NLP_CHRISTINE at SemEval-2023 Task 10: Utilizing Transformer Contextual Representations and Ensemble Learning for Sexism Detection on Social Media Texts

Christina Christodoulou
Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens
Institute for Language and Speech Processing, *Athena* Research Center
christinachristodoulou1997@gmail.com

Abstract

The paper describes the SemEval-2023 Task 10: *Explainable Detection of Online Sexism (EDOS)*, which investigates the detection of sexism on two social media sites, Gab and Reddit, by encouraging the development of machine learning models that perform binary and multi-class classification on English texts. The EDOS Task consisted of three hierarchical sub-tasks: binary sexism detection in sub-task A, category of sexism detection in sub-task B and fine-grained vector of sexism detection in sub-task C. My participation in EDOS comprised fine-tuning of different layer representations of Transformer-based pre-trained language models, namely BERT, AlBERT and RoBERTa, and ensemble learning via majority voting of the best performing models. Despite the low rank mainly due to a submission error, the system employed the largest version of the aforementioned Transformer models (BERT-Large, ALBERT-XXLarge-v1, ALBERT-XXLarge-v2, RoBERTa-Large), experimented with their multi-layer structure and aggregated their predictions so as to get the final result. My predictions on the test sets achieved 82.88%, 63.77% and 43.08% Macro-F1 score in sub-tasks A, B and C respectively.

1 Introduction

The Task and Its Importance Sexism emerged as a term from the *second-wave* feminism of the 1960s through the 1980s to comprise discrimination, prejudice and stereotyping based on someone's sex, typically targeting women and girls (Masequesmay, 2022). Even though our way of life has improved since then, it appears that sexism persists. On top of that, considering the immense influence of social networks on our lives, sexism is expressed online through text, image, video and sound on a daily basis. It has become evident that the anonymity and invisibility of the online environment foster the *online disinhibition effect*, meaning the tendency to exhibit negative behavior online rather than in person without taking into account the consequences (Wright et al., 2019),(Fox et al., 2015). In recent years, the amount of sexist content in social media ranging from cases of discrimination, abuse, stalking, bullying, impersonation, defamation, verbal and sexual harassment, non-consensual pornography, misogyny to cases of hate speech mainly against female individuals or groups has been on the rise. This has serious repercussions for the the mental health, the social status and the online experience of the targets, not to mention the negative impact on the community. To illustrate, the majority of women, who have experienced or witnessed sexist behavior online, state that they are discouraged from pursuing their political careers and upholding their rights, while they are led to self-censor and limit their presence on social media (Meco and MacKay, 2022). What is worse, acts of violence and abuse online can disrupt the targets' physical safety as well as their families' safety (Valenti, 2022). Online content moderators working in social media companies are responsible for mitigating such type of content according to each company's internal policies. Nevertheless, manual moderation of social media content is a time-consuming and laborious process and may often depend on the moderator's judgement (Pitsilis et al., 2018). Online content moderators may also suffer from various mental health problems, for instance PTSD and depression, owing to their continuous exposure to offensive, violent and pornographic content (Arsht and Etcovitch, 2018). The challenges and severe ramifications of manual social media moderation along with the increase of social media content have revealed that the need to automatically detect and eliminate such content is imperative.

Task 10 of SemEval-2023 competition named *Explainable Detection of Online Sexism (EDOS)* addressed this particular need by urging participants to develop machine learning models, which are able

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 595–602

to detect sexist posts from two social networks, Gab and Reddit, by means of performing binary and fine-grained multi-class classification on English texts (Kirk et al., 2023). The EDOS shared task comprised three hierarchical sub-tasks: Sub-task A was focused on identifying whether a post is sexist or not. Sub-task B was devoted to recognizing four categories of sexism. Sub-task C aimed at detecting sexism based on eleven fine-grained vectors of sexism. Sub-task A contained both sexist and non-sexist texts, whereas the other two sub-tasks were exclusively focused on the sexist texts from the first sub-task.

