NLP Coursework - PCL Classification

Repository: https://gitlab.doc.ic.ac.uk/ahf119/nlp_cw

Jean Durand
Imperial College London
jd123@ic.ac.uk

Hugo Frelin
Imperial College London
ahf119@ic.ac.uk

Igor Sadalski
Imperial College London
is1220@ic.ac.uk

1 Introduction

In this paper, we will explore the use of Natural Language Processing (NLP) in detecting patronising and condescending language (PCL) in a binary classification problem. The interest in utilising NLP to automatically detect harmful language has increased rapidly in recent times. There are successful models for detecting language such as hate speech; transformer-based models can achieve an F1 score of 96% (Saleh et al., 2023). However, PCL remains a challenging issue. This can be attributed to the inherent subtleties of such language, making it hard even for human annotators to classify data correctly. Furthermore, the use of PCL is not always conscious, and is many times used in good-will. In order to encourage research within this field, Pérez-Almendros et al. (2020) created the 'Don't Patronize me!' dataset, which is used for PCL classification. The work described in this paper explores how one can use this dataset to train PCL classification models. We compare the performance of transformer-based models, and explore techniques such as ensembling and data augmentation, with more traditional NLP models like logistic regression and bag-of-words (BoW) in detecting PCL.

2 Data Analysis of the training data

The 'Don't Patronize me!' dataset contains 993 paragraphs that are labeled as containing patronising text and 9476 paragraphs that does not contain patronising text. First, we will perform a quantitative analysis of the class labels, followed by a more qualitative assessment of the data.

This dataset is inbalanced as it has more paragraphs that do not contain patronising text. Thus, it is interesting to analyse how these labels correlate with other features in the input data. Figure 1 displays the distribution of sequence lengths for text sections with and wihout PCL, as well as the averages. Even though the mean is slightly higher for sequences with PCL, the distributions are very similar.

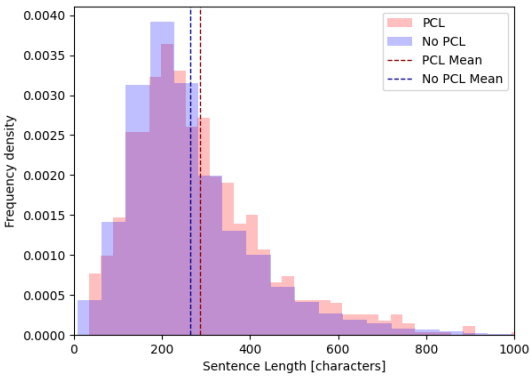


Figure 1: Figure displaying the distributions of sequence length, measure in characters, with and without PCL.

Furthermore, 'Don't Patronize me!' was created by searching for articles containing potentially vulnerable communities. Now, Figure 2 displays the proportion of articles with PCL for each keyword compared to the proportion of the entire dataset. The proportion of samples found using the keywords 'homeless', 'in-need' and 'poor-families' with PCL is roughly 4-5 times higher than the samples with keywords 'migrants' or 'im-migrants'. Thus, the dataset is skewed towards certain classes. This is a bias in the dataset that one should be aware of, as it can induce biases further down the development pipeline.

Each paragraph in the dataset was labeled by two annotators as either containing *0 - No PCL*, *1 - Borderline PCL* or *2 - Contains PCL*. This was then rolled up to a value between 0 and 4, where 0 means both annotators said *No PCL* and 4 means both annotators said *Contains PCL*. A paragraph

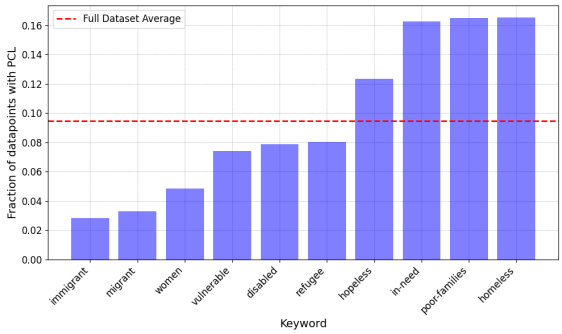


Figure 2: Figure displaying the fraction of text sequences with PCL for different keywords compared to the full dataset fraction.

was labeled as patronising if it contained a rolled up value of greater or equal to 2. Now, only 391 of the 993 paragraphs that were labeled as containing patronising text were labeled as 4. This already speaks to the subjectivity and difficulty of the task of labeling text that contains PCL as not even the annotators always agreed on what constitutes as PCL. As an example, the following paragraph is labeled as containing PCL:

> "Bond went out of his way to help the less fortunate, often going on the road with Kim to take food to the homeless."

