A BERT-Based Transfer Learning Approach for Multi-lingual Hate Speech Detection and Text Classification

Hafez Ghaemi
s289963

Juan José Márquez Villacís
s287313

Ionut Cosmin Nedescu
s292495

***Abstract*—The spread of hateful and offensive content is a rising concern for social media platforms and a substantial amount of resources are being spent to address it. Automated techniques based on machine learning (ML) and natural language processing (NLP) offer promising solutions to the problem of hate speech identification. The advent of transformers was a milestone in the history of NLP and numerous models have been proposed based on these powerful and efficient units to address various tasks, including hate speech detection. In this project, we investigate and reproduce one of these works that introduced fine-tuning strategies of the pre-trained BERT (Bidirectional Encoder Representations from Transformers) model, and applied them to hate speech datasets that include annotated tweets. The most successful strategy based on the F1-score metric was the one using a convolutional neural network (CNN) applied to transformer embeddings. After this analysis, we also evaluated the performance of the proposed strategies on two other text classification problems for sentiment analysis and clinical trial eligibility detection. Furthermore, we modified the models to be compatible with multi-lingual text classification, and applied them on a task for multi-lingual hate-speech detection in three Indo-European languages.**

***Index Terms*—Natural Language Processing (NLP), Transformer encoder, BERT, hate speech detection, text classification, social media**

I. Problem Statement and Introduction

The social networking platforms, such as Twitter, Meta (Facebook), and Reddit have enabled people to express their opinion and share information in a manner that was not possible before. Although these platforms foster many constructive conversations, they have also proved to have multiple drawbacks. One of the main problems with social media that is reinforced by the anonymous nature of these platforms is the toxic behaviour of a number of users. The exploitation of social media for propagating racial, sexual, and other forms of hateful content can even lead to real-world violence [1].

The social media giants such as Meta and Twitter, and the NLP research community are actively devising new techniques to detect hateful content in text. This classification task should not be treated lightly because on one hand misclassifying a tweet as hateful (false positive) would hinder individuals from exercising their freedom of speech. On the other hand, misclassifying hateful tweets as non-hateful would make the social media an unsafe and toxic space.

Before the prevalence of deep learning, most efforts focused on extracting informative features from text to feed into traditional machine learning algorithms, such as support vector machine (SVM), naive Bayes classifier, and logistic regression. Surface-level features such as bag of words (BoW), word-level and character-level n-grams yield the best performance with these methods [2]–[4]. Before the age of transformers, neural networks were used to extract text embeddings and classify them. For example, Djuric et al. [5] proposed a model based on continuous bag of words to extract paragraph2vec embeddings, and then trained a binary classifier to distinguish between hate and clean speech. Deep learning architectures such as FastText, CNN, GRU, and LSTM along with different feature embeddings including word2vec, GloVe, and character n-grams have also been explored for hate speech identification [6]–[8]. After the introduction of transformers [9], and subsequently BERT [10] and [11], transfer learning and fine-tuning large language models pre-trained on vast language corpora has gained popularity and been applied to many supervised and unsupervised NLP tasks. In this project, we implement and replicate the experiments in first work that proposed BERT fine-tuning strategies for hate speech detection. Mozafari et al. [12] introduced four fine-tuning strategies based on fully-connected neural networks, bidirectional LSTM, and convolutional neural networks (CNN) and achieved state-of-the-art performance on two hate-speech datasets.

After the replication these experiments, we also propose two extensions. The first extension is applying the proposed fine-tuning strategies to two other text classification datasets, the first one being an IMDB review dataset for sentiment classification, and the second one a dataset for determining eligibility of cancer patients for clinical trials. The second extension is adapting the current models to be compatible with multi-lingual hate speech detection. By these extensions, we will show that the fine-tuning strategies are robust enough to yield a good performance on other classification tasks and with another version of BERT such as multi-lingual BERT.

