# PingAnLifeInsurance at SemEval-2023 Task 10: Using Multi-Task Learning to Better Detect Online Sexism

Mengyuan Zhou, Xiaolong Hou, Meizhi Jin, Xiyang Du, Cheng Chen, Lianxin Jiang, Jianyu Li and Zhenggang Wei Ping An Life Insurance Company of China, Ltd.

{ZHOUMENGYUAN425, HOUXIAOLONG430, JINMEIZHI005, DUXIYANG037, CHENCHENG498, JIANGLIANXIN769, LIJIANYU002, WEIZHENGGANG617} @pingan.com.cn

### Abstract

This paper describes our system used in the SemEval-2023 Task 10: Towards Explainable Detection of Online Sexism (Kirk et al., 2023). The harmful effects of sexism on the internet have impacted both men and women, yet current research lacks a fine-grained classification of sexist content. The task involves three hierarchical sub-tasks, which we addressed by employing a multitask-learning framework. To further enhance our system's performance, we pre-trained the roberta-large (Liu et al., 2019b) and deberta-v3-large (He et al., 2021) models on two million unlabeled data, resulting in significant improvements on sub-tasks A and C. In addition, the multitask-learning approach boosted the performance of our models on subtasks A and B. Our system exhibits promising results in achieving explainable detection of online sexism, attaining a test f1-score of 0.8746 on sub-task A (ranking 1st on the leaderboard), and ranking 5th on sub-tasks B and C.

## 1 Introduction

The use of sexist language and behavior in online spaces has long been a problem, with harmful effects on those who are targeted. This issue has become even more pronounced in recent years, as the internet has become a platform for social and political discourse, making it easier for sexism to be disseminated on a large scale (Plaza et al., 2023).

Automated tools have been developed to address this issue, with many platforms now using machine learning models to detect and assess sexist content at scale (Lopez-Lopez et al., 2021). However, most of these tools only provide generic, high-level classifications for sexist content, without further explanation. This lack of interpret ability can be problematic, as it can make it difficult for users and moderators to understand how automated tools make their decisions.

To address this issue, In Task 10, we need to develop English-language models for sexism detec-

tion that are not only more accurate, but also more explainable (Kirk et al., 2023). To achieve this, the official provides not only supervised data but also 2 million unlabeled data for sexist content classifications from two prominent online platforms: Gab and Reddit.

The task consists of three hierarchical subtasks. The first subtask, Binary Sexism Detection, is a two-class (or binary) classification task in which systems must predict whether a post is sexist or not sexist. The second subtask, Category of Sexism, is designed for posts identified as sexist in the first subtask. Here, systems must predict one of four categories: threats, derogation, animosity, or prejudiced discussions. The final subtask, Fine-grained Vector of Sexism, is also for posts identified as sexist in the first subtask. This subtask requires systems to predict one of 11 fine-grained vectors.

The development of more accurate and explainable models for detecting sexism online has the potential to make a significant impact, not only by flagging sexist content, but also by providing users and moderators with explanations for why such content is considered sexist (). This, in turn, can help to build trust in automated tools and make online spaces more inclusive and welcoming for all.

MT-DNN (Multi-Task Deep Neural Network) (Liu et al., 2019a) is a neural network-based multitask learning framework designed to address multiple natural language processing tasks simultaneously using a single neural network model. In this task, we consider hierarchical classification as a multi-task learning job within this framework. Compared to single-task learning, multi-task learning can leverage the relatedness and complementarity among tasks to enhance model generalization (Hashimoto et al., 2017). MT-DNN achieves multi-task learning by sharing a common bottomlevel language representation and task-specific toplevel structures.The MT-DNN model comprises

2188

a base module, which is a bidirectional Transformer encoder, and multiple task-specific top-level structures. The bottom-level encoder generates sentence-level language representations, while the top-level structures are responsible for processing different tasks (Peters et al., 2018).

Based on the MT-DNN structure, we can build a new system that is suited for hierarchical tasks. We employed the RoBERTa and DeBERTa-v3 models as our base models for multi-task learning. Following this, we designed distinct head layers for three sub-tasks, and subsequently trained all tasks within the same model. The advantage of this framework lies in its ability to effectively utilize data from multiple tasks to train a more generalizable and adaptable model while avoiding the problem of insufficient generalization or overfitting(Guo et al., 2018). Our system achieved a macro F1 score of 0.8746 in sub-task A, ranking first among all competitors. Additionally, our system achieved macro F1 scores of 0.7094 and 0.5308 in subtasks B and C, respectively.

