JUST-DEEP at SemEval-2022 Task 4: Using Deep Learning Techniques to Reveal Patronizing and Condescending Language

Mohammad Makahleh and **Naba Bani Yaseen** and **Malak Abdullah**
Jordan University of Science and Technology
Irbid Jordan
maalmakahleh20, nmbaniyaseen21 @cit.just.edu.jo, mabdullah@just.edu.jo

Abstract

Classification of language that favors or condones vulnerable communities (e.g., refugees, homeless, widows) has been considered a challenging task and a critical step in NLP applications. Moreover, the spread of this language among people and on social media harms society and harms the people concerned. Therefore, the classification of this language is considered a significant challenge for researchers in the world. In this paper, we propose JUST-DEEP architecture to classify a text and determine if it contains any form of patronizing and condescending language (Task 4- Subtask 1). The architecture uses state-of-art pre-trained models and empowers ensembling techniques that outperform the baseline (RoBERTa) in the SemEval-2022 task4 with a 0.502 F1 score.

1 Introduction

The language used when talking about other people significantly impacts society and individuals. Talking about others and using caring and sympathetic language to express them causes them to be concerned, regardless of the author's intention, which is often to help others by raising awareness of their cause. Unfair treatment of vulnerable groups on social media increases exclusion and inequality(Kučak et al., 2018).

Several researchers study modeling language that intentionally undermines others, such as offensive language or hate speech (Zampieri et al., 2019; Faraj and Abdullah, 2021). PCL modeling is still a relatively new topic of research in NLP. For illustration, the use of PCL in the media is frequently unconscious, subtler, and more subjective than the sorts of discourse that are typically addressed in NLP. To the best of our knowledge, a particular focus on PCL for vulnerable communities has not yet been considered. Through a broader context, some work on PCL had been studied on communication between two people, such as in social media interactions, where others patronize an individual. For example, the authors in (Inui et al., 2019) published the talk down corpus for detecting condescension in Reddit comment-reply pairs. Finding or creating a high-quality dataset that covers all or most cases of PCL is very difficult and requires significant effort from specialized researchers. As researchers in NLP, we use it to investigate PCL in vulnerable communities(Perez-Almendros et al., 2020).

In this paper, we describe our model on task 4 of Semeval 2022 sub task1 (P'erez-Almendros et al., 2022) to identify PCL and categorize the linguistic techniques used to express it. Specifically when referring to communities identified as being vulnerable to unfair treatment in the media. We compare the outcomes of multiple pre-trained models (BERT, RoBERTa, and ensemble on them). We furthermore show the effect of tuning their hyperparameters values (batch-size and epoch) on model prediction. Also, we illustrate our proposed architecture where we feed the data to an ensemble model with the stacking of 2 BERT models and -in parallel- to another ensemble model with the stacking of 2 RoBERTa models. Finally, the results of the two ensembling models are fed to a max voting ensemble, which predicts the outcome with achieving the best F1 score at 0.502.

The rest of the paper is organized as follows; section 2 presents a Related Work, and section 3 offers Methodology. Results are discussed in section 4. Finally, section 5 presents conclusion.

2 Related Work

Many researchers applied machine learning models to classify text (Abdullah and Shaikh, 2018; Qawasmeh et al., 2019). Lai et al.(Lai et al., 2015) classified text using a recurrent convolutional neural network. They captured the key components in texts using the max-pooling layer from word representations. Then they grabbed contextual information from it. To check the efficacy of the proposed method, they conducted several experiments

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 308 - 312

on four commonly used data sets. As a result, they found that their suggested method outperforms the latest methods in many data sets.

Gunal et al.(Kowsari et al., 2019) studied text classification algorithms on several sides. Like limitations of each technique and its application in real-world situations are discussed. They found some steps useful in decreasing the time complexness and memory complexity of existing text classification algorithms like central component analysis and incidental projection, etc.