II. Methodology

In this section, we explain the architectures employed to solve the tasks at hand. For all of the architectures, we used BERT as the backbone network. BERT, bidirectional encoder representations from transformers, uses layered transformers to learn semantically rich embeddings from text sequences. For the mono-lingual (English) datasets, we used $\text{BERT}_{\text{base}}$ that

is pre-trained on two large-scale English Wikipedia and Book Corpus containing 2500M and 800M tokens respectively using a masked language modelling (MLM) objective. $\text{BERT}_{\text{base}}$ has 12 layers (transformer blocks), 12 self-attention heads, with embeddings of size 768. The total number of trainable parameters add up to around $110M$. For multi-lingual tasks, we used multi-lingual BERT, with the same architectural features but pre-trained on 104 languages with the largest Wikipedia.

By employing the BERT backbone, we adopt four different architectural modifications (fine-tuning strategies) to adopt the model on the different tasks. In each of these strategies, we first load the pre-trained BERT weights, add the components related to the strategy, and then fine-tune the model on the training set of the current task. These four architectures are explained below:

- **Single linear layer:** The simplest model receives the pre-processed tweet as the BERT input. We use the [CLS] token output vector (the classification token representing the final sequence encoding) from BERT and pass it through a linear layer of neurons of the size of the vector embedding (768). This layer is connected to a layer with three neurons that with softmax activations to calculate the probabilities for each class.
- **Multi-layer ANN:** As the second architectural extension, instead of a single linear layer at the top of the [CLS] output, we create a larger and more robust model by adding a multi-layer fully-connected neural network with two hidden layers of size 768 that use a leaky Relu activation function with a negative slope equal to 0.01. The final layer has three neurons with softmax activations for class prediction.
- **Sequential Bi-LSTM:** The previous extensions only used the BERT [CLS] token output for the classification layer. Although the [CLS] may be thought as an embedding representing the whole sequence, we may be able to capture more semantic information using all outputs of the last transformer encoder. Therefore, the last transformer outputs are fed into a bidirectional LSTM layer for further sequential processing. The final hidden state of this layer goes through a fully connected network similar to the first architecture (single linear layer) for classification.
- **CNN layer:** The last architecture takes advantage of convolution on transformer outputs. For every transformer encoder, we concatenate the vector embeddings to create a matrix of dimensions 768 by 64. The number of channels for convolution is 13 that is equal to the number of transformers plus one (the first channel uses the BERT initial token embeddings). We chose a 3 by 768 kernel and a stride of 1 to construct the convolutional layer. The outputs go through a max pooling layer and the resulting vector connects to three neurons with softmax activations for class prediction.

A graphical representation of the proposed architectures are given in Fig. 1.

It is noteworthy to mention that depending on the number of classes in the classification task, the number of neurons in the final softmax layer can be changed to adopt the model architecture to the task at hand.

III. Experiments

In this section, we first introduce hate speech datasets used for evaluation by Mozafari et al. [12]. Afterwards, we discuss the pre-processing steps, model implementation details and provide the classification results. In the last part, we present two different extensions. The first extension is the evaluation of the proposed architectures on two other text classification sub-tasks, one sentiment classification task and one on detecting eligible cancer patients for clinical trials. The second extension is a multi-lingual extension of the proposed frameworks for multi-lingual hate speech detection.

A. Hate Speech Datasets

The two different hate speech datasets were used by the authors for evaluating the proposed models. The first dataset is gathered by Waseem and Hovey [13] who collected it by searching through tweets containing racial, sexual and religious slurs and annotated the tweets manually with three classes of "racism", "sexism", and "neither". They merged this dataset with another one with many overlapping tweets provided by Waseem [14]. Currently, only the tweet IDs, and not their whole text are available online. We tried to extract the tweets corresponding to these IDs using the Twitter API, however almost all of the hate-speech tweets belonging to "racism" and "sexism" classes are deleted or the tweet writer's account has been suspended. Given this fact, for this project, it was not plausible to perform evaluation on these partially extracted data to replicate the results obtained by Mozafari et al. [12].

The second dataset is gathered by Davidson et al. [15]. They collected 84.4 million tweets containing hate speech words and phrases provided at hatebase.org, and random sampled $25k$ tweets and annotated them with the help of the CrowdFlower crowd sourcing platform users. This annotated dataset has three classes (hate, offensive, and neither) and is fully available online. The class distribution of this dataset is given in Fig 2. As can be seen the dataset suffers from an acute class imbalance. We will partially address this issue by stratified sampling discussed in section III-C.

