

# UMass PCL at SemEval-2022 Task 4: Pre-trained Language Model Ensembles for Detecting Patronizing and Condescending Language

David Koleczek and Alex Scarlatos and Preshma Linet Pereira and Siddha Karakare

University of Massachusetts Amherst

{dkoleczek, ajscarlatos, ppereira, skarkare}@umass.edu

## Abstract

Patronizing and condescending language (PCL) is everywhere, but rarely is the focus on its use by media towards vulnerable communities. Accurately detecting PCL of this form is a difficult task due to limited labeled data and how subtle it can be. In this paper, we describe our system for detecting such language which was submitted to SemEval 2022 Task 4: Patronizing and Condescending Language Detection. Our approach uses an ensemble of pre-trained language models, data augmentation, and optimizing the threshold for detection. Experimental results on the evaluation dataset released by the competition hosts show that our work is reliably able to detect PCL, achieving an F1 score of 55.47% on the binary classification task and a macro F1 score of 36.25% on the fine-grained, multi-label detection task.

## 1 Introduction

Everyone has seen, experienced, and expressed patronizing and condescending language (PCL). Someone is patronizing or condescending when they communicate in a way that talks down to others, positions themselves in a superior position to the subjective group, or describes them in a charitable way in order to raise a feeling of compassion for the target person or group. Such language is seen frequently in social media or other mediums where hate is the norm. Many previous works have addressed offensive language and hate speech (Zampieri et al., 2019) (Basile et al., 2019) in such communities. However, until (Perez-Almendros et al., 2020) there has not been work in modeling PCL where 1) it is targeted towards vulnerable communities such as refugees or poor families 2) occurs in the *general media or news*. The authors note that this form of PCL is not usually meant to be harmful, and can often have good intentions, such as when the author is trying to raise awareness about an at-risk group. Yet, research in sociolinguistics has shown that regardless of the intent,

PCL has negative effects on potential vulnerable communities such as perpetuating stereotypes, reinforcing societal misbehaviors, and oversimplifying deep-rooted problems. The authors of *Don't Patronize Me! An Annotated Dataset with Patronizing and Condescending Language towards Vulnerable Communities* originally proposed a new dataset to help in detecting such cases of PCL and are currently hosting a SemEval-2022 competition: Patronizing and Condescending Language Detection (Pérez-Almendros et al., 2022).

In this paper we describe our contribution to both competition subtasks: binary and multi-label classification. Our final ensemble leverages many techniques including pre-trained language models, data augmentation, intermediate fine-tuning on related datasets, and optimizing the detection threshold to empirically maximize F1. For the binary task, our system achieves an F1 score of 55.47%, or 6.36% higher than a comparable RoBERTa-based baseline model. We achieve a macro F1 score of 36.25% on the multi-label task, or 25.84% higher than a similar RoBERTa baseline. These results show encouraging progress for the task of detecting fine-grained and linguistically subtle types of PCL, although there is still room for improvement.

## 2 Background

The Patronizing and Condescending Language Detection task contains two subtasks. In the first subtask, binary classification, the system is given a paragraph of text and it must predict if it contains *any* form of PCL. In the second subtask, multi-label classification, a system is given a paragraph of text and it must identify if it contains each of seven types of PCL (one paragraph can contain multiple categories of PCL).

### 2.1 Dataset

The binary classification task dataset contains 10,636 total labeled text samples that were selected

Text	Keyword	Country	Label
This wally of an MP just wants his name in the press , what has he ever achieved fro Southend , what did this idle wally do to get the scanner tuned on at the hospital ... zero ! He raised in parliament the fact that his was his mummies birthday , but forgot to mention the homeless families on our streets .	homeless	gb	3
Our life has completely changed from when he as an able-bodied young man running around 5000 miles an hour organising everyone . Now , he 's more disabled than anyone that he ever helped .	disabled	nz	4
The World Health Organization did not give a reason for the increase in deaths , but a provincial health official in Sindh said that the disease hit areas where poor families did not vaccinate their children .	poor-families	us	2
However , she said immigrant patients urgently need treatment and counselling from health-care providers who speak Punjabi or Hindi , and that 's what Roshni – which means light – will offer them.	immigrant	ca	1
In February 2015 , five of the men , aged 23 to 25 at the time , went to the station to protest the "" arbitrary and violent "" arrest of one of their friends from the Cova da Moura neighbourhood , known for its large population of immigrants from Cape Verde , a former Portuguese colony off Africa 's northwest coast .	immigrant	in	0