The quote above (paragraph id 9518 in the dataset) is labelled as 4, meaning that both annotators agreed that it contains patronising text. Additionally the categories of PCL are Compassion, Shallow Solution or Unbalanced Power Relation. Most people will struggle to determine where exactly PCL occurs, and how to dissect the paragraph into the different PCL categories. The untrained eye might not even detect any PCL at all. It is clear that Bond is acting in an ethically defendable manner, but it is the nuances about how it is described which makes it patronising and condescending.

Furthermore, the following sequence (paragraph id 5132 in the dataset):

> "Focus on the homeless"

is labeled as 1, meaning that one annotators thought it was a borderline case. The contrast between the two examples demonstrates how subtle differences in phrasing can significantly affect the perception of condescension. The second sentence distances itself more from the power relations between the rich and the poor and does not demonstrate any intent.

Now for an NLP model to perform well at this task it must not only gain an understanding of the sentence, but of the power dynamics conveyed within it. It is a very challenging task that even humans struggle to agree on.

3 Modelling

We addressed the problem by splitting the PCL dataset into a training set, a validation set and a test set. The test set was the official validation set used by Pérez-Almendros et al. (2020). We used the validation set to do the hyperparameter tuning and the investigation of further model improvements and we used the test set when we benchmarked our final model against existing approaches. In all these datasets if the text had a PCL label of greater or equal to 2, we considered the text as patronising. We choose binary classification instead of multi class-classification as there was large data sparsity for some labels making it hard to reliably learn some of the classes.

3.1 Modelling choices and hyperparameters

Our solution was based on the pre-trained transformer models BERT and RoBERTa (Devlin et al., 2018; Liu et al., 2019). We used these pre-trained transformers and added a dropout layer and a linear probing layer to finetune these models for the downstream classification task. We considered the *bert-base-cased*, *bert-base-uncased* and *roberta-base* models from the huggingface transformers library (Wolf et al., 2020). We wanted to see if the tone introduced by capitalisation affects the ability of the model to pick up on PCL. However, since capitalisation does not significantly change the meaning of sentences in English, we did not expect to see a big difference in the cased and uncased BERT model.

The training of our model worked as follows. We used both the training set and the validation set in the algorithm. We trained the model for 7 epochs, but we implemented early stopping based on the F1 score on the validation set. We used a batch size of 32 and evaluated the model at every 100 steps. If the model did not improve the validation F1 score for 3 consecutive evaluations, the algorithm was stopped and the checkpoint with the highest validation score was saved.

In order to optimise the models, we conducted a hyperparameter search training 60 different combinations of pre-trained models, learning rates,

learning schedules and data augmentation techniques. The full results of which are shown in the Figure 9 in the Appendix. Additionally, the optimal learning rate was found to be 0.00005 combined with a linear learning rate schedule.

3.2 Further model improvements

As previously mentioned, the dataset used has an inbalance inbetween the labels. Considering this we decided to increase the weight of positive examples by using an experimentally derived factor of 2. The loss function we used was a weighted cross-entropy loss function where we assigned a weight of 2 to the positive class.

Secondly, we used various data augmentation techniques to upsample the minority class. Data augmentation has been shown to be a very efficient tool in boosting performance in NLP classification. As suggested by previous work we thus implemented and tested: word insertion, word substitution, word deletion, word swapping and back translation (Wei and Zou, 2019; Yaseen and Langer, 2021).