B. Pre-processing

Since we are working with sequential text directly, there is no need to remove stop words from data samples. For pre-processing, we replaced some textual components with their corresponding BERT tokens, specifically mentions of users (in tweets), numbers, hashtags, and common emoticons were replaced by <user>, <number>, <hashtag>, <url>, and <emoticon> respectively. As the last step, punctuation marks and invalid characters were removed and all text samples were converted to lowercase. For the Davidson dataset that contains tweets, that are generally short, we truncated each sample to

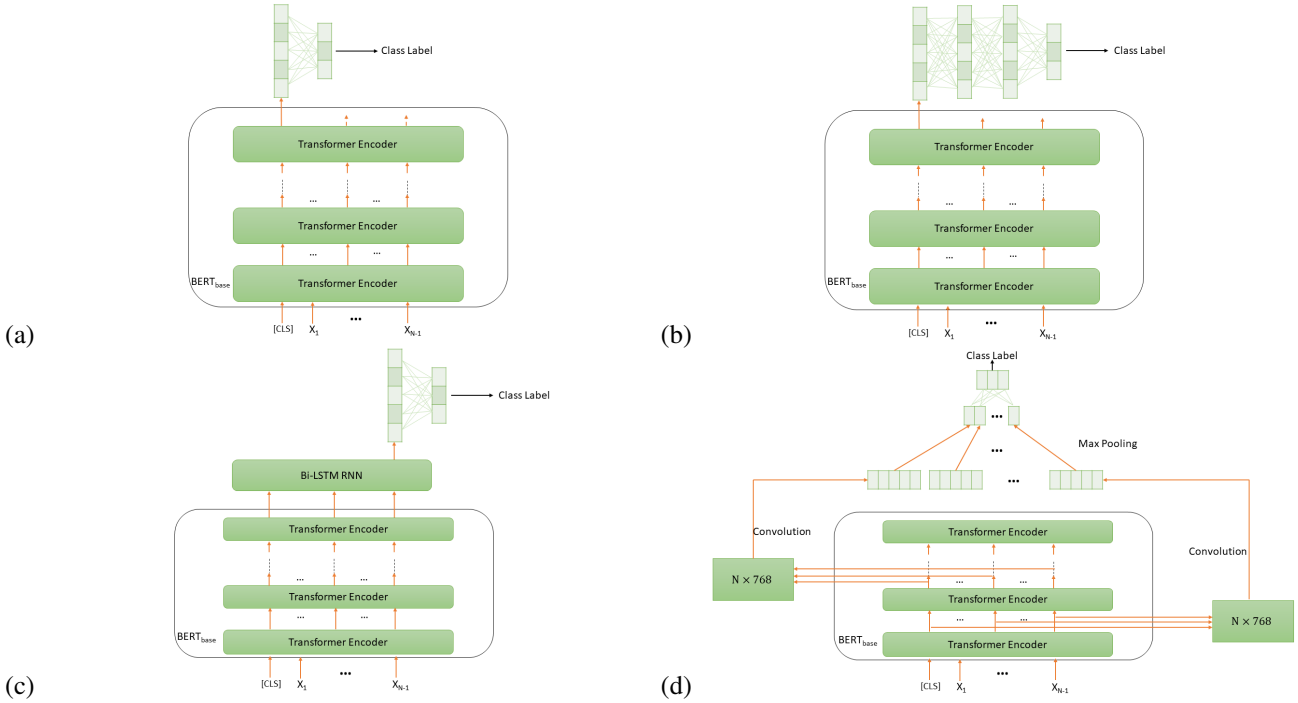

Fig. 1. The proposed BERT-based fine-tuning strategies. (a) Single linear layer (b) Multi-layer neural network (c) Sequential Bi-LSTM (d) CNN Layer

