McRock at SemEval-2022 Task 4: Patronizing and Condescending Language Detection using Multi-Channel CNN, Hybrid LSTM, DistilBERT and XLNet

Marco Siino, Marco La Cascia, and **Ilenia Tinnirello**
Università degli Studi di Palermo, Palermo, Italy
{marco.siino, marco.lacascia, ilenia.tinnirello}@unipa.it

Abstract

In this paper we propose four deep learning models for the task of detecting and classifying Patronizing and Condescending Language (PCL) using a corpus of over 13,000 annotated paragraphs in English. The task, hosted at SemEval-2022, consists of two different subtasks. The Subtask 1 is a binary classification problem. Namely, given a paragraph, a system must predict whether or not it contains any form of PCL. The Subtask 2 is a multi-label classification task. Given a paragraph, a system must identify which PCL categories express the condescension. A paragraph might contain one or more categories of PCL. To face with the first subtask we propose a multi-channel Convolutional Neural Network (CNN) and an Hybrid LSTM. Using the multi-channel CNN we explore the impact of parallel word emebeddings and convolutional layers involving different kernel sizes. With Hybrid LSTM we focus on extracting features in advance, thanks to a convolutional layer followed by two bidirectional LSTM layers. For the second subtask a Transformer BERT-based model (i.e. DistilBERT) and an XLNet-based model are proposed. The multi-channel CNN model is able to reach an F1 score of 0.2928, the Hybrid LSTM model is able to reach an F1 score of 0.2815, the DistilBERT-based one an average F1 of 0.2165 and the XLNet an average F1 of 0.2296. In this paper, in addition to system descriptions, we also provide further analysis of the results, highlighting strengths and limitations. We make all the code publicly available and reusable on GitHub[1].

1 Introduction

With the exponential growth of contents shared on social networks, a lot of new challenging tasks have emerged. Many are currently studied and addressed by scholars, and a pletora of novel machine learning approaches have been proposed (Arpaci et al., 2021), (Hosseinalipour and Ghanbarzadeh, 2022), (Siino et al., 2020). Some of the most common tasks, often co-located with international conferences, are those about fake news (Rangel et al., 2020), hate speech (Bosco et al., 2018), misogyny (Fersini et al., 2018) and cyberbulling (Kumar et al., 2018) detection.

For these purposes there is a constantly growing need for tools that can automatically extract and classify information from online feeds, to face with consolidated as well as with emerging social issues. Interest in Natural Language Processing (NLP) has increased in recent years with advances in machine and deep learning architectures. There have been significant efforts in developing methods to automatically detect and classify text content available online nowadays.

Together with the already mentioned tasks, an emerging one is about detecting Patronizing and Condescending Language (PCL) (Pérez-Almendros et al., 2020). The PCL Detection Task hosted at SemEval-2022 is covered in detail in (Pérez-Almendros et al., 2022) and briefly discussed here. The main task is made of two subtasks. The first one is a binary classification problem where, given a paragraph, a model has to predict wheter the paragraph contains or not PCL. The second one is a multi-label classification task where each paragraph has to be labelled with one to seven categories of PCL. Classes are not mutually exclusive and so a paragraph could express one or more categories of PCL.

To face with the first subtask we propose two deep models. The first one is a multi-channel Convolutional Neural Network (CNN). Such a network consists of parallel word embedding and convolutional layers to allow different sets of weights for trained embeddings - because of different kernel sizes employed by convolutional layers. In terms of *Precision*, *Recall* and *F1*, results of our model show certain room for improvements in future work. The

[1] https://github.com/marco-siino/McRock-SemEval-2022-Task4

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 409 - 417
July 14-15, 2022 ©2022 Association for Computational Linguistics

second model is a hybrid bidirectional LSTM. Such a network is composed by a convolutional layer and two bidirectional LSTM layers.

For the second subtask we propose two Transormer-based models (Vaswani et al., 2017). The first one is a lighter and faster version of BERT (i.e. DistilBERT) (Sanh et al., 2019). Our model is opportunistically trained on an undersampled version of the training dataset. The model is able to outperform RoBERTa (Liu et al., 2019). The second is an XLNet-based one (Yang et al., 2019). The model is based on a generalized autoregressive pretraining method. It enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. Under comparable experiment setting, XLNet outperforms BERT (Devlin et al., 2019) on several tasks, often by a large margin, including question answering, natural language inference, sentiment analysis, and document ranking. Our model implementation is opportunistically trained on an undersampled version of the training dataset. The model is able to outperform RoBERTa (Liu et al., 2019) in terms of average F1.