Using deep neural networks with LSTM modules, Semberecki et al.(Semberecki and Maciejewski, 2017) classified text documents. They tried to construct feature vectors, which represent the documents to be categorized: each feature vector represents the sequence of words that are included in the documents. First, they convert the terms into vector representations and then use the sequences of these vector representations as features of the documents. They evaluated the feasibility of this approach to text categorization utilizing a set of Wikipedia articles. They show that the LSTM network-based approach with documents represented as vectors achieves an accuracy of 86%.

3 Methodology

Our approach methodology can be summarized as follows: We begin by describing the dataset for this task. Then, the preprocessing step is described. Our final section describes how JUST-DEEP can identify the patronizing language in the text.

3.1 Data Set:

The SemEval-task 1 competition provided three files (rial, train, and test dataset(Perez-Almendros et al., 2020)) The files contain several columns as follows:

- par_id: the identification number for each paragraph.
- art_id: the identification number for each article.
- keyword: word related to patronizing. .
- country_code: the code for each country.
- text: the text that we want to classify.
- label: denotes if the text contains patronizing or not (4 levels 0,1,2,3 where 0 and 1 means no PCL, as well as 2 and 3, means high PCL).

3.2 Data pre-processing:

In this section we describe the basic data preprocessing steps we applied for all of our experiments:

1. In the beginning, we transform the Label column into a binary format. If the value is zero or one, we convert it to zero. And if it is two or three, we convert it to One. So with this transformation, the Label is a Binary column with two values: the Zero means that there are no PCL in the text, and the value One implies that there is PCL.We converted the column to binary because we are working on sub-task one, so we want to determine if it contains PCL or not; we don't care about the degree of PCL.

2. Pre-trained models don't work with raw text, so we converted the text into numbers and added unique tokens to separate sentences at the beginning and end of each sentence. Then we pass the resulting sequence to the models to perform the classification process.

3. Pre-trained models work with fixed-length sequences. So we used a simple strategy to choose the appropriate maximum length for all sequences. First, we found that most series have a size of 160, so we set the size of all series to 160.

3.3 JUST-DEEP Architecture:

Our task aims to detect whether a text contains PCL language or not. As shown in Figure 1, JUST-DEEP architecture uses multiple pre-trained language models (BERT and RoBERTa) from the transformers library.

The first step is data pre-processing. The training dataset is fed to the augmentation processor, responsible for adding more data to the training dataset since it is originally imbalanced data where the number of Zero class instances is ten times the One class instances. The augmentation processor input is all One instances text. The processor augments the text of the input instances by adding the same instance to the primary dataset multiple times with slightly different text but with the same meaning. Next, the textual data is processed into their corresponding embeddings(tokens) to feed them to the classifiers.

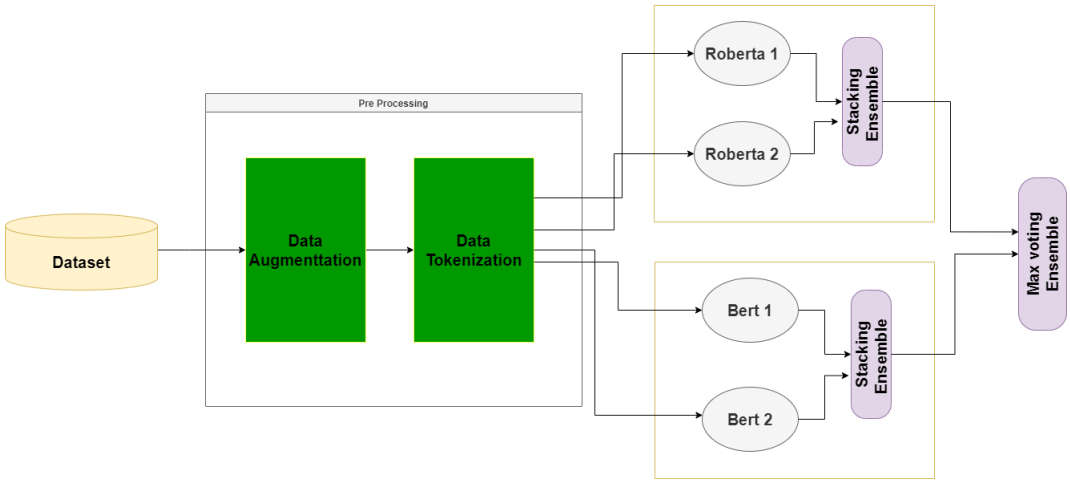

Figure 1: Model Architecture

The second step is the classification step. Finally, the input tokens are fed in parallel to two ensembling models; the first is composed of stacking of 2 BERT models, and the second contains stacking of 2 RoBERTa models.