System Strategy The system strategy presented in this paper employs and fine-tunes pre-trained Transformer models, namely BERT-Large, ALBERT-XXLarge-v1, ALBERT-XXLarge-v2, RoBERTa-Large, by utilizing the word embeddings from their multi-layer structures. The test and development set submissions for each sub-task were the result of majority voting ensemble, which combined the predictions from the models that managed to achieve the highest Macro-F1 and MCC score on the development sets.

Key Results It became evident that machine learning models have the potential to successfully identify and differentiate the sexist from the non-sexist social media content. Nevertheless, they have yet to successfully explain the type of sexism and why the content is sexist, since they can hardly distinguish between different categories of sexism as well as between additional sub-categories. The provided training sets for all EDOS sub-tasks were imbalanced, which posed a difficult problem during model training and evaluation as the models had the tendency to over-predict the majority classes. This can be illustrated from the final results as the system achieved a much higher Macro-F1 score on the binary classification task (sub-task A) than on the other two multi-class classification tasks (sub-task B and C). However, due to my error when submitted the test predicted results with false text ids, my final system submissions achieved 84^{th} place in sub-task A, 69^{th} place in sub-task B and 61^{st} place in sub-task C. Despite my mistake, which led to my low official rank in the leaderboard, I managed to produce the correct final results. If it were not for the submission error, my system would receive 45^{th} place out of 84 submissions in sub-task A, 25^{th} place out of 69 submissions in sub-task B and 29^{th} place out of 63 submissions in sub-task C. The code for the submitted system can be accessed via this link on Github: https://github.com/christinacdl/Thesis_Detection_of_Offensive_Language/blob/main/Task_10_Sexism.ipynb

2 Background

Online Sexism Tasks Task 10 of SemEval-2023 named *Explainable Detection of Online Sexism (EDOS)* explored the detection of sexism on social media by taking advantage of texts in English from Gab and Reddit (Kirk et al., 2023). In this paper, my participation in all EDOS sub-tasks is described in detail. It is also worth-mentioning that, apart from this year's shared task, online sexism has been approached before in the 5^{th} shared Task of SemEval-2022, *Multimedia Automatic Misogyny Identification*, focusing on detecting misogynistic textual and visual memes on the web (Fersini et al., 2022).

Task 10 Data The provided dataset consisted of 20,000 texts in total, half collected from Gab and half from Reddit. The dataset was split by the task organizers into 70% training, 10% development and 20% test sets. The training set for sub-task A consisted of 14,000 labelled texts, of which 3,398 were classified as sexist. The training set for sub-tasks B and C included only the 3,398 sexist texts. Additional auxiliary but unlabelled data from Gab and Reddit, each containing 1 million texts, were also offered. The provided development set comprised 2,000 entries for sub-task A and 486 entries for sub-task B and sub-task C respectively. The provided test set included 4,000 texts for sub-task A and 970 entries for sub-task B and sub-task C respectively. The labels of the development sets were provided earlier than the test sets so as to use the development sets to evaluate the systems' performance based on the Macro-F1 score. The labels of the test sets were released after the end of the competition. Table 5 in appendix A illustrates the class distribution of the training sets for all sub-tasks. The classes of sub-task C consist sub-categories of the classes in sub-task B. The categorical labels were converted into the respective numerical labels denoted in brackets for training and evaluation purposes.

3 System Overview

Transformer-based pre-trained language models such as BERT (Devlin et al., 2018) and its variations, like ALBERT (Lan et al., 2019) and RoBERTa, (Liu et al., 2019) have attracted a lot of attention since 2018, as they have managed to achieve state-of-the-art results in various NLP tasks. They have been widely used for fine-tuning downstream tasks by simply adding an additional task-specific output layer. During fine-tuning, the output of the Transformer encoder's last layer, which is the contextualized representation of the input text, is passed to the task-specific layer. However, utilizing only the last layer's output may limit the power of pre-trained representation (Yang and Zhao, 2019). Considering the multi-layer and deep structure of the pre-trained Transformer language models, different layers are able to capture various levels of contextualized representations (embeddings) of the input text. Therefore, they encode very different kinds of linguistic information, for instance surface, syntactic and semantic features in the lower, middle and higher layers respectively (Peters et al., 2018). The authors of BERT (Devlin et al., 2018), followed a feature-based approach by extracting the contextual embeddings from different layers of BERT-Base and providing them as input to a BiLSTM for a Named Entity Recognition task. Inspired by this feature-based approach, I experimented with four machine learning model architectures, which show how to utilize information from different pre-trained Transformer layers for sexism detection in social media.