The rest of the paper is made as follows. In Section 2 we provide some background on the Task 4 hosted at SemEval-2022. In Section 3 we provide a description of the models presented. In Section 4 we provide details about the experimental setup to replicate our work. In Section 5 the results on the official task and some discussion are provided. In section 6 we present our conclusion and proposals for future works.

2 Background

In this section we provide some background about the Task 4 hosted at SemEval-2022. The aim of this task is to identify PCL, and to categorize the linguistic techniques used to express it, specifically when referring to communities identified as being vulnerable to unfair treatment by the media. Participants at the Task 4 received a dataset with sentences in context (paragraphs), extracted from news articles. Although news articles were collected from different countries, they were all provided in English. The task consists of the two subtasks listed below.

1. Subtask 1: Binary classification. Given a paragraph, a system must predict whether or not it contains any form of PCL. Two opposite labelled samples from the dataset provided are shown below.

   Non-PCL Sample Text: *"Council customers only signs would be displayed . Two of the spaces would be reserved for disabled persons and there would be five P30 spaces and eight P60 ones ."*

   Non-PCL Sample Label: [0]

   PCL Sample Text: *"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?"*

   PCL Sample Label: [1]

2. Subtask 2: Multi-label classification. Given a paragraph, a system must identify which PCL categories express the condescension. The PCL taxonomy has been defined based on previous works on PCL. The proposed categories are:

   - Unbalanced power relations
   - Shallow solution
   - Presupposition
   - Authority voice
   - Metaphor
   - Compassion
   - The poorer, the merrier

Two samples from the dataset provided are shown below. For each sample the label is an array containing seven elements. For each element, symbol *1* means that the corresponding PCL category is expressed in the paragraph.

Sample Text 1: *"Yes ... because there is NO HOPE where he lives . India is a third-world country . Do n't be fooled by call centers in big cities . Most of the country is rural and most of the population is illiterate and hopeless ."*

Sample Label 1: [1, 0, 1, 0, 0, 1, 0]

Sample Text 2: *"For refugees begging for new life , Christmas sentiment is a luxury most of them could n't afford to expect under shadow of long-running conflicts ."*

Sample Label 2: [0, 0, 1, 0, 0, 1, 0]

Task organizers released a training and a dev set before the competition officially started. For both sets the gold labels were provided. During first phase - Practice phase - participants were able to develope and test their models uploading predictions on CodaLab. After releasing the unlabelled test set the second phase - Evaluation phase - started. Results for both phases are available online [2].

3 System Overview

In this section we discuss the models presented for each subtask and the design choices made by our team motivating them. For both models the code is publicly available and reusable. Further details are provided in Section 4.

3.1 Subtask 1: Binary Classification

Given the binary nature of the task and his subject, for our first submission we developed a more versatile CNN based on the one presented in (Siino et al., 2021). Such a network is composed of parallel word embedding and convolutional layers to allow different weights for embeddings and convolutional filters. A general overview of the model architecture is shown in Figure 1. The rationale of the model presented is to have more parallel convolutional-based channel, each with different word embeddings and kernel filter weights. More properly, we set kernel size of 1, 2, 16 and 32 for each of the 32 *Conv1D* layer filters. In this way we drive our model to focus more on single token, pair of tokens, group of 16 and of 32 tokens respectively. On the basis of our experiments these are the best-performing kernel sizes for the proposed task on our preliminary 10 cross-fold validation. In addition to this behaviour we expect different coordinates for each word/token in each word embedding channel, with the aim of getting a more fine-grained positioning of words/tokens in the embedding space.

Based on our preliminary experiments, we found that on five different seeds initialization, the best word embedding size for our model is 50. This size is consistent with the common values reported in literature (Melamud et al., 2016). For each dense layer we did not use any activation function. We trained our model with a binary cross-entropy loss and using the Adam optimization algorithm (Kingma and Ba, 2014).

For our second submission, we developed a light Hybrid LSTM. The model consists of a convolutional layer followed by two bidirectional LSTM layers. Such a strategy is motivated by our decision to extract relevant features from the word embedding layer before the first bidirectional one. A general overview of the model architecture is shown in Figure 2. Based on our preliminary experiments on five different seeds initialization, we found that the best word embedding size for the model was 50. For each dense layer we did not use any activation function. We trained our model with a binary cross-entropy loss and using the Adam optimization algorithm (Kingma and Ba, 2014).

