SUTNLP at SemEval-2023 Task 10: RLAT-Transformer for explainable online sexism detection

Hamed Hematian Hemati, Sayed Hesam Alavian, Hamid Beigy, Hossein Sameti
AI Group, Computer Engineering Department, Sharif University of Technology, Tehran, Iran
{ hematian.hemati.hamed, hesam.alavian70}@gmail.com, {beigy, sameti}@sharif.edu

Abstract

There is no simple definition of sexism, but it can be described as prejudice, stereotyping, or discrimination, especially against women, based on their gender. In online interactions, sexism is relatively rare but still harmful. One out of ten American adults says that they have been harassed because of their gender and have been the target of sexism, so sexism is a growing issue. The Explainable Detection of Online Sexism shared task in SemEval-2023 aims at building sexism detection systems for the English language. In order to address the problem, we use large language models such as RoBERTa and DeBERTa. In addition, we present a novel method called **R**andom **L**ayer **A**dversarial **T**raining (RLAT) for transformers which is based on adversarial training, and show its impact on boosting all subtasks' scores. Moreover, we use other discriminative and generalization techniques for subtask A to boost performance. Using our methods to make predictions over subtask A, B, and C test sets, we obtained macro-F1 of 84.45, 67.78, and 52.52 respectively, outperforming proposed baselines on all subtasks. Our code is publicly available on Github.[1]

1 Introduction

Sexism refers to any negative, abusive, or discriminatory behavior, attitude, or language that targets individuals based on their gender, particularly women, and involves prejudice, stereotyping, or discrimination (Kirk et al., 2023). In essence, sexism encompasses any negative, abusive, or discriminatory behavior or attitude that targets women based on their gender or a combination of their gender with other identity characteristics such as race, religion, or gender identity, for instance, black women, Muslim women, or trans women (Kirk et al., 2023). The internet is a breeding ground for sexism, with approximately one in ten American adults reporting that they have been harassed due to their gender, and have experienced sexist behavior (Swim et al., 2001).

In order to promote a fair and inclusive online environment, it is vital to acknowledge and combat sexism on online platforms because of their significant scale, extensive reach, and powerful influence.[2]

Sexism poses a growing problem in the online environment, and its consequences can be severe, such as making online spaces inaccessible and perpetuating social injustices. To address this issue, various automated tools have been deployed to detect and assess sexist content. Nonetheless, most of these tools only provide generic classifications without further explanation. Identifying sexist content and explaining why it is offensive will increase an automated tool's interpretability, trustworthiness, and understanding of its decisions. This, in turn, will empower both users and moderators to make informed decisions and create a fair and inclusive online environment (Kirk et al., 2023). The aim of the SemEval-2023 Explainable Detection of Online Sexism shared task is to develop and implement sexism detection systems for the English language (Kirk et al., 2023).

In order to deal with the problem, we use large language models such as RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020). We introduce a novel method called Random Layer Adversarial Training (RLAT) based on famous adversarial training and demonstrate that it has a positive impact on boosting performance across all subtasks. To the best knowledge of the authors, this is the first research using random layers of a network for adversarial training.

As part of subtask A, we also use virtual adver-

[1] https://github.com/SUTNLP/RLAT-Transformer

[2] https://www.theguardian.com/commentisfree/2015/dec/16/online-sexism-social-media-debate-abuse

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 347–356

sarial training and contrastive learning to improve the model's performance.

In subtask A, our team ranked 18th out of 84 teams, in subtask B, 9th out of 69 teams, and in subtask C, we ranked 6th out of 63 teams. In subtasks A and B, we outperform baseline methods by a substantial margin, and in subtask C, we outperform baseline by a large margin using the RLAT. Several experiments were conducted to examine the effect of the RLAT on the current task.

2 Background

2.1 Related Work

Due to the widespread prevalence of sexism in online communication, the automated detection of online sexism has become an increasingly important area of research. Various machine learning models have been proposed for detecting sexism in online texts, such as tweets, comments, and other forms of online communication (Waseem and Hovy, 2016; Davidson et al., 2017). These models have been evaluated and compared in terms of their accuracy and performance. Some other studies, such as Bolukbasi et al. (2016); Sun et al. (2019) have also examined the potential for gender bias in these models and put forth strategies to address it. Rodríguez-Sánchez et al. (2020) proposed the use of deep learning approaches to detect sexist attitudes and stereotypes in tweets. Samory et al. (2021a) focused on construct validity and reliability in sexism detection. Ghosh Chowdhury et al. (2019) creates a dataset for detection of personal recollections of sexual harassment from tweets. They further tested several models on their dataset. In other research, Parikh et al. (2019) introduced a dataset consisting of accounts of sexism in 23 categories to investigate sexism categorization as a multi-label classification task.