Model Architectures The first model architecture (*Last Hidden*) utilizes only the embeddings of each sequence contained only in the last hidden layer of a pre-trained Transformer with output shape [batch size, max seq len, hidden size]. It takes the first position token embeddings that capture the entire context and are meant for classification, meaning the [CLS] embeddings ([batch size, hidden size]). These embeddings pass from a dropout layer and, then, from a linear layer, which is responsible for classifying the texts. The final model output has dimensions [batch size, number of classes].

The second model architecture (*Concat Last 4 Hidden*) takes the output from all the hidden layers of a Transformer ([initial embeddings + total number of layers, batch size, max seq len, hidden size]) and concatenates only the last four layers into one with output dimensions [batch size, max seq len, hidden size * 4]. The [CLS] token embeddings are taken from the last four hidden layers with output dimension [batch size, hidden size * 4] and pass from a dropout layer. Finally, the output passes from a linear layer having the size [hidden size * 4, number of classes]. The final model output has dimensions [batch size, number of classes]. This was one of the pooling strategies that performed best according to the authors of BERT (Devlin et al., 2018).

The third model architecture is a Bidirectional LSTM network (*Bi-LSTM*) utilizing LSTM pooling. It is adopted from the model architecture introduced for aspect-based sentiment analysis (Song et al., 2020), an extension of it is developed though, as the proposed model is bidirectional. In this way, the model is able to process the input and retain information from both directions. First, it takes the hidden states of the [CLS] token from all layers of a pre-trained Transformer ([initial embeddings + total number of layers, batch size, max seq len, hidden size]). Then, the dimensions of the hidden states are squeezed and converted into [batch size, number of layers, hidden size] to fit into the Bi-LSTM layer. After that, the LSTM is used to connect the [CLS] token representations resulting in getting an output of the last LSTM cell as the final representation with output dimensions [batch size, total number of layers, max seq len * 2]. A dropout layer is applied to the LSTM output. Finally, a linear layer with dimensions of the set maximum sequence length multiplied by two and the number of classes [max seq len * 2, number of classes] is applied to the output from the dropout layer. The final model output has dimensions [batch size, number of classes].

The fourth model architecture utilizes weighted layer pooling (*Weighted Pooling*) that takes the weighted mean of the pre-trained Transformer's different hidden layer representations. It can take the hidden states of the [CLS] token from all layers of a pre-trained Transformer ([initial embeddings + total number of layers, batch size, max seq len, hidden size]). It calculates the weighted average of the combined representations of a selected number of layers and, then, gets and combines the [CLS] token outputs with output dimensions [batch size, hidden size]. After a dropout layer, the final model output of the linear classifier layer is [batch size, number of classes].

Majority Vote Ensemble Learning For sub-task A, the different contextual representations of ALBERT-XXLarge-v2 were employed by developing all four model architectures. For the multi-class sub-tasks B and C, ALBERT-XXLarge-v2, ALBERT-XXLarge-v1, BERT-Large and RoBERTa-Large were fine-tuned hoping that their majority ensemble would yield better results. Each pre-trained Transformer model was trained using all four model architectures presented above. Nevertheless, only the predictions of the models that achieved the highest Macro-F1 score and MCC score on the development sets were aggregated in order to get the majority vote. In other words, the final result for each input text was the class that appeared most frequently among the predicted labels.