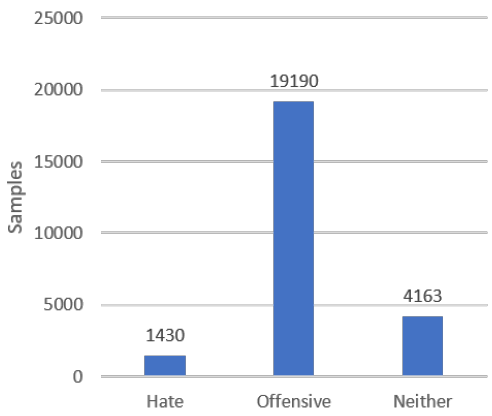

Fig. 2. Davidson hate speech dataset class distribution

64 tokens (or padded with zeros if they are shorter), and set the maximum length of the BERT tokenizer to 64.

C. Implementation and Results

To implement the fine-tuning strategies, we used the transformers package provided by Hugging Face [16] for pre-trained BERT models, and PyTorch [17] for the BERT head extensions. An implementation of the *Kedro* framework is used in order to provide an organized pipeline. This implementation includes a *classical* training and a *transformers* library adaptation of the code. The transformers adaptation uses the *Hugging Face* API, which includes optimized code for fine-tuning BERT models. Within the Kedro framework two pipelines where created: the data engineering (handling the pre-processing part) and the data science pipeline (for the implementation of the models). To switch between datasets, tasks, and fine-tuning strategies, we can easily provide Kedro with the required set of hyperparameters using a high-level and user-friendly API. For training, we used a Tesla K80 GPU with $12GB$ RAM provided by Google Colaboratory. For each experiment, the classifier was trained for 3 epochs with a batch size of 32. To avoid overfitting in deep extensions, we applied dropout with probability of 0.1 after hidden layers. The Adam optimizer [18] with a learning rate equal to $2e-5$ was used for training. All codes along with reprehensibility instructions are available at https://github.com/hafezgh/Hate-Speech-Detection-in-Social-Media.

As mentioned before, the Davidson dataset has an unbalanced class distribution. Since hate speech is a real phenomenon, the authors did not perform oversampling or undersampling to adjust the class distribution. However, to avoid overfitting, they perform stratified sampling when splitting the dataset into three separate sets with 80 percent training data, 10 percent validation data, and 10 percent test data. In stratified sampling, 0.8, 0.1, and 0.1 portions of tweets from each class (in our case hate/offensive/neither) are allocated to train, validation, and test sets respectively. Since we are dealing with class imbalance, we report the classification performance of the fine-tuning strategies in terms of precision, recall, and weighted-average F1-score that is a robust measure in presence of class imbalance. Table I shows these metrics for the fine-tuning strategies compared to the baseline model. As can be seen from this table, the CNN strategy achieves the best

performance with an F1-score of **91%**.

D. Extensions

We propose two extensions to the work published by Mozafari et al. [12] and present the results in the following sections.

1) Text Classification for Sentiment Analysis and Clinical Trial Eligibility Detection: As the first extension, we apply the proposed architectures to two different datasets to measure the effectiveness and robustness of the proposed fine-tuning strategies in other sub-domains of text classification. The first task is related to sentiment analysis. We employ a dataset containing IMDB movie reviews annotated with binary classes (positive and negative). The dataset is provided by Maas et al. [19] and has a balanced class distribution with $25k$ train, and $25k$ test samples. We use $20k$ samples from the train set for training, and $5k$ samples for validation, and trained the models with the same hyperparameters as the original experiment. The only difference is the maximum length of the BERT tokenizer which is set to 128 instead of 64 because we have also some longer reviews in the dataset. The F1-scores obtained on the test set with the different models were as follows: 97% with the linear layer, 88% with non-linear layers, 94% using the the Bi-LSTM, and 97% using the CNN model.