2.2 EDOS task

The task of EDOS comprises three hierarchical subtasks:

- SUBTASK A - Binary Sexism Detection involves a two-class (or binary) classification, where the goal of the system is to predict whether a given post is sexist or not sexist.
- SUBTASK B - Category of Sexism which involves a four-class classification for sexist posts: threats, derogation, animosity, and prejudiced discussions
- SUBTASK C - Fine-grained Vector of Sexism task involves a classification of 11 classes for posts that contain sexism, where systems are required to predict one of the 11 fine-grained vectors.

2.3 Preliminaries

In this sub-section, we will give a brief introduction to the methods and approaches we used for our text classification system.

2.3.1 Large Pre-trained Language Models

Recent research demonstrates that pre-trained language models, such as the transformers approach, are effective in text classification (Fallah et al., 2022). The BERT model has emerged as a popular state-of-the-art model in recent years (Devlin et al., 2018). This language model is capable of handling NLP tasks, including text classification (González-Carvajal and Garrido-Merchán, 2020).

Transformer models have become the most effective neural network architecture for neural language modeling. A new model architecture known as DeBERTa (Decoding-enhanced BERT with disentangled attention) improves the BERT and RoBERTa models using two novel techniques. The first is the disentangled attention mechanism, where each token is represented using two vectors that encode its content and position, respectively. The attention weights among tokens are computed using disentangled matrices of their contents and relative positions, respectively. Second, an enhanced mask decoder is used to incorporate absolute positions in the decoding layer to predict the masked tokens in model pre-training (He et al., 2020).

2.3.2 Adversarial Training

A machine learning technique known as adversarial training aims to increase the robustness and generalization of a model by training it on adversarial examples. These are inputs deliberately designed to cause misclassification or errors. As a result, perturbations to the input data are generated and added to the training set in order to create a more challenging and diverse set of data (Goodfellow et al., 2014). In recent years, adversarial training has gained popularity in the field of natural language processing (NLP) for text classification tasks (Miyato et al., 2016), which applied adversarial training to improve the performance of semi-supervised text classification models. The method has been applied to a variety of NLP applications, including

sentiment analysis, spam detection, topic modeling, hate speech, and sexism detection. Some notable works in this area include Iyyer et al. (2018), which proposed a method for generating syntactically controlled adversarial examples for text classification, Wu et al. (2017a), which demonstrated the effectiveness of adversarial training for improving the robustness of relation extraction models, and Zhu et al. (2021) which used adversarial training to make the BERT model more robust and generalized. Zhu et al. (2019) showed the impact of adversarial training on boosting the performance of several benchmarks. Vidgen et al. (2021) proposed a human-and-model-in-the-loop process for dynamically generating datasets through teaching annotators to generate adversarial examples of hate which is challenging for current models to discern. Further they trained more effective and robust hate detection models. Kirk et al. (2022) built a dataset using adversarial examples. They showed that models trained on this dataset perform significantly better at detection of emoji-based hate compared to previous methods while retaining good performance at detection of text-only hate. Samory et al. (2021b) generated adversarial examples from annotated datasets to test sexism detection models' reliability.

2.3.3 Virtual Adversarial Training

Adversarial training is a technique used to improve the robustness of supervised learning algorithms by adding perturbations to the input data. On the other hand, virtual adversarial training extends supervised learning algorithms to semi-supervised settings by generating adversarial examples in the unlabeled data space. In other words, while adversarial training focuses on regularizing supervised learning algorithms, virtual adversarial training leverages both labeled and unlabeled data to enhance the performance of these algorithms in semi-supervised settings. Recent studies have demonstrated that virtual adversarial training has achieved state-of-the-art results in various benchmark semi-supervised tasks (Miyato et al., 2016). This approach has also been utilized in various machine learning applications, including supervised and semi-supervised learning (Miyato et al., 2019), as well as sequence labeling (Chen et al., 2020a).