Additional Sexist Data In order to deal with the issue of data imbalance and assist models in identifying sexist language online, I decided to collect and feed the models with an additional dataset consisting only of 100 texts, but full of common sexist sentences and phrases in English. Although we may be familiar with those expressions, the models are not. Thus, enriching the knowledge of the models with some commonly used sexist sayings, like *Fight like a girl*, *Men will be men*, *Stop crying like a girl*, *Don't be a pussy* and *She is such a bitch*, was considered to be a good initiative. This dataset was created after having researched several related websites.[1] This dataset and the provided training set were combined and used to train the models solely for sub-task A.

4 Experimental Setup

Environment Setup The proposed system was implemented in Python programming language and the code was written on a Google Colaboratory (Colab) Pro notebook. The experiments were conducted using the *Pytorch* library and NVIDIA A100-SXM4-40GB GPU memory.

Data Used Although both labelled and auxiliary unlabelled data were provided for training, only the labelled data were utilized in each sub-task. Due to the fact that the labelled data were already split by the task organizers into training, development and test sets, no further splitting was considered necessary.

Preprocessing Steps A function including a number of regular expressions and other functions was developed to apply a series of preprocessing steps to the text of the training, development and test sets. The UTF-8 Byte Order Mark (BOM), which identifies a file as being encoded in UTF-8, was deleted and the data were encoded using the *BeautifulSoup* library.[2] The url links and usernames, which were already normalized and set as the anonymous *[URL]* and *[USER]* by the task organizers, were lowercased. The emojis were converted to their textual representation (Taehoon et al., 2022).[3] The *&* and *&* were replaced with *and*. The ASCII encoding apostrophe was replaced with the UTF-8 encoding apostrophe. The presence of certain punctuation marks (full stops, exclamation marks, question marks) was limited up to 3 consecutive characters. The consecutive non-ASCII characters were replaced by whitespace and all extra whitespace was deleted. The *Ekphrasis* library was utilized for hashtag segmentation, correction of spelling, elongated words, unpacking of contracted words as well as tokenization and lowercasing of all words (Baziotis et al., 2017).[4] The hashtags and uppercase words were annotated on both sides with the special tokens <*hashtag*> and <*/hashtag*>, <*/allcaps*> and <*allcaps*> respectively.

Hyperparameter Tuning Firstly, the required input including the input ids and the attention mask in Pytorch tensors was created for the pre-trained Transformer models. The special tokens [SEP] and [CLS] were added and the sequences were padded according to the adjusted maximum sequence length of each training set. Apart from the *RandomSampler* which was used in the train Dataloader of sub-tasks A and B, the *Imbalanced Dataset* Sampler was utilized for the train Dataloaders in sub-tasks B and C because it re-balances the class distribution when sampling from an imbalanced dataset, calculates the sampling weights automatically and avoids creating a new balanced dataset.[5][6] For all development sets, the *SequentialSampler* was utilized. For all classification sub-

[1] https://www.london.gov.uk/what-are-some-common-sexist-phrases-challenge, https://www.insider.com/phrases-that-have-sexist-histories-meanings-2019-3, https://bestlifeonline.com/subtly-sexist-things-people-still-say-at-work/

[2] https://pypi.org/project/beautifulsoup4/

[3] https://pypi.org/project/emoji/

[4] https://github.com/cbaziotis/ekphrasis

[5] https://pytorch.org/docs/stable/data.html

[6] https://github.com/ufoym/imbalanced-dataset-sampler

tasks, dropout and early stopping patience were used so as to prevent model from overfitting. The Binary Cross-Entropy Loss with Logits (*BCEWithLogitsLoss*) was utilized for the binary classification task (sub-task A), while the *Focal Loss* with gamma 2.0 and weights was employed for the multi-class classification of sub-tasks B and C in order to deal with the issue of class imbalance (Lin et al., 2020).[7][8] The weights were calculated for each class so that the models could treat all classes equally during training. *Focal Loss* required the Softmax activation function after the final classifier layer in every model architecture. The *AdamW* was selected as the optimizer. An overview of all model hyperparameters is presented in table 4 of Appendix A.