Table 1: Samples from the competition dataset

	Train	Validation	Test
<b>Positive</b>	683	111	199
<b>Negative</b>	6436	1145	1894

Table 2: Label distribution in train, validation and test splits for the binary classification subtask

from news stories from the News on the Web corpus (Davies, 2013). All the selected paragraphs are from news stories that were published between 2010 and 2018 from 20 English speaking countries. Each paragraph is assigned a label ranging from 0 (not containing PCL) to 4 (being highly patronizing or condescending). Each paragraph is also associated with one of ten keywords used to retrieve the text from the corpus. Samples from this dataset can be found in Table 1 and the distribution of positive and negative labels in each of our data splits can be found in Table 2.

The dataset for the multi-label classification task contains paragraphs from the binary classification task which were labeled as containing PCL. Each paragraph is further labeled with at least one of the seven classes: unbalanced power relations, shallow solution, presupposition, authority voice, metaphor, compassion, and the poorer, the merrier. Each is associated with a country-code and keyword as before. Additionally, each paragraph has the start and end spans of where in the paragraph the particular category of PCL is contained. Further background information about this dataset can be found in the paper published by the competition hosts (Perez-Almendros et al., 2020).

## 2.2 Baselines

Recent advances in natural language processing have enabled complex and subtle tasks to be solved by fine-tuning large, pre-trained language models (Kalyan et al., 2021). These models can also be fine-tuned on small datasets (such as only 1,000 samples) and while still achieving significant predictive power (Mosbach et al., 2020), or even with just a few samples (Brown et al., 2020). As such, the competition authors provided baseline results using a pre-trained RoBERTa model, one of the go-to models for fine-tuning for a classification task (Liu et al., 2019). They provide results of multiple baselines which help put into context the improvements made by the participating systems. For example, one such baseline is simply random guessing, or assigning each paragraph a random label with equal probability. In our work we build upon the RoBERTa baseline by applying techniques often used in machine learning. Namely, we use early stopping using a validation dataset and set hyper-parameters according to best practices discussed by (Mosbach et al., 2020). We also will experiment with a host of NLP techniques including data augmentation and other language models.

## 3 Subtask 1: System Overview

Considering the challenge of the task and the scarcity of positive examples of PCL, our approach was to experiment with methods focused on enriching our small dataset, such as data augmentation and external datasets. We started with fine-tuning a pre-trained RoBERTa (Liu et al., 2019) model to use as a foundation. RoBERTa has been shown

to be very powerful for a range of NLP tasks such as text classification, while also being straightforward to implement and train using libraries such as HuggingFace (Wolf et al., 2019). We use its architecture unchanged with pretrained weights downloaded from HuggingFace. We add a standard binary classification head on top of RoBERTa’s pooled outputs. For preprocessing input into the model, we use HuggingFace’s auto-tokenizer. In the following subsections, we detail the additional methods and modifications to the RoBERTa baseline. The models used in our final ensemble use a mix of these methods and the exact ensemble methodology is described in section 3.6.

### 3.1 Pre-trained Language Models

In addition to RoBERTa, we use MPNet (Song et al., 2020). This model inherits the pre-training advantages of both Masked Language Modeling (Devlin et al., 2018) and permuted language modeling (Yang et al., 2019) under a unified view which splits tokens in a sequence into non-predicted and predicted parts. Experimental results show that MPNet outperforms previous models like BERT, XLNet and RoBERTa by 4.8, 3.4 and 1.5 points respectively on GLUE tasks. In our work, we fine-tune MPNet using the hyperparameters suggested by the authors, which can be found in Appendix A.