2.3.4 Contrastive Learning

A popular method for learning embedding spaces is contrastive learning, which ensures that pairs of data samples with similar labels are represented closely and pairs with dissimilar labels are represented at a greater distance. In supervised or unsupervised settings, it can produce task-specific or general-purpose representations using different loss functions (Zhang et al., 2022). This technique has previously been used in research to perform various specific tasks such as learn sentence embeddings (Gao et al., 2021; Fang et al., 2020), classify text by producing better text representations using contrastive samples (Du et al., 2021), and to solve hierarchical text classification problems (Wang et al., 2022).

3 System Overview

Overall we developed our models based on large pre-trained language models and random layer adversarial training. Subtask A is built on random layer adversarial training, virtual adversarial training and contrastive learning. Subtasks B and C are solved similarly and are dependent only on random layer adversarial training. So we describe our system in two different sections.

3.1 Subtask A

3.1.1 Base Transformer

As the base transformer for this subtask, we use RoBERTa-Large. To calculate the loss of each batch of training data, we employ cross-entropy loss. Considering the imbalance of the training dataset, sample-based loss weighting was adopted to overcome this issue by weighting samples based on the inverse ratio of their class frequency in the training dataset. Using L_{CE}, we present cross-entropy loss in Equation 5.

3.1.2 Random Layer Adversarial Training

Several researchers in the field of NLP have used adversarial training to enhance generalization of their model. These reseraches mostly apply adversarial training to the embedding layer (Zhu et al., 2021; Liu et al., 2020; Wang et al., 2019; Zhu et al., 2019; Cui et al., 2022a; Lu et al., 2022). Sankaranarayanan et al. (2018) shows intermediate layers could be quite effective in further regularization and boost of performance for adversarial training to be applied to.

Through experiments on development data, we observe that a random layer perturbation method works best for the current task. In order to do that we devise an algorithm called Random Layer Adversarial Training (RLAT) which employs different

random layers to which adversarial training is applied. Suppose the i-th dense layer of network is denoted by H_i and word embedding layer is denoted by W_E. According to the Equation 1, in each batch a random layer is uniformly selected for adversarial training. The selected random layer is denoted by A_L.

$$A_L = Uniform(\{H_1, \cdots, H_L, W_E\}) \quad (1)$$

The RLAT method tries to find the optimal parameters θ^* by minimizing the maximum possible adversarial perturbation δ to the outputs of a random layer, inside a norm ball of ϵ which is stated as follows:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\mathcal{D},\mathcal{U}} \left\{ \max_{\delta:\|\delta\|\leq\epsilon} L(f_\theta(X+\delta), Y) \right\} \quad (2)$$

In Equation 2, network parameters are denoted by θ, $\mathcal{D}$ and $\mathcal{U}$ are dataset distribution and the uniform distribution on random layer selection respectively. X is the random layer output and Y represents the label. f_θ is the network function and L represents the loss function. K-projected gradient descent (K-PGD) (Madry et al., 2017) is adopted to train the network using adversarial training. K is a hyperparameter which is tuned using development data.

3.1.3 Virtual Adversarial Training (VAT)

To further boost the transformer's generalization capability, VAT is adopted as another objective in the training process of our model. We use an algorithm which is developed based on VAT called SMART (Jiang et al., 2019). SMART is applied to the embedding layer. We denote the loss of VAT by L_{VAT}.

Virtual adversarial training adds a perturbation to the input and minimizes the distance between the output of the original input and the output of the perturbed input. The perturbation is added in a way so that the distance between the two outputs is maximized. Contrary to adversarial training, this method does not use label information. According to Miyato et al. (2019), virtual adversarial loss can be formulated like Equation 3.

$$\begin{aligned} L_{VAT} &= D[f_\theta(X), f_\theta(X+\delta_{vadv})] \\ s.t.\ \delta_{vadv} &= \arg\max_{\delta:\|\delta\|\leq\epsilon} D[f_\theta(X), f_\theta(X+\delta)] \end{aligned} \quad (3)$$

In Equation 3, X is the input, f_θ and θ are the function and parameters of the network respectively, D is a distance metric, δ_{vadv} is the perturbation and, ϵ is the a constraint for the maximum 2-norm of the perturbation.