Both word insertion (inserting words into random places in the sequence) and word substitution (substituting random words) was done contextually using a BERT model. The idea behind this is that it generates more plausible data augmentation than a traditional look-up dictionary. We also implemented deletion and swapping. Deleting randomly deletes words from sequences, and swapping randomly swaps adjacent words. Lastly, the back-translation was done by translating the sequences to a second language, and then translating it back.

In our third modelling improvement technique we explored the ensembling technique known as stacking. We stacked three models, and experimented with two different conditions on the of the model outputs for classification. The three ensembled classifiers were based on the three different pretrained models, RoBERTa, BERT - uncased and BERT - cased. The three models were also trained on different data augmentations, with the hope of them learning different patterns of PCL.

3.3 Selected results

Figure 3 shows how the learning schedule improves the training of the model. The linear learning rate scheduler increases the learning rate from a low rate to a higher rate linearly in the initial stages of the training.

As shown in the leftmost graph, with the linear learning schedule, both training and validation losses decline more consistently, mirroring a steady rise in the F1 score. Conversely, as the rightmost graph illustrates, without a learning rate scheduler, the losses vary significantly. This indicates that a learning rate scheduler contributes to more stable training.

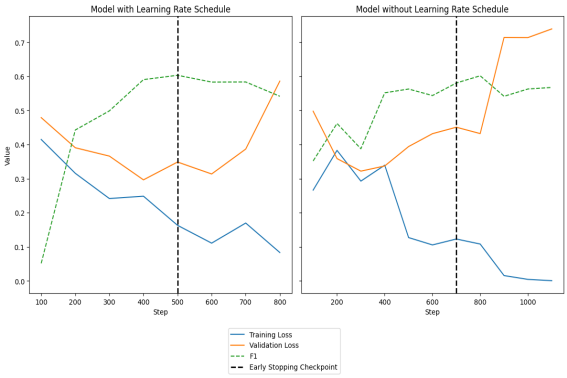


Figure 3: Loss curve for a ROBERTA model that used a learning rate scheduler in comparison to a ROBERTA model that does not use a learning rate scheduler

Figure 4 displays the results of the RoBERTa-based classifier on the validation data which has been upsampled using the different augmentation techniques described. For this task, the best performing augmentation technique was contextual insertion. However, having no augmentation scored almost as well. It is also evident that using back translation results in the poorest performance. Data augmentation not improving model performance in all cases can be due to various factors specific to this task. Given its subjective nature, the training data inherently contains a lot of noise, which could be amplified by augmentation techniques. The method that improved performance, contextual insertion, preserves the original text and its sequence, merely adding extra words. This suggests that for this particular task, precise phrasing is crucial, and altering it could eliminate important signals in the training data.

Figure 5 show how the models perform individually, as well as how different ensembling methods, majority vote and logical or, perform. In each case the model hyperparameters that performed the best on the validation data was chosen. It shows how even though the RoBERTAa based classifier has the highest validation score, the majority vote ensemble generalises better and has the highest F1 score on the test set, which means the

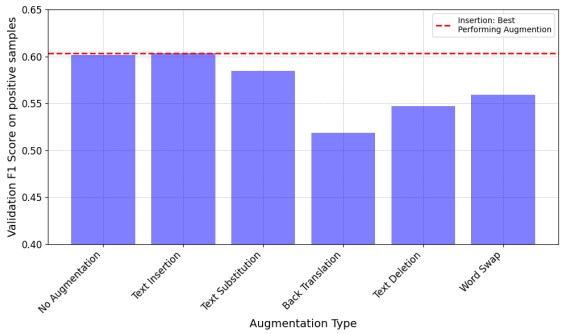


Figure 4: Bar plots comparing the validation F1 score of the Roberta based classifier when trained on data with different augmentations.

model is more robust. Using logical OR ensembling makes the recall higher, and the precision lower in comparison to the majority vote method. Thus, one can tweak the behaviour of the ensembled models rather easily dependent on the need and requirements of the downstream tasks.