3.2 Subtask 2: Multi-Label Classification

For our first submission at the Subtask 2 we choosed a transformer-based model lighter than BERT (i.e. DistilBERT). Due to the high number of experiments to perform, we needed a faster model to train. DistilBERT is a smaller general-purpose language representation model. In DistilBERT the original size of BERT model is reduced by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. In terms of knowledge distillation, while BERT is the teacher, DistilBERT is the student. Student is represented by a compact model and is trained to reproduce the behaviour of the larger model (i.e. the teacher). Such a compact model is trained with a linear combination of three losses: the *distillation loss* (i.e. L_{ce}), the *masked language modeling loss* (i.e. L_{mlm}), and the *cosine embedding loss* (i.e. L_{cos}). Because of the distilled nature of the model, training and fine-tuning on a specific dataset for a specific task is of prominent importance. For a detailed discussion of DistilBERT refer to (Sanh et al., 2019). While we firstly compared the results on the dev set provided, we finally trained our model on the full training set - union of train and dev set - providing predictions on the test set. In addition we found beneficial maintaining the information about casing of characters. So we did not lowercase the text provided, implementing a cased version of DistilBERT and setting as output for each label seven digits corresponding to the seven categories of PCL. Finally we preprocessed each sample to include country and keyword of each paragraph in the input text.

For the second submission we implemented an XLNet-based model. Different unsupervised pre-