3.1.4 Contrastive Training (CON)

We employ contrastive training to further boost the discrimination power of our model. (Cui et al., 2022b) shows that contrastive training is effective when applied to the last hidden layer of the network. Kaku et al. (2021) shows that contrastive training could also be effective when applied to intermediate layers beside the final layer. Following Kaku et al. (2021), we use contrastive training not only to the final layer but also to some intermediate layers. We do this by applying NT-XENT loss (Chen et al., 2020b) to the hidden representation of the last L hidden layers of the transformer. According to the NT-XENT loss, if samples with indices of i and j have the same label in a batch, their contrastive loss is computed by the Equation 4, where l_{ij} denotes the contrastive loss and $2N$ is the size of batch. L is a hyperparameter which is tuned using development data. The overall contrastive loss is denoted by L_{CON}.

$$l_{ij} = \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp(sim(z_i, z_k)/\tau)} \quad (4)$$

3.1.5 Overall

To train the transformer for each batch all losses including L_{CE}, L_{RAT}, L_{VAT}, and L_{CON} are calculated and the total loss of L_T is calculated using Equation 5. λ_1, λ_2, and λ_3 are hyperparameters that are tuned using development data.

$$L_T = L_{CE} + \lambda_1 L_{RLAT} + \lambda_2 L_{VAT} + \lambda_3 L_{CON} \quad (5)$$

3.2 Subtasks B and C

DeBERTa-v3-Large is used as the transformer for both subtasks. We adopt the RLAT as described in 3.1.2 alongside cross-entropy for both subtasks. The total loss for both subtasks B and C is computed as follows:

$$L_T = L_{CE} + L_{RLAT} \quad (6)$$

4 Experimental Setup

4.1 Dataset and Evaluation

According to the organizers, the dataset consists of 20,000 samples where each sample is a comment collected from Reddit or Gab (Kirk et al., 2023). Each sample is tagged by three annotators. For subtask A, when the annotators' votes are unanimous in favour of one label, this is taken as the gold label. If there is any disagreement, one of the experts reviews the entry and decides the gold label. For subtasks B and C, when at least two annotators agree on a label, this is taken as the gold label but in cases of 3-way disagreement, one of the experts decides the gold label (Kirk et al., 2023). The number of train, development and test set samples are shown in Table 1 for each subtask.

Data	Train	Development	Test
Subtask A	14,000	2,000	4,000
Subtask B	3,398	486	970
Subtask C	3,398	486	970

Table 1: Number of train/development/test samples for each subtask

For subtask A, a comment is given and the objective is to classify it as sexist or not sexist. For subtasks B and C, a sexist comment is given and the objective is to classify the comment as one of the sexist categories. Datasets for subtasks B and C are highly unbalanced presenting a new challenge for our method to handle. Figures 2c and 2b present the sample count of each class in subtasks B and C respectively (names of labels in figures are reduced to the first number of classes to avoid using too much space). In subtask B, the ratio of the most frequent class to the least frequent is approximately 5 and for subtask C, this ratio is much severe and is about 14. This obviously indicates that as we dig deeper into subtasks, not only are there fewer data for each class, but the data imbalance worsens as well. There are 2, 4, and 11 classes for subtasks A, B, and C respectively (Kirk et al., 2023).

To estimate the performance of the system, the organizers employ macro-F1 score as the main metric for all subtasks (Kirk et al., 2023).

4.2 Parameter Settings

We use Huggingface[3] transformer models and their respective checkpoints. For all subtasks, AdamW (Loshchilov and Hutter, 2017) algorithm is adopted as the optimization algorithm. For each subtask we adopt early stopping with patience of 4 and after training the model checkpoint with the best performance on development set is used for prediction on test set. All of the later mentioned hyperparameters are set using development set for each subtask.

4.2.1 Choice of Base Transformers

The base transformer for each subtask is chosen between RoBERTa-Large and DeBERTa-Large. The choice is based on the performance of the base transformer on the development data of each subtask. Consequently RoBERTa-Large is adopted for subtask A and DeBERTa-v3-Large is adopted for subtasks B and C.

4.2.2 Subtask A

We train for 10 epochs. Learning rate is set to $4e{-}6$ and the weight decay is $1e-2$, L is set to 4. K is set to 2 for PGD. λ_1 is set to 1, λ_2 and λ_3 are set to 0.5 and 0.4 respectively.

4.2.3 Subtask B

We train the model for 30 epochs. Learning rate is set to $1e-5$ and weight decay is $9e-3$, K is set to 2 for PGD.