### 3.2 Intermediate Fine-Tuning

We leverage several external data sources which are related to our task of PCL detection. (Phang et al., 2018) originally proposed *Supplementary Training on Intermediate Labeled Data Tasks* (STILTs), which we refer to as intermediate fine-tuning. They observed that by 1) starting from a pre-trained language model such as BERT, 2) further training it on an intermediate, labeled task, and 3) finally fine-tuned on the target task; improved final performance on a variety of NLP tasks. (Pruksachatkun et al., 2020) and (Vu et al., 2020) provide further insight into tasks that are well suited for intermediate fine-tuning. We used these insights to develop four independent binary classification tasks on Stereoset (Nadeem et al., 2020), BiasCorp (Onabola et al., 2021), Social Bias Inference Corpus (SBIC) (Sap et al., 2020), and TalkDown (Wang and Potts, 2019) datasets, and used them for intermediate fine-tuning. Two of these tasks, SBIC and Biascorp, are used in our final ensemble, while the other two were left out due to having no performance uplift on our task’s test set.

**BiasCorp** is a dataset containing about 43,000 labeled examples of racial bias from Fox News, BreitbartNews and YouTube. Each text has 3 bias ratings from 0 to 5 and corresponding annotator confidence scores from 1 to 10. To turn this into a binary classification task, we use a similar approach from their paper by first computing a confidence weighted score, and then we threshold this score where scores above 1 are labeled as positive examples.

The **Social Bias Inference Corpus** is a labeled dataset of over 125,000 examples of social media posts from Reddit, Twitter, and hate sites like Stormfront. For our binary classification task, we use the provided offensiveness rating of a post, and use a threshold of 0.3 for positive examples.

### 3.3 Sentence-Level Features

Transformers are generally able to vastly outperform regression on engineered features. However, in some text labeling tasks, such as essay scoring, it has been shown that engineered features can be used in tandem with Transformer output to improve performance (Uto et al., 2020). This can be achieved simply by concatenating a vector of features  $f_n$  to BERT’s CLS vector. An extended classification head will now take the extended vector as input, and similarly project to the final prediction. The classification head, including the added weights, is trained along with the fine-tuning of BERT.

Our set of features was inspired by (Uto et al., 2020), but we excluded the readability metrics because they are not as relevant for our task. Specifically, for text sample  $x_n$ , we calculate *the number of words, number of sentences, number of exclamation marks, question marks, and commas, average word length, average sentence length, the number of nouns, verbs, adjectives, and adverbs, and the number of stop words*. Each one of these values is used as a separate dimension of  $f_n$ . We leveraged the spaCy Python library (Honnibal and Montani, 2017) to calculate these values. We also included the feature calculation as a data pre-processing step, since calculating the features at run-time would be too costly.

### 3.4 Original Labels

Simply collapsing the 0-4 label in the dataset to a binary value results in a loss of relevant information. To combat this, we translate each label to a *probability* of a sample containing PCL. Specifi-

cally, we perform the following mapping: 0 - 0.0, 1 - 0.25, 2 - 0.5, 3 - 0.75, 4 - 1.0. We refer to these probabilities as the *original labels*. We adapt our loss function for this reformulated learning setup by computing the binary cross-entropy loss between the predicted probability and the original label.

### 3.5 Backtranslation

Data augmentation for NLP tasks is becoming increasingly popular with a vast variety of existing techniques. Data augmentation refers to any strategy aimed at increasing the amount of data available for training, by only leveraging the existing training set or domain knowledge about the task (Feng et al., 2021). (Sennrich et al., 2015) is a model-based technique for data augmentation where the original text is translated into a desired language and then back to the original language to rephrase it. This process can introduce a different style of sentence for the same meaning. We have used French as our intermediate language for backtranslation on our data. To implement backtranslation for our task, we used the MarianMT model (Tiedemann and Thottingal, 2020) provided by HuggingFace to convert between French and English.