Evaluation Measures The system efficiency and final ranking was evaluated on the Macro-F1 score of the test sets. The Macro-F1 scores of the development sets assisted in observing the model performance and in tuning the model hyperparameters. Besides F1 score, my system was assessed in terms of Mathews Correlation Coefficient (MCC) and Confusion Matrix.

5 Results

Despite the experimentation to deal with class imbalance, the model fine-tuning and ensemble learning with a view to achieving high system performance, it became evident that the system did not perform as adequately as expected due to the low final rank. The models that achieved the highest Macro-F1 and MCC scores in the development sets were involved in the ensemble and are demonstrated in table 1. A baseline of Macro-F1 and MCC scores was determined based on the total model performance. The models that were not included in the ensemble scored lower than 80%, 60% and 40% Macro-F1 score, while at the same time lower than 60%, 40% and 40% MCC score in sub-tasks A, B and C respectively. From table 1, it is revealed that a different architecture model managed to achieve the highest Macro-F1 and MCC scores in each sub-task. More specifically, the ALBERT-XXLarge-v2 bidirectional LSTM, the RoBERTa-Large Weighted Pooling and the RoBERTa-Large Last Hidden model achieved higher scores than the other models in sub-tasks A, B and C respectively. Nevertheless, my submitted system performed inadequately on the test sets compared to the development sets. This is mainly due to my error during concatenation of the test set predictions with false text rewire ids. In table 2, my recalculated test scores as well as my official test scores in brackets

Sub-task A		
Model	**Macro-F1**	**MCC**
ALBERT-XXLarge-v2 Last Hidden	83.82	67.74
ALBERT-XXLarge-v2 Concat Last 4 Hidden	82.23	64.58
ALBERT-XXLarge-v2 Bi-LSTM	**84.41**	**68.84**
ALBERT-XXLarge-v2 Weighted Pooling	84.33	68.72
Sub-task B		
Model	**Macro-F1**	**MCC**
ALBERT-XXLarge-v2 Last Hidden	66.48	48.98
ALBERT-XXLarge-v1 Last Hidden	62.28	42.02
ALBERT-XXLarge-v1 Bi-LSTM	64.59	46.16
BERT-Large Last Hidden	65.71	48.50
BERT-Large Bi-LSTM	65.11	48.80
BERT-Large Weighted Pooling	64.88	45.16
RoBERTa-Large Last Hidden	69.24	51.19
RoBERTa-Large Weighted Pooling	**70.87**	**55.93**
RoBERTa-Large Weighted Pooling Imb. Dat. Sampler	68.50	50.60
Sub-task C		
Model	**Macro-F1**	**MCC**
ALBERT-XXLarge-v2 Weighted Pooling	40.62	45.20
ALBERT-XXLarge-v1 Last Hidden	41.42	41.36
BERT-Large Last Hidden	48.34	49.90
BERT-Large Bi-LSTM	48.34	49.90
BERT-Large Weighted Pooling	48.93	51.67
RoBERTa-Large Last Hidden	**53.06**	**53.04**
RoBERTa-Large Bi-LSTM	40.55	38.13

Table 1: Evaluation Metrics on Development Sets in %.

[7] https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

[8] https://pypi.org/project/focal-loss-torch/

are illustrated. The correct results were calculated as 82.88%, 63.77% and 43.08% for sub-tasks A, B and C respectively. From table 3, it is evident that certain classes could not be successfully detected due to the low amount of available data. The confusion matrices of the test sets for each sub-task are demonstrated in Appendix A as well.

Macro-F1 Score			**MCC**
Sub-task A			
Majority Vote Ensemble	**Dev Set**	**Test Set**	**Test Set**
4 models Ensemble	84.66	82.88 (50.29)	66.21
Sub-task B			
Majority Vote Ensemble	**Dev Set**	**Test Set**	**Test Set**
9 models Ensemble	74.02	63.77 (22.93)	45.06
Sub-task C			
Majority Vote Ensemble	**Dev Set**	**Test Set**	**Test Set**
7 models Ensemble	54.22	43.08 (08.68)	46.14

Table 2: Final Results from Development and Test Set Submissions in %.