4.2.4 Subtask C

We train the model for 30 epochs. Learning rate is set to $6e-6$ and weight decay is $7e-3$, K is set to 4 for PGD.

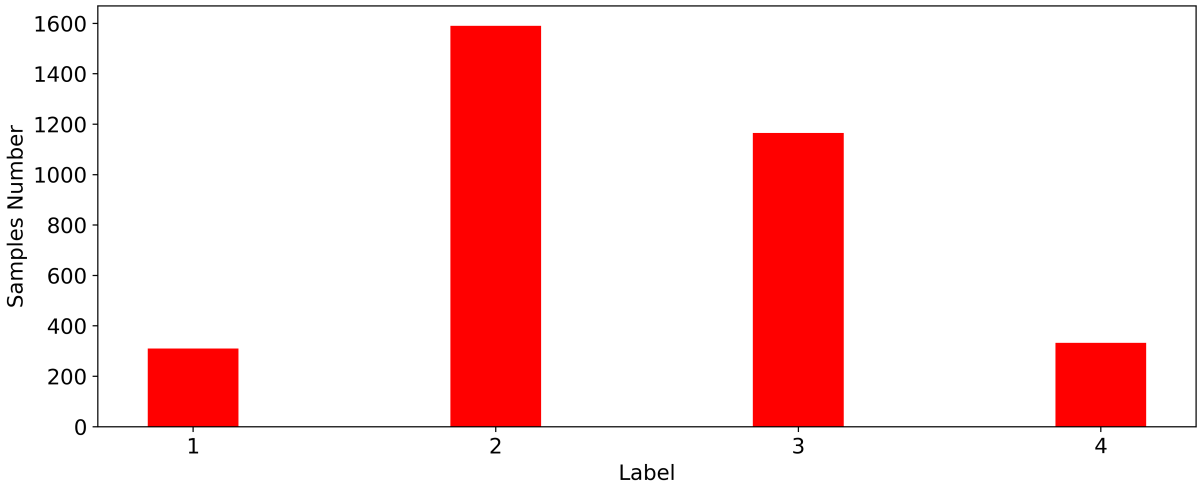
5 Results

5.1 Model Performance

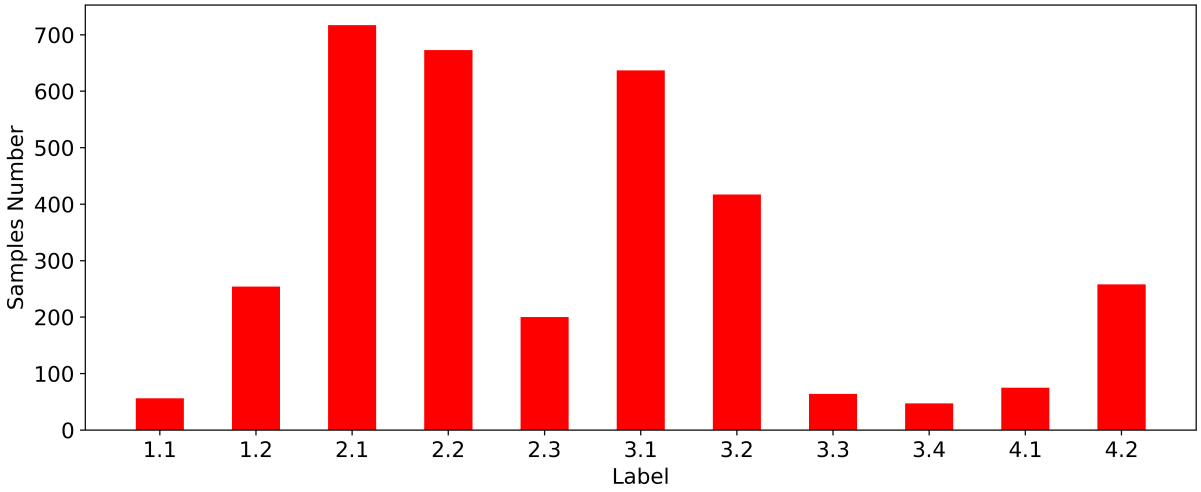
As the organizers of the task have not proposed a baseline, for each subtask we set the base transformer of the subtask with only cross-entropy loss as the baseline method for the subtask. Hence RoBERTa-Large is used for subtask A baseline and DeBERTa-v3-Large is used for subtasks B and C baselines. All of the following results are based on the test data.