# 2 Task Description

The dataset used in this study contains 20,000 posts in English, collected from two platforms - Reddit and Gab. The posts were evenly split between the two platforms, with 10,000 posts from each. Gab is a social network known for having a user base with far-right views. The dataset was divided into three subsets, with a split ratio of 0.7:0.1:0.2. This means that 14,000 posts were used for training the model, 2,000 were used for testing model in develop phase, and 4,000 were used for final evaluation.

We have examined the class distribution for each task, and it is apparent that the data-sets for subtask A, sub-task B, and sub-task C are imbalanced, with differing numbers of posts available for training and testing. During the testing phase, sub-task A has 16,000 available posts for training. In contrast, sub-task B and sub-task C have only 3,884 available posts for training. This unequal distribution is observed in all three data-sets, as illustrated in Figure 1.

As illustrated in Figure 2, Figure 3 and Figure 4, we observe imbalanced class distributions across all three sub-tasks. In sub-task A, which involves binary classification, the sexist class and not-sexist class. Not-sexist class is the majority classes. For Task B, it is a multi-task classification task consisting of four categories: "1. threats, plans to harm and incitement", "2. derogation", "3. animosity", and "4. prejudiced discussions". The majority category is "2. derogation". The majority classes is "2. derogation". In Task C, which is also a multi-task classification task, there are 11 categories: "1.1 Threats of harm", "1.2 Incitement and encouragement of harm", "2.1 Descriptive attacks", "2.2 Aggressive and emotive attacks", "2.3 Dehumanizing attacks and overt sexual objectification", "3.1 Causal use of gendered slurs, profanities, and insults", "3.2 Immutable gender differences and gender stereotypes", "3.3 Backhanded gendered compliments", "3.4 Condescending explanations or unwelcome advice", "4.1 Supporting mistreatment of individual women", and "4.2 Supporting systemic discrimination against women as a group". The majority classes is "2.1 Descriptive attacks".

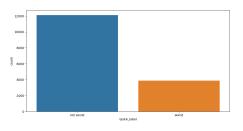


Figure 1: Sub-task A Label Distribution.

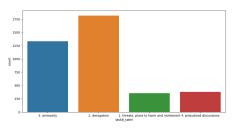


Figure 2: Sub-task B Label Distribution.

# 3 Methodology

Our team has developed a system for this task that utilizes the MT-DNN (Multi-Task Deep Neural Network) multi-task learning framework. This framework enables us to train a single neural network model to perform multiple related tasks simultaneously, allowing the model to share knowledge and leverage correlations between tasks to improve performance.

To enhance the performance of our model, we additionally pre-trained it on 2 million unlabeled data collected from Gab and Reddit using the RoBERTalarge and DeBERTa-v3-large models with 5, 7,

Sub-Task	Further Pretrained Epoch	Pretrained model	F1 Score in Test Phase
Sub-Task A	5	roberta-large	0.8669
	7	roberta-large	0.8680
	9	roberta-large	0.8646
	5	deberta-v3-large	0.8718
	7	deberta-v3-large	0.8740
	9	deberta-v3-large	0.8746
Sub-Task B	5	roberta-large	0.7171
	7	roberta-large	0.7194
	9	roberta-large	0.7184
	5	deberta-v3-large	0.718
	7	deberta-v3-large	0.7212
	9	deberta-v3-large	0.7171
Sub-Task C	5	roberta-large	0.5225
	7	roberta-large	0.5338
	9	roberta-large	0.5278
	5	deberta-v3-large	0.5308
	7	deberta-v3-large	0.5605
	9	deberta-v3-large	0.5426
Sub-Task A Multi model Fusion	9	deberta-v3-large	0.8751

Table 1: Experiment results for Task 10

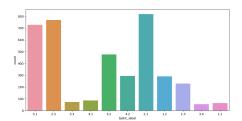


Figure 3: Sub-task C Label Distribution.

and 9 epochs. We have tested these six models across all three sub-tasks, using same performance metrics(F1-Score) to evaluate their effectiveness.

Through our testing, we have determined that our final model, which combines the MT-DNN framework with pre-training on the additional unlabeled data, achieves the best performance across all three sub-tasks. This model has demonstrated high F1 scores, indicating its ability to effectively classify and analyze the text data in our task.