Class	**Macro-F1**
Sub-task A	
Sexist	92.33
Non-Sexist	73.43
Sub-task B	
1.threats	72.41
2.derogation	67.56
3.animosity	61.56
4.prejudiced discussions	53.55
Sub-task C	
1.1 threats of harm	0.5
1.2 incitement and encouragement of harm	64.05
2.1 descriptive attacks	54.75
2.2 aggressive and emotive attacks	51.52
2.3 dehumanising attacks and overt sexual objectification	52.42
3.1 casual use of gendered slurs, profanities and insults	63.31
3.2 immutable gender differences and gender stereotype	52.96
3.3 backhanded gendered compliments	19.04
3.4 condescending explanations or unwelcome advice	0.0
4.1 supporting mistreatment of individual women	13.79
4.2 supporting systemic discrimination against women as a group	52.11

Table 3: Macro-F1 Score of each class on Test Sets in %.

6 Limitations

Class imbalance, especially in sub-tasks B and C, was a major issue during participation in EDOS. The difficulty of the system to distinguish between categories of sexism and between vectors of sexism was apparent. This could be mainly due to the class imbalance or other features of the texts. It is also difficult for a system not only to to detect whether a text is sexist, but also explain why it is sexist based on a plain text without any additional information of the user and the target.

7 Conclusion

The proposed system for SemEval-2023 Task 10 comprised fine-tuning of different contextual representations of Transformer-based pre-trained language models and majority voting ensemble of the best performing models with a view to detect sexist texts from social media and identify the category and sub-category of sexism. Unfortunately, due to a submission error, the Macro-F1 score results from the test set submissions proved to be much lower than expected compared to the development set submissions leading to low rank in the leaderboard. As part of future work, I would incorporate more sexist data in the training set so as to handle the class imbalance and experiment with different Transformer models trained on social media posts, like BERTweet, or more advanced versions of models, such as DeBERTa. Finally, I would experiment with various hyperparameters and pre-processing methods to achieve higher performance.

References

Andrew Arsht and Daniel Etcovitch. 2018. The human cost of online content moderation.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the*

11th International Workshop on Semantic Evaluation (SemEval-2017), pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. 2022. SemEval-2022 task 5: Multimedia automatic misogyny identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 533–549, Seattle, United States. Association for Computational Linguistics.

Jesse Fox, Carlos Cruz, and Ji Young Lee. 2015. Perpetuating online sexism offline: Anonymity, interactivity, and the effects of sexist hashtags on social media. *Computers in Human Behavior*, 52:436–442.

Hannah Rose Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023. SemEval-2023 Task 10: Explainable Detection of Online Sexism. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

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. *CoRR*, abs/1907.11692.

Gina Masequesmay. 2022. Sexism.

Lucina de Meco and Alcy MacKay. 2022. Social media, violence and gender norms: The need for a new digital social contract.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48(12):4730–4742.

Youwei Song, Jiahai Wang, Zhiwei Liang, Zhiyue Liu, and Tao Jiang. 2020. Utilizing BERT intermediate layers for aspect based sentiment analysis and natural language inference. *CoRR*, abs/2002.04815.

Kim Taehoon, Tahir Kevin, Wurster, and Jalilov. 2022. Emoji.

Jessica Valenti. 2022. Toxic twitter - women's experiences of violence and abuse on twitter.

Michelle F. Wright, Bridgette D. Harper, and Sebastian Wachs. 2019. The associations between cyberbullying and callous-unemotional traits among adolescents: The moderating effect of online disinhibition. *Personality and Individual Differences*, 140:41–45. Personality pathologies in the world: beyond dichotomies.

Junjie Yang and Hai Zhao. 2019. Deepening hidden representations from pre-trained language models for natural language understanding. *CoRR*, abs/1911.01940.