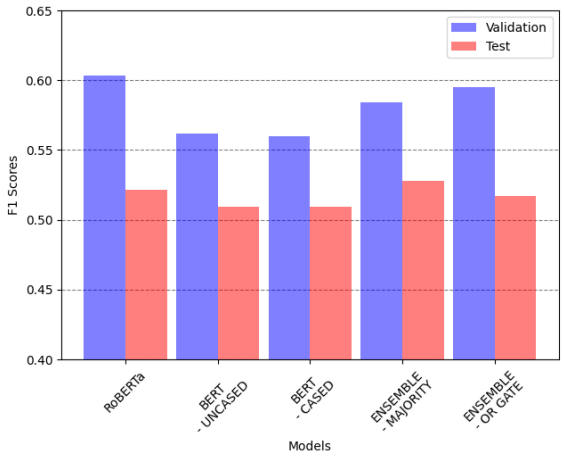


Figure 5: Comparison of F1 Scores of 3 different models (RoBERTa, BERT cased and uncased) and two different ensembling techniques (majority and OR-gate) on test and validation datasets

3.4 Model Benchmarking

In order to benchmark our models, we trained and tested a Bag of Words and a Logistic Regression classifier. To improve performance of these algorithms for we used the data augmentation techniques described in section 3.2. Instead of using a simple counter for Bag of Words and Logistic Regression we tested Term Frequency-Inverse Document Frequency (TF-IDF), a statistic reflecting the importance of a word in a document. We implemented it for the logistic regression as it improved the performance of the model. The test set F1 score of the baseline models, as we all as the test set F1 score of the ensembled models, is presented in Table 1.

Metric	Ensemble Majority Vote	BoW Counter	Logistic Regression TF-IDF
F1	0.53	0.30	0.35

Table 1: Performance on the official validation set, used by us as a test set (Pérez-Almendros et al., 2020)

The following quote (paragraph id 4494) is labelled as containing PCL

> "We shall remember him for the immense contribution he made to the many vulnerable sectors of humanity, women, children, orphans, the disabled and refugees"

The category of PCL is Unbalanced Power Relation and it was labelled correctly by our majority vote ensembling model, but incorrectly by the best performing baseline model, the logistic regression. The PCL of this sequence is conveyed in the power dynamics between the 'him' and the marginalised groups it is referring to, and is highly contextual. Thus, a count of the words which is independent of the ordering of them lose all the information which makes it PCL and thus our simple baseline model struggles to label it correctly. The transformer model is better at picking up on this contextual importance.

4 Analysis

To what extent is the model better at predicting examples with a higher level of patronising content?

In order to determine to what extent the model is better at predicting high levels of patronising content, we looked at the rolled up PCL value that was given to the piece of the text by the two annotators. Thus 0 is the lowest level of patronising content and 4 is the highest level of patronising content. Figure 6 shows how often the model predicted that the text contains patronising content for each of these labels in the test set. We can see that the model does well at predicting PCL correctly in the two extreme cases. When the label of the text was 0, in only 3% of the paragraphs did the model incorrectly predict that the text contains patronising content. On the other hand, when the label of

the text was 4, the model correctly predicted that the text contains patronising content 66% of the time. Interestingly enough, the model does not do well at distinguishing between label 1 and label 2. This shows that the model is still not good at picking up on the subtle nature of patronising content and finds it difficult to distinguish between the borderline cases.

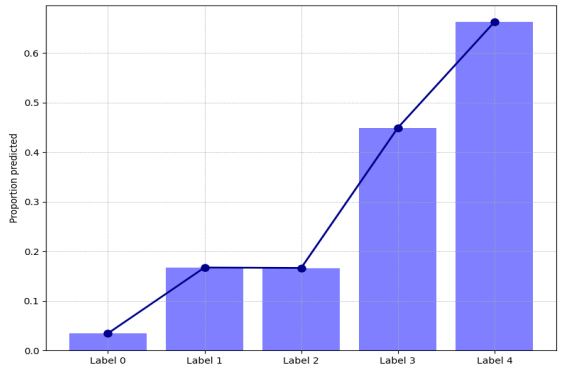


Figure 6: Proportion of PCL predicted per PCL label. Samples with a label greater than one is considered PCL.

How does the length of the input sequence impact the model performance?