Results of subtask A are shown in Table 2. All of RLAT, VAT, and CON contribute positively to boost the performance of the model. As can be seen excluding any of them results in a drop in performance. RLAT contributes most to the boost as its exclusion has the largest drop. Tables 3 and 4 provide results of subtask B and C, respectively. It is evident that adding RLAT has a huge effect on

[3] https://huggingface.com/



(a) Subtask B



(b) Subtask C

Figure 1: Distribution of classes for Subtasks B and C in train data

Model	macro-F1	macro-Precision	macro-Recall
RoBERTa-Large (Baseline)	84.81	85.08	84.56
RoBERTa-Large + RLAT + VAT	85.05	85.20	84.92
RoBERTa-Large + VAT + CON	85.03	85.18	84.88
RoBERTa-Large + RLAT + CON	85.11	85.15	**85.08**
RoBERTa-Large + RLAT + VAT + CON	**85.45**	**85.86**	85.07

Table 2: Subtask A results

Model	macro-F1	macro-Precision	macro-Recall
DeBERTa-v3-Large	66.70	65.90	**67.77**
DeBERTa-v3-Large + RLAT	**67.78**	**69.16**	67.75

Table 3: Subtask B results

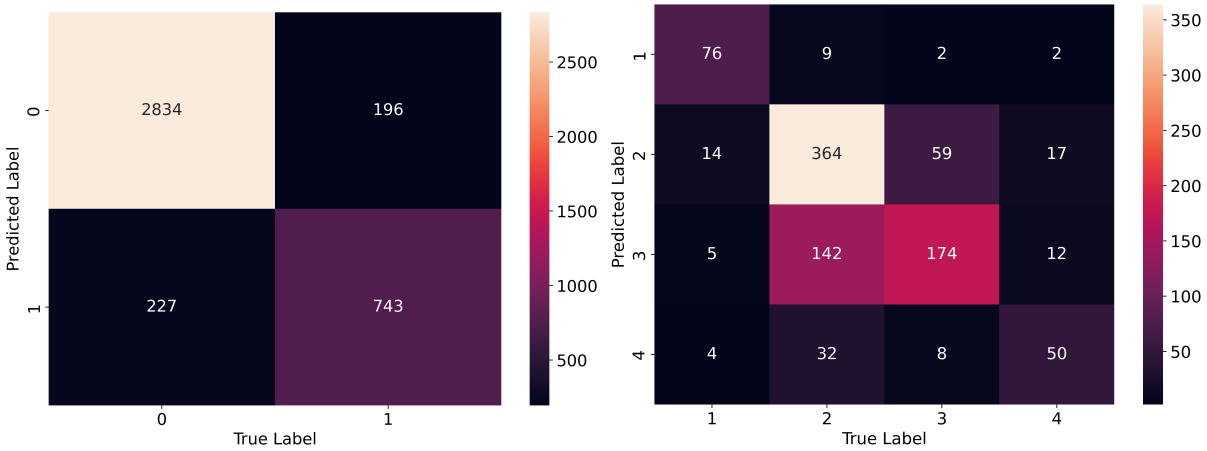
the performance of both subtasks. Further, confusion matrices for subtasks A, B, and C are depicted in Figure 2. In the confusion matrix of subtask B, we observe that the model is unable to accurately

Model	macro-F1	macro-Precision	macro-Recall
DeBERTa-v3-Large	45.40	47.21	45.08
DeBERTa-v3-Large + RLAT	**52.52**	**53.67**	**52.89**