A Appendix

Hyperparameters	**Sub-task A**	**Sub-task B**	**Sub-task C**
Number of Classes	1	4	11
Number of Epochs	4	10	10
Sequence Length	190	100	100
Batch Size	16	32	32
Learning Rate	2e-5	2e-5	2e-5
Weight Decay	0.01	0.01	0.01
Warm-up Steps	0	0	0
AdamW Epsilon	1e-8	1e-8	1e-8
AdamW Betas	0.9, 0.999	0.9, 0.999	0.9, 0.999
Dropout	0.3	0.3	0.3
Gradient Clipping	1.0	1.0	1.0
Early Stopping	3	5	5
Random Seed	42	42	42

Table 4: Model Hyperparameters in Each Sub-task.

Sub-task A	
A. Sexist (1)	3,398
B. Non-sexist (0)	10,602
Sub-task B	
1.threats (0)	310
2.derogation (1)	1,590
3.animosity (2)	1,165
4.prejudiced discussions (3)	333
Sub-task C	
1.1 threats of harm (0)	56
1.2 incitement and encouragement of harm (1)	254
2.1 descriptive attacks (2)	717
2.2 aggressive and emotive attacks (3)	673
2.3 dehumanising attacks and overt sexual objectification (4)	200
3.1 casual use of gendered slurs,profanities and insults (5)	637
3.2 immutable gender differences and gender stereotype (6)	417
3.3 backhanded gendered compliments (7)	64
3.4 condescending explanations or unwelcome advice (8)	47
4.1 supporting mistreatment of individual women (9)	75
4.2 supporting systemic discrimination against women as a group (10)	258

Table 5: Categorical & Numerical Labels with Class Distribution in Training Sets.

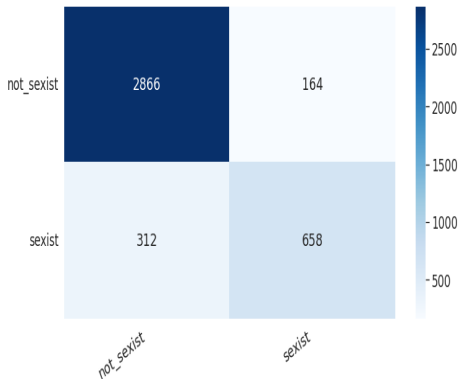

Figure 1: Test Set Confusion Matrix of Sub-task A

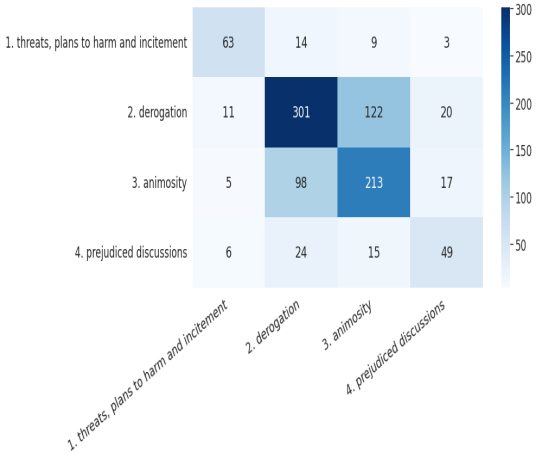

Figure 2: Test Set Confusion Matrix of Sub-task B

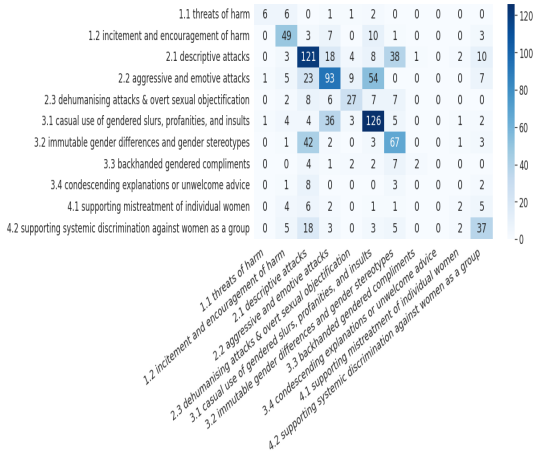

Figure 3: Test Set Confusion Matrix of Sub-task C