[2] https://sites.google.com/view/pcl-detection-semeval2022/ranking

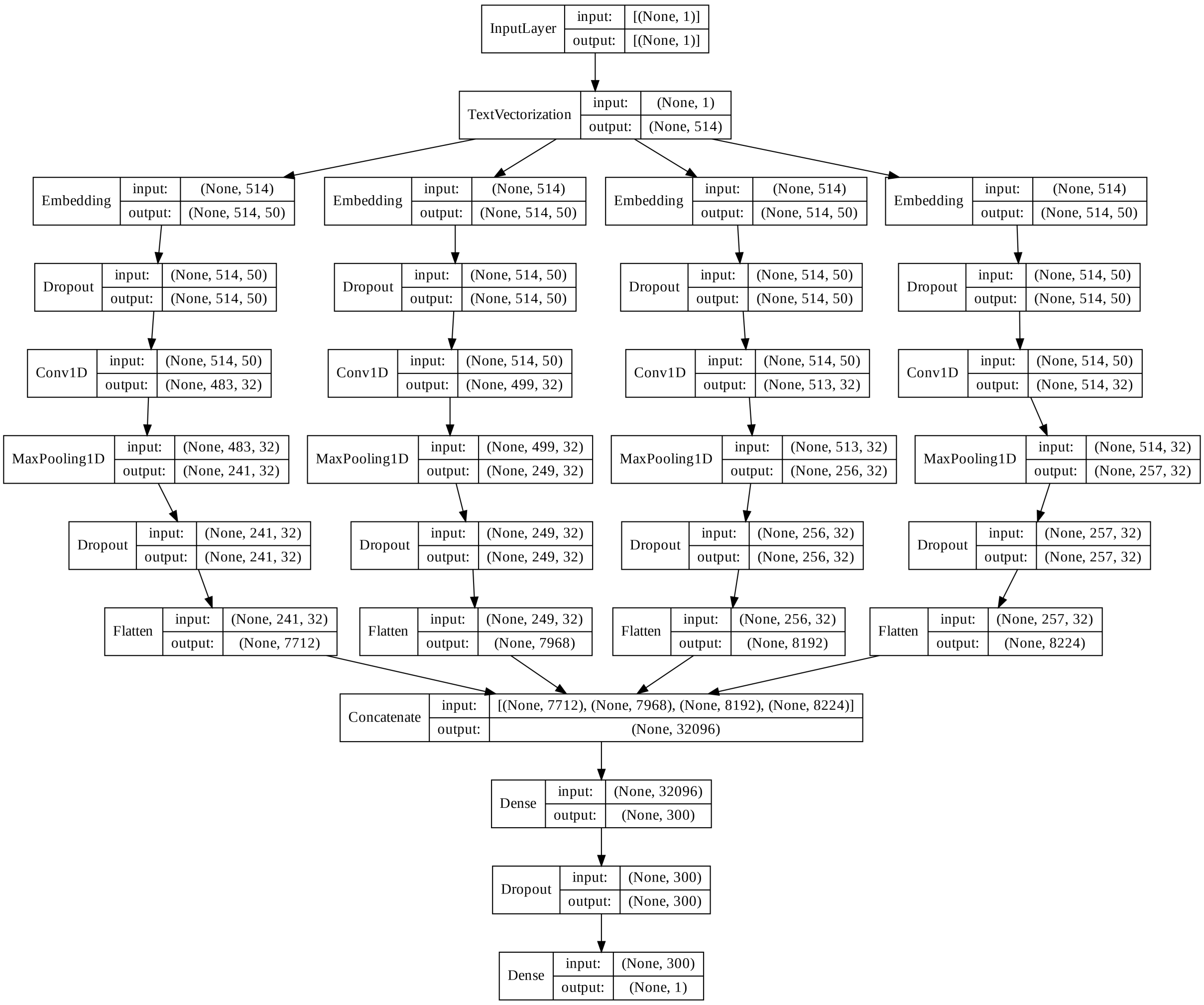

Figure 1: Overview of the multi-channel CNN presented for the first subtask at SemEval-2022. Each channel has a different kernel size at *Conv1D*, driving model attention on different sized windows of words. The kernel size of filters used at each *Conv1D* are 1, 2, 16 and 32. Each convolutional layer has 32 filters separately trained during training phase. Such a strategy allows extraction of different-sized features for a fine-grained learning.

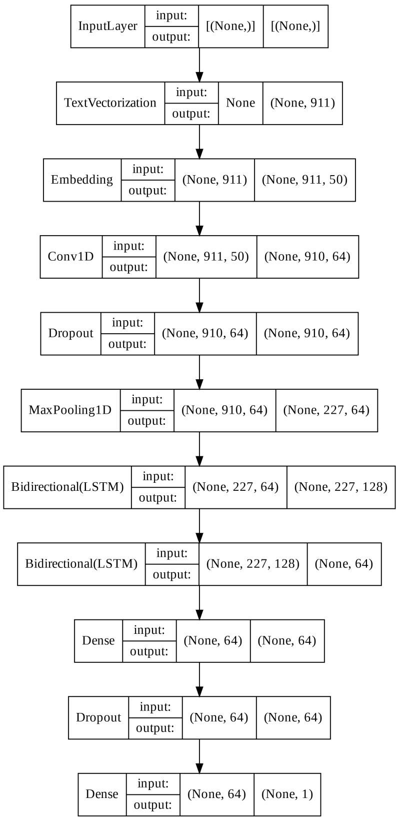
InputLayer | input: | output: | [(None,)] | [(None,)]
TextVectorization | input: | output: | None | (None, 911)
Embedding | input: | output: | (None, 911) | (None, 911, 50)
Conv1D | input: | output: | (None, 911, 50) | (None, 910, 64)
Dropout | input: | output: | (None, 910, 64) | (None, 910, 64)
MaxPooling1D | input: | output: | (None, 910, 64) | (None, 227, 64)
Bidirectional(LSTM) | input: | output: | (None, 227, 64) | (None, 227, 128)
Bidirectional(LSTM) | input: | output: | (None, 227, 128) | (None, 64)
Dense | input: | output: | (None, 64) | (None, 64)
Dropout | input: | output: | (None, 64) | (None, 64)
Dense | input: | output: | (None, 64) | (None, 1)

Figure 2: Overview of the Hybrid LSTM presented for first subtask hosted at SemEval-2022. The presence of the *Conv1D* layer is motivated by our intention to extract relevant features from the previous embedding layer. The kernel size of the 64 filters used by the convolutional layer is 2. Such a strategy should allows extraction of relevant bi-grams from the input text.

training objectives have been explored in literature. XLNet implements a generalized autoregressive pretraining method that uses a permutation language modeling objective to combine the advantages of autoregressive and autoencoding methods. The neural architecture of XLNet is developed to work seamlessly with the autoregressive objective, including the integration of Transformer-XL and the careful design of the two-stream attention mechanism. XLNet achieves substantial improvement over previous pretraining objectives on various tasks. Among them, autoregressive language modeling and autoencoding have been the two most successful pretraining objectives. Furthermore, XLNet integrates ideas from Transformer-XL (Dai et al., 2019) into pretraining. An XLNet model integrates two techniques from Transformer-XL, namely the relative positional encoding scheme and the segment recurrence mechanism. The relative positional encodings is applied based on the original sequence. Furthermore, the recurrence mechanism is included into the proposed permutation setting and enable the model to reuse hidden states from previous segments.