Figure 7 depicts how the word length of the text impacts the test F1 score for each quantile of text length. Comparing with the averaged F1 score of the model of 0.53 we can see that sequence length does not have a significant influence on the model performance. This corresponds well to the analysis of Figure 1,a where it was shown that the sequence length distribution of both the PCL and the no PCL classes is similar. Text length alone is not enough to distuinguish between text that contains PCL and text that does not contain PCL.

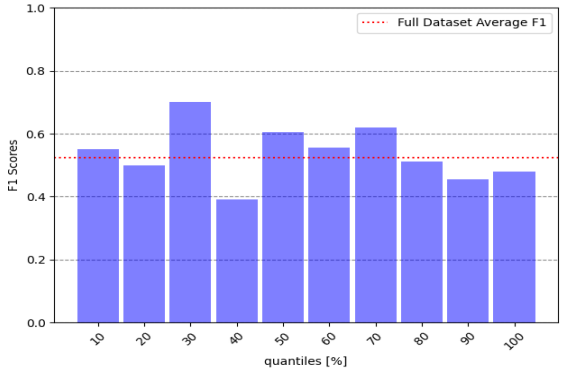


Figure 7: Model performance as function of sequence length, sectioned by quantiles, for ensemble-majority model on test set

To what extent does model performance depend on the data categories?

Figure 8 displays the test F1 score on the positive class for different data categories. As evident, the majority vote ensembled model performs much better on categories such as 'in-need', 'homeless' and 'vulnerable' than the categories 'immigrant' and 'refugee'. One explanation of this is the data set distribution of PCL for each category, displayed in Figure 2. There is a clear pattern that the more instances of PCL the category has in the data set, the more accurately the category will classified. Biased datasets leading to biased predictions is a common issue within machine learning, and this is a clear case of it. Furthermore, it is also hard to know what the true underlying distrubution is. Some categories will have a larger degree of PCL than others.

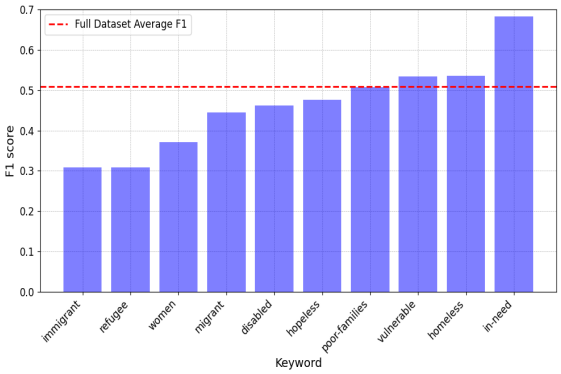


Figure 8: The test F1 score for different data categories.

5 Conclusion

In this report we investigated the use of various NLP in the detection of PCL in a binary classification problem. We found that finetuning pre-trained language models for the classification task outperformed existing NLP benchmarks. The model that performed the best was an ensemble method that combined a BERT-cased, BERT-uncased and a RoBERTa model with different augmentation techniques. However, we have noticed that our model picks up on a bias in the dataset where it predicts certain classes more accurately than others. Thus, next steps would explicitly handle this bias, by sampling the training data so that the classes have equal distributions. Additionally, we can extend this problem by predicting PCL on a class-level rather than a simple binary classification problem. This will give a more nuanced view of the model performance.

References

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*.

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*.

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. *arXiv preprint arXiv:2011.08320*.

Hind Saleh, Areej Alhothali, and Kawthar Moria. 2023. Detection of hate speech using bert and hate speech word embedding with deep model. *Applied Artificial Intelligence*, 37(1):2166719.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Usama Yaseen and Stefan Langer. 2021. Data augmentation for low-resource named entity recognition using backtranslation. *arXiv preprint arXiv:2108.11703*.

A Appendix

Parameter	Values
Learning Rate (LR)	0.00001, 0.00005
Learning Schedule	linear, constant
Augmentation Strategies	augment0: None augment1: Inserted augment2: Subbed augment3: Back Translated augment4: Deleted augment5: Swapped

Table 2: Summary of hyperparameter configurations.