### 3.6 Ensemble Model

Ensembling is a technique that leverages multiple trained models to improve task performance on the test set. For our final ensemble of three independently trained models, we used a relatively simple approach. Given a set of models  $M$  and a test dataset  $D$ , let  $\hat{y}_{ni}$  be  $M_i$ 's estimate of the probability that text sample  $x_n$  is a positive sample. This value is taken directly from  $M_i$ 's softmax layer output. Each label is then assigned a final positive probability of  $\hat{y}_n = \frac{1}{|M|} \sum_{i=1}^{|M|} \hat{y}_{ni}$ . We use this augmented softmax score to make the final prediction for each label in the test set.

### 3.7 Choosing an Empirically Optimal Threshold for F1 Score

The F1 score is the metric used for evaluation in the competition and it is defined as the harmonic mean of precision and recall:

$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

Note that the output of our transformer models for binary classification is the softmax of a final linear layer. In order to compute the F1 score, those

continuous outputs must be binarized. Typically, that is done via a simple transformation:

$$f(x, threshold) = \begin{cases} 1, & x \geq threshold \\ 0, & x < threshold \end{cases} \quad (1)$$

, where  $x$  is the softmaxed model output and threshold is a value commonly set to be 0.5. Since the model is optimized using cross entropy loss, it is unclear if 0.5 is the best threshold to use in order to maximize F1 score. Since the F1 score is a *harmonic mean*, we would like to balance precision and recall in order to achieve the highest score. Roughly, precision can be increased (and recall decreased) by increasing the threshold, and vice versa by decreasing the threshold.

In order to find the best threshold, we use a simple, brute force algorithm to determine the best threshold on the validation data set. The algorithm takes vectors  $y_{true}$  and  $y_{out}$ , which are the true binary labels and the normalized model output, respectively. For each model output, we treat it as a *possible threshold* and compute the F1 score using the true labels and  $y_{out}$  binarized via  $f(y_{out}, model\ output)$  (assume  $f$  operates on vectors). The algorithm outputs the threshold that achieved the highest F1 score.

---

#### Algorithm 1: Brute-force F1 Threshold

---

```

Input: y_true, y_out
best_score ← 0
best_threshold ← 0.5
for threshold in y_out do
    bin_y_out ← f(y_out, threshold)
    score ← f1_score(y_true, bin_y_out)
    if score > best_score then
        best_score ← score
        best_threshold ← threshold
return best_threshold

```

---

## 4 Subtask 2: System Overview

Our system for task 2, multi-label classification, adapts many of the techniques we used in the binary classification task. Similar to our task 1 system, we leverage RoBERTa as a base model for further fine-tuning. However, the target is now a vector of binary labels. Another way to view the problem is as 7 binary classification tasks, one for each category of PCL. Our final ensemble includes models treating the problem in this way, and also training one multi-label classifier directly. We also

independently train a binary classification model to generate a prediction for each class using the same model setup from the binary task. In either case, the final ensemble of models was computed identically to the binary classification task by treating each category independently. The sigmoid function was applied to each model output before creating the ensemble.

#### 4.1 Data Processing

The dataset for the multi-label classification task contains only paragraphs from the binary classification task which were labeled as containing PCL. This creates a disconnect between the data that would be seen during evaluation (i.e. there are samples without any category of PCL). To handle this discrepancy in data distributions, we take every non-PCL paragraph from the binary classification task dataset and add it to the multiclass dataset with a label vector of all 0s.

### 5 Experimental Setup

We start by splitting our dataset into a train and test set according to the split provided by the competition authors, which is 80% train and 20% test data. From the training split, we take 15% to use as a validation split. Table 2 shows the distribution of positive and negative labels within the train, validation and test splits.