Training and fine-tuning of an XLNet for a specific task is of prominent importance. While we firstly compared the results on the dev set provided, we finally trained our model on the full training set - e.g., union of train and dev set - providing predictions on the test set.

4 Experimental Setup

We implemented our first two models using Keras[3] and TensorFlow[4]. The dataset provided for the binary classification task is unbalanced in terms of negative and positive PCL instances. To face with this issue we undersampled the negative instances. On the basis of our preliminary experiments, we found beneficial undersampling negative instances to be just six times more the positive ones. Furthermore we found beneficial to include in each sample (both for training and prediction) the keyword and the country field of each paragraph from the dataset. Then we used a batch size of 100. We empirically found that a good early stopping point for the training phase is obtained with 10 epochs and a learning rate of 0.001. We ran the experiments on Google Colab using the default GPU (NVIDIA Tesla K80). The training time was around 15 seconds for each

[3] https://keras.io/

[4] https://www.tensorflow.org/

of the ten epochs. The official metrics used for the task were Precision, Recall and F1 on positive instances (sample containing PCL). But during our development phase we focused on the model loss (i.e., *binary crossentropy loss*). This choose was dictated by the fact that the gold labels of the test set were not provided.

The models for Subtask 2 were implemented using Simple Transformers[5]. We used DistilBERT and XLNet as the pre-trained language models. We preprocessed the dataset to include, within the text of each sample, the country and the keyword of the paragraph. To train our final models we built a single dataset consisting of the train and the dev set. Then we undersampled negative instances (i.e. Non-PCL samples) to alleviate bias in the unbalanced dataset provided. We ran the experiments on Google Colab, using an NVIDIA Tesla K80 GPU. The official metrics used for the task were F1 for each category and average F1 among them. In this case too, during our development phase we focused only on the loss of the models to perform some fine-tuning.

5 Results

For Subtask 1 the metrics used are Precision, Recall and F1 defined as shown in 1,2,3 respectively.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

Each *True Positive* (TP) is computed on the positive instances (i.e. paragraphs containing PCL). So a TP is a sample containing PCL and correctly classified, a *False Positive* (FP) is a sample without PCL but wrongly classified as a PCL sample, a *False Negative* (FN) is a sample containing PCL but wrongly classified as not containing PCL. Therefore, Precision is the number of the correctly predicted PCL samples over the total number of predicted PCL samples. Recall is the number of the correctly predicted PCL samples over the total number of actual PCL samples. Finally the F1 Score is the harmonic mean of Precision and Recall.

The final ranking for the first subtask is drawn up accordingly to the F1 score on the test set provided.

	F1	P	R
RoBERTa-baseline	48.29	34.99	77.89
Multi-Channel CNN	32.29	23.46	51.76
Hybrid LSTM	26.32	31.47	22.61
Random-baseline	17.35	10.40	52.26

Table 1: Performance comparison on dev set. The results of the two baseline methods provided by the organizers (i.e. RoBERTa and Random baseline) compared to our models based on a multi-channel CNN and Hybrid LSTM.

In Table 1 are shown the results on the dev set provided by the organizers. Results are ordered according to F1 score. Our model based on multi-channel CNN is able to outperform the Random-baseline provided in terms of F1 and Precision, obtaining similar results in terms of Recall. RoBERTa-baseline performs better along the three metrics provided.

It is interesting to note that RoBERTa is a model pre-trained on over 160GB of text. Compared to our proposed model it requires much more in terms of resources and time needed. Despite such efforts, RoBERTa outperforms our model by only 16% and around 11% in terms of F1 and Precision. The most significant difference is with Recall. This means that the proportion of actual positives identified correctly by our model is lower compared to RoBERTa. This could be mainly due to the inability of our model at contrasting the bias learned because of the unbalanced dataset provided, where Non-PCL paragraphs are, in fact, the vast majority. Our team did an additional submission involving two deep models based on an Hybrid LSTM (i.e. made of convolutional and bidirectional LSTM layers) and on an XLNet (Yang et al., 2019). Our proposed Hybrid LSTM is able to outperform the Random-baseline provided in terms of F1 and precision. RoBERTa-baseline performs better along the three metrics provided. Compared to the Hybrid LSTM model, the multi-channel CNN outperforms the Hybrid LSTM. However the Hybrid LSTM performs better with regard to precision. Such a result leads to the conclusion that Hybrid LSTM correctly predicts an higher number of actual PCL paragraphs with respect to the total predicted PCL paragraps. Therefore, further investigation might be conducted on combinations of main components of the two proposed models in the effort to improve the F1.

