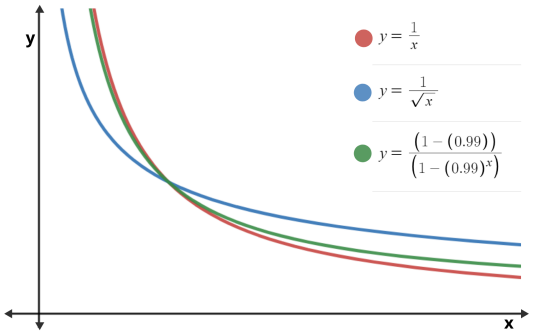

Figure 1: The plots of the mathematical functions that define INS, ISNS and ENS class weighting schemes. The actual class weight may involve additional constants.

$$ENS[class] = \frac{1-\beta^{n_{class}}}{1-\beta} \tag{4}$$

where β is a parameter that can take values as 0.9, 0.99, 0.999 and so on, ENS refers to the effective number of samples and n_{class} is the number of samples in the given class.

The weight of the class is then defined as being the inverse of the effective number of samples.

$$weight[class] \propto \frac{1-\beta}{1-\beta^{n_{class}}} \tag{5}$$

For a very high value of β , this class weight comes very close to the INS class weight.

Figure 1 is a graphical representation of the mathematical functions that define the class weights we have discussed above. The performance of the PLMs we have considered with different class weighting strategies can be seen in Table 2 and Table 3 (Subtask 1 and Subtask 2 respectively).

Subtask 1: Binary Classification				
RANK	TEAM NAME	PRECISION	RECALL	F1-SCORE
38	Team PiCkLe	0.46	0.5804	0.513

Subtask 2: Categorical Classification									
RANK	TEAM NAME	UNB	SHA	PRE	AUT	MET	COM	THE	F1 AVG
35	Team PiCkLe	0.1091	0.2254	0.1439	0.2101	0.1916	0.0651	0.1151	0.1515

Table 7: PiCkLe's Results on the official leaderboard for subtask 1 and subtask 2.

4 Experimental Setup

To ensure comparability, all models are trained on the same train, dev and test split. Further, the train-dev splits are the same splits provided by the task organizers in the practice splits.

The models were developed on Keras[2] (Chollet et al., 2015), and implemented using the transformers library by HuggingFace[3] (Wolf et al., 2019). We experimented with learning rates of 1e-5,2e-5 and 5e-5 for all models, finding the best results at 2e-5. For all the models, we fixed the max length parameter at 256 tokens and the batch size parameter to 16. The finetuning for the models was performed on Google Colab GPU. We trained each model for 1-2 epochs and found the best results at 2 epochs. The value of β for ENS class weighting was taken as 0.9997 for Subtask 1 and 0.99 for Subtask 2 based on experiments with different values.

Class weighting was implemented using the *class_weight* parameter during model fitting. We used the Autotokenizer offered by HuggingFace's transformers library to tokenize our inputs. We implemented the token separated metadata by setting the *add_special_tokens* parameter of the tokenizer as True and using the *text_pair* parameter. We used the Adam (Kingma and Ba, 2014) optimiser by Keras. The loss function used is binary cross-entropy and categorical cross-entropy for Subtask 1 and Subtask 2 respectively.

5 Results and Analysis

Based on the performance of different PLMs with different configurations (Table 2 and Table 3) on the development set, for Subtask 1 we submitted a finetuned BERT model with ENS class weighting and token separated metadata. For Subtask 2 we submitted a finetuned RoBERTa model with ENS class weighting and token separated metadata.