Using our validation split, for each experiment, we perform a hyperparameter sweep by varying the peak learning rate  $\in \{3e-5, 2e-5, 1e-5, 5e-6\}$ . We use early stopping where the model is evaluated at the end of each epoch, and if for 10 consecutive epochs performance does not improve, we stop training and save the model from the epoch with the highest validation F1 score. For final evaluation on the test set we use this model that achieved the highest F1 score on the validation set. We choose a batch size that maximizes GPU usage, and use the default maximum sequence length of each model. A listing of the full hyperparameters used can be found in Appendix A. Random seeds for Python, numpy, and PyTorch were fixed at 221 for all experiments.

For our final results and submission to the competition, we use the validation split for early stopping and train including the test dataset. All models were trained with a learning rate of  $2e-5$  unless otherwise noted. We submitted two identical systems, only differentiated by the random seed used. All

Binary	Description	P	R	F1
Model 1	RoBERTa, intermediate fine-tuning, features	67.95	47.75	56.08
Model 2	MPNet, original labels	65.95	55.85	60.48
Model 3	RoBERTa, features, backtranslation	61.11	49.55	54.73
Ensemble		62.04	60.36	61.19

Table 3: Subtask 1 - Validation split scores for each of our models that composed the final ensemble. The scores for each model use the default threshold, while the ensemble’s threshold is optimal for the validation split.

Multi-label	Description	P	R	F1
Model 1	RoBERTa, features, backtranslation	42.90	30.52	35.01
Model 2	RoBERTa, backtranslation, LR= $1e-5$	46.53	36.18	39.13
B4MC	Binary model 1 for each category	2.51	100.0	4.83
Ensemble		49.38	35.72	40.50

Table 4: Subtask 2 - Validation split scores for each of our models that composed the final ensemble.

results shown in Tables 3 and 4 are the random seed that performed better.

Our experiments were run a system with the following hardware; GPU: NVIDIA RTX 3090, CPU: AMD 5950X, RAM: 64GB 3200Mhz, SSD: 4TB SATA.

### 6 Results

Submitted systems were evaluated on the F1 score (macro F1 score for the multi-label task). We built several models that each combine different methods. Observe that in Tables 3 and 4 each model has a varying degree of balance between precision and recall. This diversity in outcomes and methodology between individual models helps our final ensemble achieve better overall results than any individual model. Our best ensemble for subtask 1 placed 22nd out of 78 and our best ensemble for subtask 2 placed 18th out of 49.

For subtask 1, our best model on the validation split was the one that used MPNet as a base and used the original labels during training. Model 1 used RoBERTa as a base and was first trained using intermediate fine-tuning, before being trained on

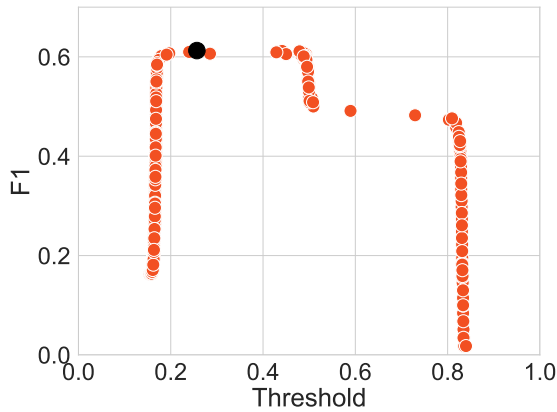


Figure 1: Plot of the resulting F1 score when using each validation softmax score (unbinarized) as the detection threshold.

the final dataset using sentence level features. Intermediate fine-tuning was done by first training using the SBIC task for 5 epochs, followed by training for 5 epochs on the Biascorp task. Each intermediate task was trained with a learning rate of  $5e-6$ . Finally, model 3 used RoBERTa as a base and was trained with sentence level features and additional backtranslated samples.

For subtask 2, our best model used RoBERTa as a base and was trained with backtranslated samples and a learning rate of  $1e-5$ . Model 1 was similar except it used sentence level features and our default hyperparameters found in Appendix A. Binary for multi-category (B4MC) was 7 individual models (one for each category), trained using the same methodology as our best binary model, model 1.