Table 4: Subtask C results

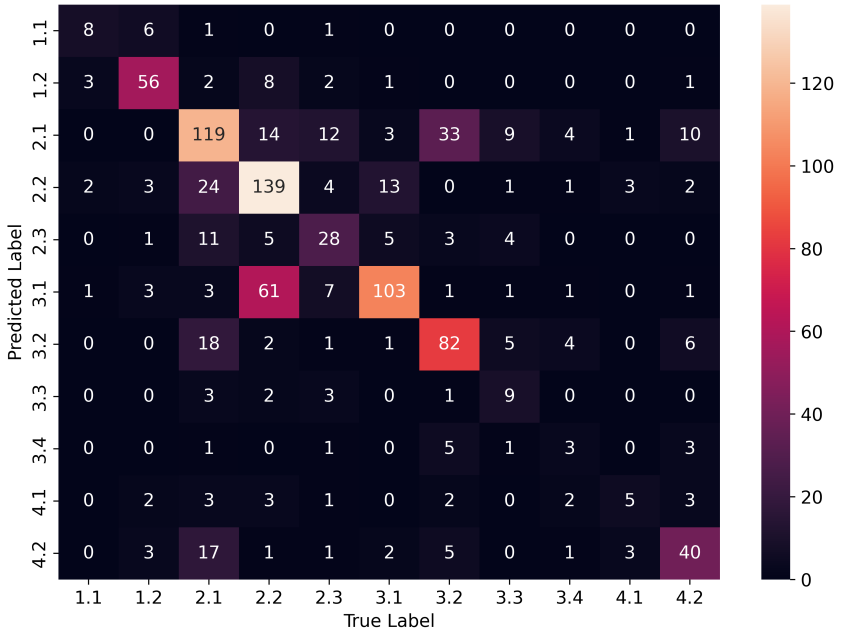
Model	Subtask B macro-F1	Subtask C macro-F1
DeBERTa-v3-Large	66.70	45.40
DeBERTa-v3-Large + AT (embedding layer)	66.40(−0.30)	48.61(+3.21)
DeBERTa-v3-Large + RLAT	67.78(+1.08)	52.52(+7.12)

Table 5: RLAT Analysis



(a) Subtask A

(b) Subtask B



(c) Subtask C

Figure 2: Confusion matrix on test data for Subtasks A, B, and C

discriminate between classes 2 (Derogation) and 3 (Animosity). A closer look at the confusion matrix of subtask C reveals more information. As we see a significant portion of samples with label 3.2 (Immutable gender stereotypes) are classified as class 2.1 (Descriptive attacks), and a huge portion of samples with label 2.2 (Aggressive and emotive attacks) are classified as class 3.1 (Casual use of gendered slurs, profanities insults).

5.2 RLAT Analysis

In order to better understand the RLAT method, here we try to analyze it. The results for the RLAT method are shown in Table 5. Many researchers use the embedding layer to conduct adversarial training with transformers (Dong et al., 2020; Wu et al., 2017b; Ju et al., 2019; Zhu et al., 2021; Liu et al., 2020; Wang et al., 2019; Zhu et al., 2019; Cui et al., 2022a; Lu et al., 2022). To this end we compare the RLAT method with the method in which Adversarial Training (AT) is applied only to the embedding layer. For subtask B, when adversarial training is applied to the embedding layer, our model experiences a $-0.3\,\%$ drop in Macro-F1 score in comparison to the baseline. When RLAT is applied our model experiences a $+1.08\,\%$ gain in Macro-F1 score compared to the baseline. For subtask C, our model with RLAT experiences a $+7.12\,\%$ gain in Macro-F1 score in comparison to the baseline which is higher by $+3.91\,\%$ compared to the case when AT is applied to the embedding layer. This analysis shows the superiority of using RLAT in comparison to conventional AT.

6 Conclusion

This paper presents SUTNLP's submission to SemEval-2023 Task 10 "Explainable Detection of Online Sexism" competition. To solve this problem, we use transformers and further propose Random Layer Adversarial Training (RLAT) to boost the performance of the base models by large margins. We conducted experiments to show the superiority of RLAT compared to conventional adversarial training used in NLP for the current task. For all subtasks, the performance of the model is evaluated by macro-F1. Using macro-F1 criteria, appending the RLAT to base transformers shows gains of $1.08\,\%$ in Macro-F1 and $7.12\,\%$ Macro-F1 for subtasks B and C respectively. In the future, we intend to analyse the RLAT method to determine how it performs on other NLP tasks and further improve it.

References

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. 2020a. SeqVAT: Virtual adversarial training for semi-supervised sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8801–8811, Online. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xuange Cui, Wei Xiong, and Songlin Wang. 2022a. Zhichunroad at semeval-2022 task 2: Adversarial training and contrastive learning for multiword representations. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 197–203.

Xuange Cui, Wei Xiong, and Songlin Wang. 2022b. ZhichunRoad at SemEval-2022 task 2: Adversarial training and contrastive learning for multiword representations. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 197–203, Seattle, United States. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Xin Dong, Yaxin Zhu, Yupeng Zhang, Zuohui Fu, Dongkuan Xu, Sen Yang, and Gerard De Melo. 2020. Leveraging adversarial training in self-learning for cross-lingual text classification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1541–1544.