Finally, the predictions of the two ensembling models are joined by a max voting classifier to produce the final prediction output. We conducted several experiments with different models and hyperparameters, but the JUST-DEEP architecture achieved the best results.

Figure 2 show an example use case for architecture. If we fed the sentence *'Fast food employee who fed disabled man becomes internet sensation'* which is an example from the train data labeled as 1 (contains PCL). In augmentation step we generate more instances with same meaning of this row by the augmentation processor which might add data like *'Fast food worker who fed weakened guy becomes internet rumor'*. Next, the sentence is converted into numeric token and passed to both ensemble models.
For instance, the first ensemble model which contains stacking of 2 RoBERTa models produce the prediction of 1 for this row. The other ensemble model with stacking of 2 Bert classifiers predicts Zero as a class label. Finally, the last step which implement a max voting will generate One as the final output label for the instance.

4 Result

We performed several experiments to determine which is the best suitable model for this task and which model produces the highest value of F1 score. First, we experimented with BERT and RoBERTa pre-trained models and tuned their hyperparameters (batch_size and epoch) to achieve the maximum possible F1 score. Table 1 shows the experiments we did, the models, their parameters, and the value of the f1 score we got in each experiment in the test dataset.

As a first step, we explored BERT and RoBERTa's best hyperparameters, such as batch_size, epochs, and max sequence length. Table 2 describes these hyperparameters.

We use five different models: BERT model, RoBERTa model, stacking of 2 BERT, stacking of 2 RoBERTa, and JUST-DEEP model as described in the architecture section. As shown in the table1 model, JUST-DEEP achieves the best results compared to the rest of the models we tested, as it reached an F1 score value equal to 0.502.

As we explained earlier, in the JUST-DEEP model, we trained the dataset with augmentation. Then we applied the Stacking Ensemble Technique separately on 2 BERT Models and 2 RoBERTa Models. Then, we took the Max voting between the two models as a result; therefore, this model produces the best results because it combines the two models that we used, BERT and RoBERTa.

The other models show less favorable results; the application of the stacking ensemble technique to the 2 BERT models gives the highest results after the JUST-DEEP, where the F1 result is close to 46%, followed by the model resulting from the application of the stacking ensemble technique to the 2 RoBERTa, where it achieved results close to 0.44. Then, RoBERTa's model achieved the F1 with a score of 0.43. The worst model was Bert achieved 0.42.