Metric	Value
f1-score	0.56
learning_rate	0.00001
learning_schedule	linear
augment	substitution

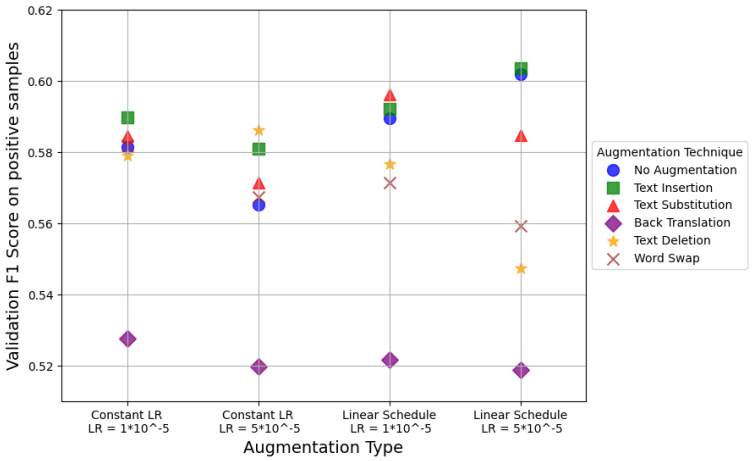
Table 3: Hyperparameters for best BERT-Base Cased model

Metric	Value
f1-score	0.6
learning_rate	0.00005
learning_schedule	linear
augment	inserted

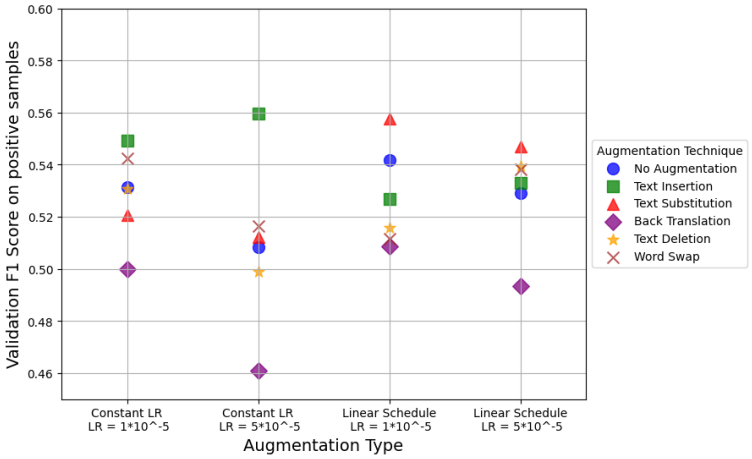
Table 4: Hyperparameter for best RoBERTa model

Metric	Value
f1-score	0.56
learning_rate	0.00005
learning_schedule	linear
augment	swapped

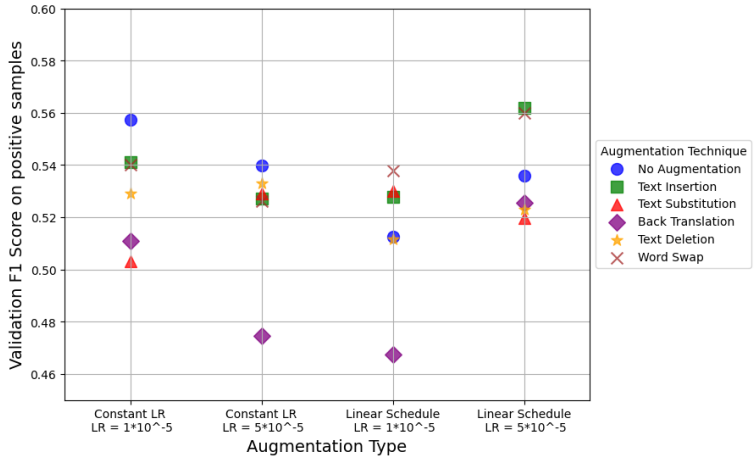
Table 5: Hyperparameters for best BERT-based Uncased model



a) Results for RoBERTA based classifier



b) Results for cased BERT based classifier



c) Results for uncased BERT based classifier

Figure 9: Results of hyperparameter search for the three pre-trained models experimented with, and used in ensembling.