## 6.1 Thresholds

We use Algorithm 1 to find an optimal F1 threshold on the validation split in order to increase the F1 score for our final ensemble. Analysis of this optimization for our binary task ensemble is in Figure 1. Based on that plot, we set our prediction threshold to be 0.32, about in the middle of the range of thresholds that achieved the highest F1 scores. Notice that if we had picked a threshold of 0.5, we likely would have achieved a lower score due to the drop in scores around that mark.

We performed similar analysis for subtask 2. The thresholds we choose were 0.5, 0.31, 0.49, 0.27, 0.29, 0.21, and 0.4 for the categories unbalanced power relations, shallow solution, presupposition, authority voice, metaphor, compassion, and

the poorer the merrier, respectively.

## 6.2 Error Analysis

To further analyze how our model compares to the baseline, we examine samples from the test set where 1) the baseline made an error, but our model made the correct prediction, and 2) where both the baseline and our model made the incorrect prediction. We examine false positives and false negatives in both cases.

In Table 5, we can see that the baseline’s false negatives tend to be more poetic examples. They paint vivid, dramatic scenes, and use strong adjectives like "shocking" and "hopeless". This indicates that our model may have picked up on this poetic language signal as an indicator of PCL.

In the baseline’s false positives, we see phrases that may be indicative of PCL, such as "in need of food", "treat men and women differently", and "being disabled". However, in context, these phrases are not condescending to the subjects of the statements. It’s possible that the baseline was simply acting on the presence of these phrases, while our model was able to identify the lack of other supporting signals as evidence that the sentences were not PCL.

In Table 6, we can see where both the baseline and our model failed, where there are a variety of possible issues. The first two false negatives are only identifiable as PCL given their subtext (assuming the perspective of refugees and creating a division between disabled and non-disabled people), rather than obvious choices of words. While the third false positive uses flowery language (a possible PCL indicator), the identifying piece, "people from poor families have more perseverance...", uses a positive sentiment. The models seem to struggle with the "poorer the merrier" category, which this is an example of. It’s possible that the model is overwhelmed with more negative sentiments from other categories so it tends to correlate that too strongly with PCL.

The false positives all use the kinds of words that seem to be common in the actual positive examples, such as "plight", "challenges", "disadvantaged", and "vulnerable". What the models seem to fail at is distinguishing between pointing out a group’s disadvantage and using that disadvantage to patronize the group.

Overall, the models seem to be very active on words and phrases that might indicate PCL. While

Text	Label	Baseline Pred.	Ensemble Pred.	Categories
real poverty of britain : shocking images of uk in the sixties where poor really meant poor these hard-hitting photographs offer a glimpse into the harrowing day-to-day for poor families living in britain during the sixties .	pos.	neg.	pos.	Authority Voice, Compassion
rather sad . good set of pictures that tells a tale of survival , subsistence living , and hopeless nature of life in the tribal societies . exploiting an unexpected geo-political bonanza is a temporary relief that is not sustainable . education and economic development seem miles away .	pos.	neg.	pos.	Presupposition, Metaphors, Compassion
as a family , my father was a policeman and he used to come home with food ( monthly ) , and my mother used to pack small parcels and we used to give them to the poor families .	pos.	neg.	pos.	Unb. power rel., Shallow solu.
mrs charo said some people were injured while escaping the floods . she added that they are now in need of food , clothing and clean drinking water .	neg.	pos.	neg.	-
she added : i would also like to carefully point out that the issue was not her religious beliefs , but rather it is about choosing to treat men and women differently by shaking the hands of women but not men .	neg.	pos.	neg.	-
i realised that it was not impossible to achieve anything in the world despite being disabled .	neg.	pos.	neg.	-

Table 5: Samples where the ensemble improved over the baseline.