The second classification dataset is taken from a Kaggle competition [20]. The objective in the competition is to create a model able to predict if a patient diagnosed with some type of cancer is eligible for a treatment given its intervention studies and diagnoses from clinical trials. The task a binary classification task, and the dataset consists of one million samples, with 500 thousand samples belonging to each class. All of the model hyperparameters are the same as the original experiment with the only difference in the maximum length of the BERT tokenizer. Because the dataset is large and training takes a longer time than before, we use a smaller maximum length equal to 50 for this experiment. The F1-score obtained when applying the current models were 95.31% for linear layer, 95.6% for non-linear layers, 95.45% for Bi-LSTM, and 95.29% for the CNN model. The best performing model was obtained using the non-linear layers strategy. F1-score for the test set was 95.29%. The results of these experiments were monitored using the *wandb* tool, that is compatible with the transformers library, along with the corresponding plots. It is noteworthy to mention that the result obtained by our best model (95.6%) outperforms the original paper [20] that achieved a F1-score of 93% by using a CNN model.

2) Multi-lingual Hate Speech Detection: Our second proposal is a multi-lingual extension of the BERT-based architectures. In order to apply this extension, we use the multi-lingual BERT model instead of $\text{BERT}_{\text{base}}$ in all of the architectures. The multi-lingual BERT is trained on 104 languages with the largest Wikipedia using a masked language modelling (MLM) objective. To evaluate the multi-lingual models, we use the dataset from the HASOC (Hate Speech and Offensive Content Identification) 2020 competition [21]. This dataset consists of 10,000 doubly-annotated tweets in three Indo-European languages (English, German, and Hindi). The provided labels are used to solve two related sub-tasks. The first, and easier, sub-task is a coarse-grained binary classification problem with two classes; hate and offensive (HOF), which includes any hate speech, profane, or offensive content, and non hate-offensive (NOT) constituting regular and safe tweets. The harder sub-task classifies the HOF class into three classes named hate speech (HATE), offensive (OFFN), and profane (PRFN), and the classification task will have four classes (the mentioned three labels plus the NOT (safe) class). As can be seen from Fig. 3, this dataset is unbalanced (in terms of hate speech classes and not the languages) similar to the Davidson dataset. Therefore, we will use stratified sampling with 80/10/10 ratios to split the dataset into train, test, and validation for each sub-task. We use the same hyperparameters as the original experiment and run each of the fine-tuning strategies on each sub-task. Table II shows the fine-grained results obtained on these two sub-tasks and on different languages compared to the best submissions in the competition that we outperformed by a large margin. The best language-wise and overall results are obtained using the linear layer, and LSTM fine-tuning strategies.

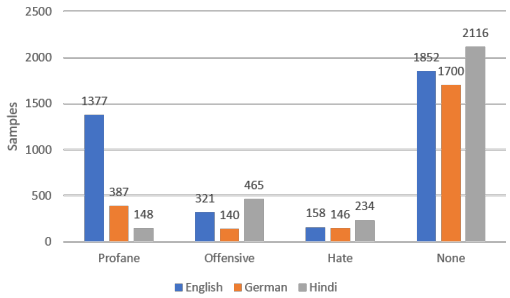

Fig. 3. HASOC multi-lingual hate speech dataset class distribution

IV. Discussion

We can further investigate the quality of the fine-tuning strategies in hate speech detection by analyzing the classification errors on the two benchmarks considered. Fig. 4 shows the confusion matrix obtained using the CNN strategy on the Davidson dataset. We can see that although the model has performed relatively well in terms of recall and precision for the offensvie and neither classes, the large number of false positives (FP) and false negatives (FN) for the hate class can be problematic. The presence of FP samples in an automatic hate speech detection system would fringe upon the users' freedom of expression and the presence of FN samples may allow hateful comments to remain and propagate through the social network.

Regarding the multi-lingual dataset, Fig. 5 shows the confusion matrix related to each sub-tasks achieved by the overall best-performing model (linear layer strategy). The low recall and precision especially for hate and offensive classes can be further investigated using a manual inspection of the test

TABLE I
RESULTS ON THE DAVIDSON HATE SPEECH TEST DATASET USING DIFFERENT FINE-TUNING STRATEGIES AND COMPARISON WITH THE BASELINE MODEL

Method	Precision (%)	Recall (%)	F1-Score (%)
Davidson et al. [15]	91	90	90
$BERT_{base}$+Linear Layer	91	92	91
$BERT_{base}$+Non-linear Layers	90	92	90
$BERT_{base}$+LSTM	91	92	91
$BERT_{base}$+CNN	**91**	**92**	**91**