Yangkai Du, Tengfei Ma, Lingfei Wu, Fangli Xu, Xuhong Zhang, Bo Long, and Shouling Ji. 2021. Constructing contrastive samples via summarization for text classification with limited annotations. *arXiv preprint arXiv:2104.05094*.

Haytame Fallah, Patrice Bellot, Emmanuel Bruno, and Elisabeth Murisasco. 2022. Adapting transformers for multi-label text classification. In *CIRCLE (Joint Conference of the Information Retrieval Communities in Europe) 2022*.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Arijit Ghosh Chowdhury, Ramit Sawhney, Rajiv Ratn Shah, and Debanjan Mahata. 2019. #YouToo? detection of personal recollections of sexual harassment on social media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2527–2537, Florence, Italy. Association for Computational Linguistics.

Santiago González-Carvajal and Eduardo C Garrido-Merchán. 2020. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on conversational question answering. *arXiv preprint arXiv:1909.10772*.

Aakash Kaku, Sahana Upadhya, and Narges Razavian. 2021. Intermediate layers matter in momentum contrastive self supervised learning. *Advances in Neural Information Processing Systems*, 34:24063–24074.

Hannah Kirk, Bertie Vidgen, Paul Rottger, Tristan Thrush, and Scott Hale. 2022. Hatemoji: A test suite and adversarially-generated dataset for benchmarking and detecting emoji-based hate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1352–1368, Seattle, United States. Association for Computational Linguistics.

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 (SemEval-2023)*. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

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*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Junyu Lu, Hao Zhang, Tongyue Zhang, Hongbo Wang, Haohao Zhu, Bo Xu, and Hongfei Lin. 2022. Guts at semeval-2022 task 4: Adversarial training and balancing methods for patronizing and condescending language detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 432–437.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Takeru Miyato, Shin-Ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993.

Pulkit Parikh, Harika Abburi, Pinkesh Badjatiya, Radhika Krishnan, Niyati Chhaya, Manish Gupta, and Vasudeva Varma. 2019. Multi-label categorization of accounts of sexism using a neural framework. 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)*, pages 1642–1652, Hong Kong, China. Association for Computational Linguistics.

Francisco Rodríguez-Sánchez, Jorge Carrillo-de Albornoz, and Laura Plaza. 2020. Automatic classification of sexism in social networks: An empirical study on twitter data. *IEEE Access*, 8:219563–219576.

Mattia Samory, Indira Sen, Julian Kohne, Fabian Flöck, and Claudia Wagner. 2021a. "call me sexist, but...": Revisiting sexism detection using psychological scales and adversarial samples. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 573–584.

Mattia Samory, Indira Sen, Julian Kohne, Fabian Flöck, and Claudia Wagner. 2021b. "call me sexist, but..." : Revisiting sexism detection using psychological scales and adversarial samples. *Proceedings of the International AAAI Conference on Web and Social Media*, 15(1):573–584.

Swami Sankaranarayanan, Arpit Jain, Rama Chellappa, and Ser Nam Lim. 2018. Regularizing deep networks using efficient layerwise adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. *arXiv preprint arXiv:1906.08976*.

Janet K Swim, Lauri L Hyers, Laurie L Cohen, and Melissa J Ferguson. 2001. Everyday sexism: Evidence for its incidence, nature, and psychological impact from three daily diary studies. *Journal of Social issues*, 57(1):31–53.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, Online. Association for Computational Linguistics.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving neural language modeling via adversarial training. In *International Conference on Machine Learning*, pages 6555–6565. PMLR.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022. Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. *arXiv preprint arXiv:2203.03825*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Yi Wu, David Bamman, and Stuart Russell. 2017a. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783.

Yi Wu, David Bamman, and Stuart Russell. 2017b. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783, Copenhagen, Denmark. Association for Computational Linguistics.

Rui Zhang, Yangfeng Ji, Yue Zhang, and Rebecca J. Passonneau. 2022. Contrastive data and learning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*, pages 39–47, Seattle, United States. Association for Computational Linguistics.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*.

Danqing Zhu, Wangli Lin, Yang Zhang, Qiwei Zhong, Guanxiong Zeng, Weilin Wu, and Jiayu Tang. 2021. At-bert: Adversarial training bert for acronym identification winning solution for sdu@ aaai-21. *arXiv preprint arXiv:2101.03700*.