Our results in the given subtasks on the test set are shown in Table 7 for subtask 1 and subtask 2. We have ranked 38 on subtask 1 with an F1 score of 0.513. For subtask 1, our best performing model on the test set is BERT with token separated metadata and ENS class weighting. This model performs better than the baseline RoBERTa model and falls in the top half of all the models entered into the competition. For subtask 2, our submitted model for the evaluation phase was RoBERTa with token separated metadata and ENS class weighting. This model seems to have performed very well on the development set however it has failed to give the same performance on the evaluation set. While we saw an average F1 of 0.419 on the development set, we get a lower F1 of 0.1515 on the evaluation set. This model still ranks 35 on the leaderboard and has performed better than the baseline model. Moreover, this model is amongst the top 20 models in terms of the F1 scores on 'the' class which was in the smallest proportion in the training set, showing how the cost-sensitive learning that we have performed has been effective in taking into account the minority classes.

While we recognise that our model has performed well for subtask 1, the model still lacks in terms of learning what exactly represents PCL. Recognising PCL is an inherently difficult task because of the subtle nature of the language used and the lack of an exact benchmark of what constitutes PCL. With further tuning of parameters and attempts at improving paragraph representations, we may improve the performance of our existing model.

For Subtask 2, on analysis, we believe that a possible issue with the chosen model for submission is that its performance on the development set is heavily biased by the distribution of labels in the training and development set. This is evident by

[2] https://keras.io/

[3] https://huggingface.co/docs/transformers/index

the fact that 'unb' which is in the highest proportion in Subtask 2 has a development F1 score of 0.838 on the 'unb' class (which also raises the average F1), signifying that a large number of samples have been correctly classified as 'unb' in the development set. We believe that since this model has a high tendency to classify samples as 'unb', it gained a high F1 for 'unb' on the development set where this class was statistically well represented with 71% of the samples with PCL having the class 'unb' in the development set, however, the same distribution may not present in the evaluation set revealing a pitfall of our model.

6 Conclusion

The task of predicting Patronizing and Condescending Language (PCL) is relatively new in the field of Natural Language Processing and comes with its challenges as discussed in this paper. We used a finetuning approach to build models to identify and classify PCL and explored the performance of various models as well as training variations and present them as a comparative in this paper. We explore techniques to deal with class imbalance, which is a rampant problem in real-world datasets by considering various class weighting techniques which work based on cost-sensitive learning. We also explore the idea of using metadata to optimize our model by adding a context that represents the target community or situation being referred to in a given paragraph.

In the future, we would like to explore other options that utilise the power of task-specific metadata. We would also like to work with other transformer-based models such as T5 (Raffel et al., 2019) and ELECTRA (Clark et al., 2020). We would also like to work on improving the ability of our model to recognise the subtle use of language which is embodied by PCL.

Acknowledgements

We would like to thank Dr Vijay Kumar Bohat and Aniruddha Chauhan for their support during writing this paper. We would also like to thank the organisers of SemEval-2022 for conducting this competition.

References

Francois Chollet et al. 2015. Keras.

Lilie Chouliaraki. 2010. Post-humanitarianism: Humanitarian communication beyond a politics of pity. *International Journal of Cultural Studies*, 13(2):107–126.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. 2019. Class-balanced loss based on effective number of samples. *CoRR*, abs/1901.05555.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. 28.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Charles Elkan. 2001. The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Conference on Artificial Intelligence: 4-10 August 2001; Seattle*, 1.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, pages 625–660.

Susan Fiske. 1993. Controlling other people: The impact of power on stereotyping. *The American psychologist*, 48:621–8.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

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. *CoRR*, abs/1907.11692.

Sik Hung Ng. 2007. Language-based discrimination: Blatant and subtle forms. *Journal of Language and Social Psychology*, 26(2):106–122.

Malte Ostendorff, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, Georg Rehm, and Bela Gipp. 2019. Enriching bert with knowledge graph embeddings for document classification.

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. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5891–5902.

Carla Pérez-Almendros, Luis Espinosa-Anke, and Steven Schockaert. 2022. SemEval-2022 Task 4: Patronizing and Condescending Language Detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Harish Tayyar Madabushi, Elena Kochkina, and Michael Castelle. 2019. Cost-sensitive BERT for generalisable sentence classification on imbalanced data. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 125–134, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. *CoRR*, abs/1905.07129.