[5] https://github.com/ThilinaRajapakse/simpletransformers

	F1	P	R
hudou (1)	65.10	64.60	65.62
RoBERTa (44)	49.11	39.35	65.30
Multi-CNN (69)	29.28	23.40	39.12
Hybrid LSTM (*NA*)	28.15	29.62	26.81
mahangchao (79)	4.48	10.59	2.84
makahleh (80)	0.0	0.0	0.0

Table 2: Performance comparison on test set. In table are shown the RoBERTa-baseline, the first classified (i.e. *hudou*), the two last classified and our models results. In parentheses are shown the positions in the final ranking according to F1 score. *NA* stands for *Not Assigned* because only the best result of the two model submitted is considered for final ranking.

In Table 2 are shown the results on the test set provided by the organizers without the gold labels. Results are ordered based on the F1 score. Compared to the winner (e.g. *hudou*), RoBERTa exhibits the most significant gap in Precision. Which means that proportion of positive instances correctly classified by the winner team is significantly more compared to RoBERTa. However, in this case too, RoBERTa outperforms our model with similar gap along the three metrics with respect to the results presented for the dev set. Our two submitted models exhibit similar performances on the test set. In this case too, the most significant gap is in Recall.

For Subtask 2 the metric used is F1 along the seven categories provided and the final ranking was drawn up considering the average F1 along the seven categories on the test set provided. For this subtask there is an important bias due to the unbalanced nature of the dataset with regard to each category. In Table 3*(a)* the results on the dev set are shown. Results are ordered based on the average F1 score. For each category our XLNet is able to outperform the Random-baseline. The average F1 is 15% more than such a baseline. It is worth noting that results with a random predictor are not uniformly distributed along each category. This distribution provides further evidences about the unbalanced nature of the dataset with regard to this multi-label classification subtask. Furthermore the random predictor outperforms F1 score of RoBERTa in four of the seven categories provided. However RoBERTa performs a lot better in detecting *Unb*, *Pre* and *Com* language (namely, *Unbalanced power relations*, *Presupposition* and *Compassion*). These performances could be motivated by the greater number of samples in the dataset expressing the first category. Compared to RoBERTa our DistilBERT-model does better for five categories out of seven. And for this single category (i.e. *Presupposition*) the gap is under 4%. Compared to our other submission, the XLNet heavily outperforms DistilBERT in terms of F1 for each category and in the final average F1. In Table 3*(b)* we report the results of the first model, our proposed models, RoBERTa and the last classified one, according to the final ranking drawn up considering the average F1. In this case too our models outperform RoBERTa, in terms of F1, for six out of seven categories. On the test set, RoBERTa performs better in detecting *Unb*. However, compared to the results on dev set, our two proposed models perform with a lower average F1 gap. And there is just a category (i.e. *Metaphor*) where DistilBERT significantly outperforms the XLNet. It is worth noting that the best performing model is able to reach an average F1 of 46.89, outperforming of over 20% and 36% our proposed models and RoBERTa respectively. This lead to a conclusion about the very large room for improvement in this multi-label task. Some of the difficulties in reaching an average F1 of at least 50% could be due to the unbalanced dataset as much as the intrinsic complexity of the task.

6 Conclusion

We propose four deep learning models to detect and classify PCL on the English dataset provided by task organizers at SemEval-2022. For the first subtask we developed a Multi-Channel CNN, training parallel word emebeddings and convolutional layers with different kernel sizes and an Hybrid LSTM. While results of these architectures exhibit a large room for improvements, the models are lighter and faster compared to the RoBERTa-baseline model proposed by the task organizers. For the second subtask we implemented a DistilBERT-based and XLNet-based models. Compared to RoBERTa, DistilBERT is smaller, faster and lighter. Instead, XLNet performs better on average F1 both on dev and test set. To face with the task proposed we opportunistically trained the models including the information about country and keyword related to each