TABLE II
RESULTS ON THE HASOC MULTI-LINGUAL HATE-SPEECH DATASET TEST DATASET USING DIFFERENT FINE-TUNING STRATEGIES AND COMPARISON WITH THE BASELINE MODEL

Method	Precision 1	Recall 1	F1-Score 1	Precision 2	Recall 2	F1-Score 2
English						
Best HASOC Submission [21]	52	-	-	27	-	-
$BERT_{multi-lingual}$+Linear Layer	87	87	87	**80**	**82**	**81**
$BERT_{multi-lingual}$+Non-linear Layers	85	84	84	78	82	77
$BERT_{multi-lingual}$+LSTM	**89**	**88**	**88**	77	82	77
$BERT_{multi-lingual}$+CNN	87	86	86	38	46	42
German						
Best HASOC Submission [21]	52	-	-	29	-	-
$BERT_{multi-lingual}$+Linear Layer	83	83	83	**77**	79	**78**
$BERT_{multi-lingual}$+Non-linear Layers	83	79	80	73	80	76
$BERT_{multi-lingual}$+LSTM	**84**	**84**	**84**	75	**80**	77
$BERT_{multi-lingual}$+CNN	84	83	83	65	71	68
Hindi						
Best HASOC Submission [21]	53	-	-	33	-	-
$BERT_{multi-lingual}$+Linear Layer	**66**	**71**	61	62	**74**	65
$BERT_{multi-lingual}$+Non-linear Layers	64	69	63	58	74	64
$BERT_{multi-lingual}$+LSTM	63	65	**64**	**64**	72	**65**
$BERT_{multi-lingual}$+CNN	50	70	58	62	69	62
Overall						
Best HASOC Submission [21]	-	-	-	-	-	-
$BERT_{multi-lingual}$+Linear Layer	**81**	**80**	**80**	**74**	**79**	**75**
$BERT_{multi-lingual}$+Non-linear Layers	78	78	78	71	79	73
$BERT_{multi-lingual}$+LSTM	79	80	79	73	78	74
$BERT_{multi-lingual}$+CNN	80	80	79	73	79	73

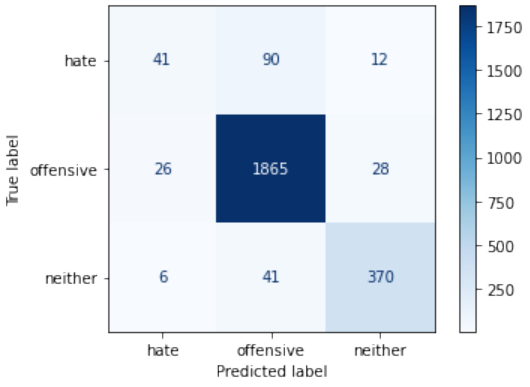

Fig. 4. Confusion matrix for the best-performing fine-tuning strategy (CNN) on the Davidson dataset

set and the model predictions. Table III shows some of the misclassified samples from the HASOC dataset. A few observations can be made:

- In the first two samples and the fourth one, the model has failed to capture the complex semantics that led to desired annotation. There are no hateful or offensive words used in these tweets, and the model should rely entirely on the context and semantic structure to detect the right class. A richer training dataset may be a possible solution to this problem.
- In the third sample, the swear word covered with the star may not have been recognized by the language model which resulted in a false prediction.
- It is possible that some of the tweets could belong to two or more classes (such as the last two rows in the table). Therefore, the language model may sometimes predict a correct label that may be different from the annotaion due to the limitations in the dataset's labeling mechanism.