### 3.1 Model Design

As illustrated in Figure 1, the training process for our model proceeded as follows:

#### (1) Domain Pre-training

As shown in table 1, we further pre-trained our models, RoBERTa-large and DeBERTa-v3-large, for 5, 7, and 9 epochs. We then evaluated the performance of these models across all three sub-tasks. Interestingly, we found that further pre-training with a longer epoch did not result in better performance than with shorter epochs.

In our experiments, we observed that the RoBERTa-large model achieved the best performance across all three sub-tasks when pre-trained for 7 epochs. As for the DeBERTa-v3-large model, we found that training for 9 epochs resulted in the best performance for sub-task A, while training for 7 epochs produced the best performance for sub-tasks B and C. This insight helped us to optimize our model training and improve its overall performance

### (2) Mutitask Learning

As shown in figure 4, we utilized the MT-DNN framework and further pre-trained models to conduct multi-task training. During training, we saved the best model for each task based on its F1-score evaluation function, resulting in three saved check-

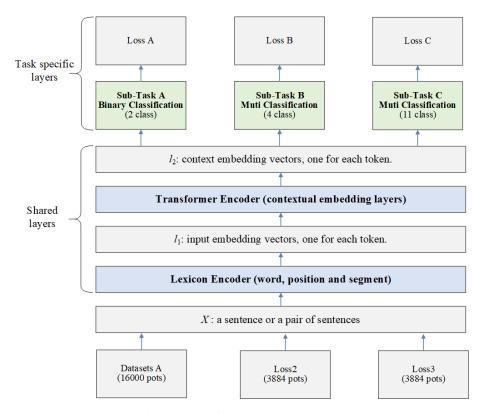


Figure 4: Multitask Learning System.

points corresponding to the best model for each of the three sub-tasks.

In the prediction phase, we used all three models to make predictions and submitted the best results for each sub-task based on the respective best model. This approach proved successful in achieving strong performance across all three sub-tasks.

### 3.2 Training Details

During the domain pre-training period, we utilized the AdamW optimization algorithm (Kingma and Ba, 2015) with 10% steps of warm-up, cosine weight decay of 1e-2, and a learning rate of 1e-5. We also set the batch size to 100 and the maximum sequence length to 120 to improve the efficiency and effectiveness of our training process.

In Mutitask Learning period, To improve the performance of our model in this multi-task learning task, we applied the AdamW optimization algorithm with 10% steps of warm-up and opened the correct\_bias item. For hyperparameters, we fine-tuned further pre-trained 24-layer RoBERTa-large and DeBERTa-v3-large models with different epochs, using a batch size of 40, dropout rate of 0.2, and cosine weight decay of 1e-2.

Additionally, we adopted a grouped layer-wise learning rate decay strategy (Ginsburg et al., 2018)

with a base learning rate of 1.2e-5, weight-ratio of 1.6, and a much higher learning rate of 5e-5 for top pooling layers. We used the stratified k-fold method to split the training data into five folds, further improving the robustness and performance of our model.

# 4 Conclusion

In this paper, we propose a complex system that combines domain pre-training and multi-task learning strategies to address the challenge of building a unified model for hierarchical sexist detection. Our approach enables us to leverage large amounts of unlabeled data to pre-train our models on domainspecific information, and then fine-tune them on the specific task of hierarchical sexist detection.

Through our experiments on the SemEval 2023 Task 10 datasets, we demonstrate that our system significantly improves model performance compared to previous approaches. Our system achieves high F1-scores across all three sub-tasks, demonstrating the effectiveness of our approach in addressing the challenges of hierarchical sexist detection.

# References

- B. Ginsburg, I. Gitman, and Y. You. 2018. Large batch training of convolutional networks with layer-wise adaptive rate scaling.
- Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers,* pages 687–697. Association for Computational Linguistics.
- Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 1923–1933. Association for Computational Linguistics.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.
- 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, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 4487–4496. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.
- Ezequiel Lopez-Lopez, Jorge Carrillo-de-Albornoz, and Laura Plaza. 2021. Combining transformer-based models with traditional machine learning approaches for sexism identification in social networks at EX-IST 2021. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural*

Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing., Málaga, Spain, September, 2021, volume 2943 of CEUR Workshop Proceedings, pages 431–441. CEUR-WS.org.

- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 2227–2237. Association for Computational Linguistics.
- Laura Plaza, Jorge Carrillo-de-Albornoz, Roser Morante, Enrique Amigó, Julio Gonzalo, Damiano Spina, and Paolo Rosso. 2023. Overview of EXIST 2023: sexism identification in social networks. In Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part III, volume 13982 of Lecture Notes in Computer Science, pages 593–599. Springer.