Text	Label	Baseline Pred.	Ensemble Pred.	Categories
many refugees do n't want to be resettled anywhere , let alone in the us .	pos.	neg.	neg.	Presupposition
the law stipulated 21 rights of the disabled persons . the disabled persons must get the national identity cards and be listed in the voters roll . even they will be able to contest in the polls .	pos.	neg.	neg.	Unb. power rel.
however , this success story is not uncommon . it often happens that people from poor families have more perseverance to fight tooth and nail in business than children of rich parents who are used to get everything they want with ease . people without a strong spirit will quickly break down and drop out from the competition .	pos.	neg.	neg.	Presupposition, Metaphors, The p., the mer.
your personal leadership has been critical to addressing the plight of the rohingya who fled to safety in your country . i thank you for all you have done to assist these men , women and children in need , he wrote in the message .	neg.	pos.	pos.	-
she also praised the efforts to stabilise the lives of refugees , by providing for their basic needs and helping them overcome their challenges , while stressing that supporting refugees is an ongoing part of the uae 's humanitarian directives , and the country has taken the responsibility to evaluate their needs and provide them with a variety of urgent and essential aid .	neg.	pos.	pos.	-

Table 6: Samples where both the baseline and ensemble were incorrect.

our best model appears able to deal with these features in somewhat intelligent ways, the main downside is that it cannot place the samples in a global context. Simply using certain adjectives to describe a certain group of people does not necessarily make a statement condescending. It depends on exactly which adjectives, which group, and what other context is in the statement. As humans, we have two main advantages in this task: 1) we understand the

struggles and stereotypes of disadvantaged groups, and 2) we also can perform deductive reasoning to determine the underlying meaning of a phrase. Possibly remedies for our model might be 1) providing many more examples, so it can build the kind of context needed to make these distinctions, and 2) to enhance the model with ways of reasoning about the problem.

## 7 Conclusion

We developed a system to detect patronizing and condescending language by media towards vulnerable communities. We combined a multitude of NLP techniques in our final ensemble. Our system leveraged backtranslation, sentence level features, intermediate fine-tuning, and pre-trained language models. We also proposed multiple training techniques and optimizing the F1 threshold. These improvements resulted in our system placing in the top 25% for subtask 1 and the top 40% for subtask 2.

## Acknowledgements

This work began as part of an advanced natural language processing course at the University of Massachusetts Amherst. We thank the teaching assistants Katherine Thai and Tu Vu and the instructor Mohit Iyyer for their wonderful teaching and feedback.

## References

- Valerio Basile, Cristina Bosco, Elisabetta Fersini, Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, et al. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.
- 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.
- Mark Davies. 2013. Corpus of news on the web (now): 3+ billion words from 20 countries, updated every day. Available online at <https://www.english-corpora.org/now/>.
- 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*.
- Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Edward Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.
- 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. *arXiv preprint arXiv:1907.11692*.
- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*.
- Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.
- Olawale Onabola, Zhuang Ma, Yang Xie, Benjamin Akera, Abdulrahman Ibraheem, Jia Xue, Dianbo Liu, and Yoshua Bengio. 2021. hbert+ biascorp—fighting racism on the web. *arXiv preprint arXiv:2104.02242*.
- Carla Perez-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.
- Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.
- Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R Bowman. 2020. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628*.
- Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *ACL*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tiejun Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.



- Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.
- Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. [Neural automated essay scoring incorporating hand-crafted features](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordani, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across nlp tasks. *arXiv preprint arXiv:2005.00770*.
- Zijian Wang and Christopher Potts. 2019. Talkdown: A corpus for condescension detection in context. *arXiv preprint arXiv:1909.11272*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. [SemEval-2019 task 6: Identifying and categorizing offensive language in social media \(OffensEval\)](#). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

## A Fine-tuning Hyperparameters

Hyper-parameter	Fine-tuning
Learning Rate	2e-5
Batch Size	16
Weight Decay	0.1
Epochs	Early stopping
Learning Rate Decay	Linear
Warmup Ratio	0.06

Table 7: Hyper-parameters for fine-tuning RoBERTa and MPNet models.