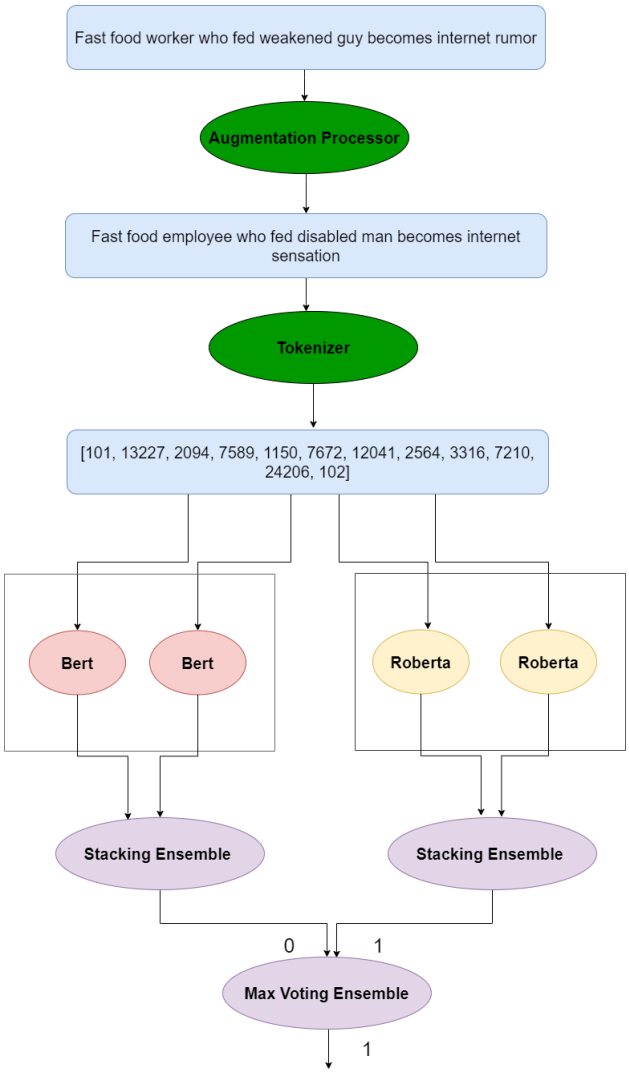

Figure 2: Example Description

5 Conclusion

Deep learning helped in the development of many aspects these days, and with in-text classification, we become to have the ability to make a lot of applications related to it and so on. In this paper, we describe our JUST-DEEP model solving the SemEval-2022 Task 4 Subtask 1 to check if the text contains any form of PCL. The JUST-DEEP model obtained an F1 score of 0.5 using ensembling of pre-trained language models BERT and RoBERTa. Our strategy depends on training data set with augmentation then applying stacking ensemble technique to 2 BERT and 2 RoBERTa and take max voting between the models to achieve F1 score higher than the other experiments we conduct where we use plain BERT and RoBERTa models with different hyperparameters.

References

Malak Abdullah and Samira Shaikh. 2018. Teamuncc at semeval-2018 task 1: Emotion detection in english and arabic tweets using deep learning. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 350–357.

Dalya Faraj and Malak Abdullah. 2021. Sarcasmdet at semeval-2021 task 7: Detect humor and offensive based on demographic factors using roberta pretrained model. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 527–533.

Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. 2019. Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

#	Classifier	Batch size	Epoch	F1
1	RoBERTa	16	10	0.43
2	RoBERTa	16	4	0.40
3	RoBERTa	8	10	0.42
4	RoBERTa	8	4	0.41
5	BERT	16	10	0.42
6	BERT	16	4	0.428
7	BERT	8	10	0.41
8	BERT	8	4	0.405
9	stacking of 2 RoBERTa	16	10	0.44
10	stacking of 2 RoBERTa	16	4	0.45
11	stacking of 2 RoBERTa	8	10	0.435
12	stacking of 2 RoBERTa	8	4	0.438
13	stacking of 2 BERT	16	10	0.46
14	stacking of 2 BERT	16	4	0.45
15	stacking of 2 BERT	8	10	46.5
16	stacking of 2 BERT	8	4	0.45
17	JUST-DEEP	16	10	0.5

Table 1: Our Experiments.

Parameter	Description	Values
Batch_Size	The number of samples per batch	16,8
Epoch	Training examples in both directions (backward and forward)	4,10
Dropout	Number of examples that can be neglected during training	0.3
Max Sequence Length	Sequence length the model can support	160

Table 2: Parameters

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Danijel Kučak, Vedran Juričić, and Goran Đambić. 2018. Machine learning in education-a survey of current research trends. *Annals of DAAAM & Proceedings*, 29.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

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 P'erez-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.

Ethar Qawasmeh, Mais Tawalbeh, and Malak Abdullah. 2019. Automatic identification of fake news using deep learning. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 383–388. IEEE.

Piotr Semberecki and Henryk Maciejewski. 2017. Deep learning methods for subject text classification of articles. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 357–360.

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). *arXiv preprint arXiv:1903.08983*.