	Unb	Sha	Pre	Aut	Met	Com	The	**AVG**
XLNet	47.99	20.41	24.61	20.06	16.67	39.24	8.89	25.41
DistilBERT	47.60	15.90	23.84	15.53	10.91	31.23	0.0	20.72
RoBERTa-baseline	35.35	0.0	29.63	0.0	0.0	28.78	0.0	13.40
Random-baseline	11.30	3.23	5.09	3.22	6.04	8.21	1.31	5.48

(a)

	Unb	Sha	Pre	Aut	Met	Com	The	**AVG**
guonihe (1)	65.60	52.94	36.90	40.66	35.90	49.18	47.06	46.89
XLNet (29)	32.32	32.93	19.18	20.55	22.22	26.35	7.14	22.96
DistilBERT (*NA*)	32.62	30.49	18.80	18.31	26.00	25.37	0.0	21.65
RoBERTa-baseline (37)	35.35	0.0	16.67	0.0	0.0	20.87	0.0	10.41
nikss (49)	0.0	1.01	0.0	0.0	0.0	0.0	1.09	0.03

(b)

Table 3: Performance comparison on dev set *(a)* and test set *(b)* for Subtask 2. The table shows F1 calculated for each category and the average F1 in the last column. For Subtask 2 our proposed models based on DistilBERT and XLNet outperform RoBERTa on both dev and test set. In parentheses are shown positions in final ranking. NA stands for *Not Assigned* in this case too.

sample. In addition we undersampled the negative instances in the dataset to avoid the model to focus more on non-PCL samples. The trained models are able to outperform RoBERTa. However, looking at the final ranking of the task, the room for improvements is significant. In future works would be useful implementing models taking advantage of a balanced dataset. Both for the binary classification task and for the multi-label one. Another interesting aspect to further investigate would be about the behaviours of the proposed models on multilingual datasets. Although pre-trained models are actually the state of the art for many NLP tasks, the hardness of the PCL detection task - proved by the final scores obtained by the winners at SemEval-2022 - could worsen the results on each metric. Finally, it could be beneficial experimenting with hybrid and ad-hoc models combining different pre-trained and non pre-trained models to improve the results specifically on this task.

Acknowledgments

We would like to thank anonymous reviewers for their comments and suggestions that have helped to improve the presentation of the paper.

CRediT Authorship Contribution Statement

Marco Siino: Conceptualization, Formal analysis, Investigation, Methodology, Software, Supervision, Validation, Visualization, Writing - Original draft, Writing - review & editing. **Marco La Cascia:** Supervision, Writing - review & editing. **Ilenia Tinnirello:** Supervision, Writing - review & editing.

References

Ibrahim Arpaci, Mostafa Al-Emran, Mohammed A Al-Sharafi, and Khaled Shaalan. 2021. A novel approach for predicting the adoption of smartwatches using machine learning algorithms. In *Recent advances in intelligent systems and smart applications*, pages 185–195. Springer.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, and Tesconi Maurizio. 2018. Overview of the evalita 2018 hate speech detection task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 2263, pages 1–9. CEUR.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

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.

Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. *Ibereval@ sepln*, 2150:214–228.

Ali Hosseinalipour and Reza Ghanbarzadeh. 2022. A novel approach for spam detection using horse herd optimization algorithm. *Neural Computing and Applications*, pages 1–15.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Ritesh Kumar, Guggilla Bhanodai, Rajendra Pamula, and Maheshwar Reddy Chennuru. 2018. Trac-1 shared task on aggression identification: Iit (ism)@ coling'18. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)*, pages 58–65.

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.*

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1030–1040.

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.

Francisco Rangel, Anastasia Giachanou, Bilal Hisham Hasan Ghanem, and Paolo Rosso. 2020. Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter. In *CEUR Workshop Proceedings*, volume 2696, pages 1–18. Sun SITE Central Europe.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108.*

Marco Siino, Elisa Di Nuovo, Tinnirello Ilenia, and Marco La Cascia. 2021. Detection of hate speech spreaders using convolutional neural networks. In *PAN 2021 Profiling Hate Speech Spreaders on Twitter@ CLEF*, volume 2936, pages 2126–2136. CEUR.

Marco Siino, Marco La Cascia, and Ilenia Tinnirello. 2020. Whosnext: Recommending twitter users to follow using a spreading activation network based approach. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 62–70. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.