TABLE III
MISCLASSIFIED SAMPLES FROM HASOC 2020 SUB-TASK 2

Tweet	Annotated	Predicted
RT @LindseyGrahamSC: When it comes to China, they will never change their behavior until someone stands up to them. I'm proud of President...	Hate	None
RT @MAGAGwen: Here's a thought: How about they go back to the countries they came from instead of forcing their teachings/customs in our c...	Hate	None
RT @KylieJenner: KYLIE F*CKING SKIN! wow. skincare and makeup go hand in hand and Kylie Skin was something i dreamt up soon after Kylie Co...	Offensive	None
RT @joaoafonso2002: "You're not a monkey, darling... tou're a gorila" Ahahhahaha	Offensive	None
@UtdAlii @FCBarcelona Ask shit questions, get shit answers, Ali.	Offensive	Profane
Bitch better have my money	Offensive	Profane

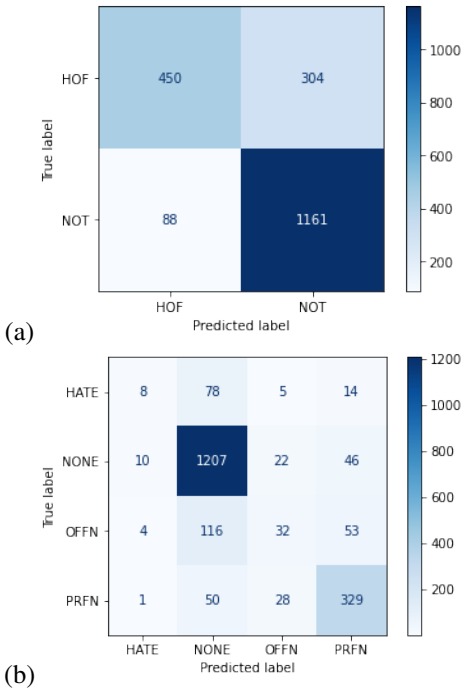

Fig. 5. Confusion matrix for the best-performing fine-tuning strategy (linear layer) on the HASOC 2020; (a) sub-task 1 (two classes). (b) sub-task 2 (four classes)

V. CONCLUSION

In this project, we reproduced one of the first works that utilized transformer-based architectures and fine-tuning strategies for the problem of hate speech detection. We also showed that these proposed fine-tuning techniques can be easily applied to other sub-tasks of text classification and achieve a great performance. The BERT backbone in the models considered can also be substituted with it multi-lingual version to address multi-lingual hate speech detection tasks. In the last part, we performed a short error analysis using confusion matrices and manual inspection of the multi-lingual hate speech benchmark and the model predictions.

ACKNOWLEDGMENT

The authors would like to thank Professor Luca Cagliero and Dr. Moreno La Quatra for their guidance and supervision during the project.

REFERENCES

[1] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," *Policy & internet*, vol. 7, no. 2, pp. 223–242, 2015.

[2] Y. Mehdad and J. Tetreault, "Do characters abuse more than words?" in *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2016, pp. 299–303.

[3] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th international conference on world wide web*, 2016, pp. 145–153.

[4] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.

[5] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate speech detection with comment embeddings," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 29–30.

[6] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th international conference on World Wide Web companion*, 2017, pp. 759–760.

[7] B. Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proceedings of the first workshop on abusive language online*, 2017, pp. 85–90.

[8] Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in *European semantic web conference*. Springer, 2018, pp. 745–760.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[12] M. Mozafari, R. Farahbakhsh, and N. Crespi, "A bert-based transfer learning approach for hate speech detection in online social media," in *International Conference on Complex Networks and Their Applications*. Springer, 2019, pp. 928–940.

[13] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on Twitter," in *Proceedings of the NAACL Student Research Workshop*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 88–93. [Online]. Available: https://aclanthology.org/N16-2013

[14] Z. Waseem, "Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter," in *Proceedings of the First Workshop on NLP and Computational Social Science*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 138–142. [Online]. Available: https://aclanthology.org/W16-5618

[15] T. Davidson, D. Bhattacharya, and I. Weber, "Racial bias in hate speech and abusive language detection datasets," in *Proceedings of the Third Workshop on Abusive Language Online*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 25–35. [Online]. Available: https://aclanthology.org/W19-3504

[16] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[20] A. P. Aurelia Bustos, "Learning eligibility in cancer clinical trials using deep neural networks," 2018.

[21] T. Mandla, S. Modha, G. K. Shahi, A. K. Jaiswal, D. Nandini, D. Patel, P. Majumder, and J. Schäfer, "Overview of the hasoc track at fire 2020: Hate speech and offensive content identification in indo-european languages," 2021.