UniBoe's at SemEval-2023 Task 10: Model-Agnostic Strategies for the Improvement of Hate-Tuned and Generative Models in the Classification of Sexist Posts

Arianna Muti[1] and **Francesco Fernicola**[1,2] and **Alberto Barrón-Cedeño**[1]
[1]DIT – Università di Bologna, Forlì, Italy
[2]EURAC Research, Bolzano, Italy
[arianna.muti2 , francesco.fernicola2 , a.barron] @unibo.it

Abstract

We present our submission to SemEval-2023 Task 10: Explainable Detection of Online Sexism (EDOS). We address all three tasks: Task A consists of identifying whether a post is sexist. If so, Task B attempts to assign it one of four classes: threats, derogation, animosity, and prejudiced discussions. Task C aims for an even more fine-grained classification, divided among 11 classes. We experiment with fine-tuning of hate-tuned Transformer-based models and priming for generative models. In addition, we explore model-agnostic strategies, such as data augmentation techniques combined with active learning, as well as obfuscation of identity terms. Our official submissions obtain an F_1 score of 0.83 for Task A, 0.58 for Task B and 0.32 for Task C.

1 Introduction

The shared task on Explainable Detection of Online Sexism (EDOS) defines sexism as any abuse or negative sentiment directed towards women based on their gender, or based on their gender combined with one or more other identity attributes (e.g. Black women, Muslim women, Trans women) (Kirk et al., 2023).

The EDOS shared task focuses on English posts from Reddit and Gab and proposes three hierarchical sub-tasks.

Task A Binary Sexism Detection: systems have to predict whether a post is sexist or not.

Task B Category of Sexism: if a post is sexist, systems have to predict one of four mutually exclusive categories: (1) threats, (2) derogation, (3) animosity, or (4) prejudiced discussions.

Task C Fine-grained Vector of Sexism: if a post is sexist, systems have to predict one among 11 mutually exclusive subcategories, e.g., threats of harm, descriptive attacks (see Table 1, bottom).

In this paper, we present our approach to address all three subtasks. For Task A and B, we employ hate-tuned models built upon BERT (Devlin et al., 2019) and RoBERTa (Zhuang et al., 2021). We experiment with two model-agnostic techniques: obfuscation of identity terms and data augmentation with and without active learning. These strategies apply only to the model inputs and not to the internal model structure.

The first strategy involves masking gender-identifying information such as names and pronouns and aims at reducing *unintended bias* (Nozza et al., 2019; Dixon et al., 2018). The second strategy uses feedback from the model to iteratively select new training examples that positively influence the performance on the validation set.

For Task C we employ a single hate-tuned model, RoBERTa-hate, and explore the potential of generative models in such a fine-grained text classification setting via in-context learning.

2 Background

Sexist language can take many forms, ranging from overtly misogynistic or violent language to subtle forms, such as implicit bias (Sap et al., 2020) and microaggressions (Breitfeller et al., 2019). While misogyny implies hate towards women (Savigny, 2020), sexism can be concealed behind friendly statements, like in benevolent sexism (Jha and Mamidi, 2017), making sexism detection challenging.

While many relevant shared tasks have been focusing on the detection of misogyny (Fersini et al., 2018, 2020, 2022; Anzovino et al., 2018; Basile et al., 2019), some have tackled the detection of sexism as well; i.e. the two editions of sEXism Identification in Social neTworks (Rodríguez-Sánchez et al., 2021, 2022), which focused both on a binary and a multi-class categorization of sexism. In both editions, the majority of participants exploited transformer-based systems for both tasks. Some managed to improve the performance with data augmentation techniques, via back translation tech-

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1138–1147

niques (Butt et al., 2021) or task-related existing datasets (García-Baena et al., 2022).

Sexism detection has been addressed mostly as a binary or a multi-class problem by identifying the type of sexist behaviours (Parikh et al., 2019; Jha and Mamidi, 2017; Sharifirad et al., 2018). Some strategies leverage knowledge graphs (Sharifirad et al., 2018), ensemble models of neural architectures (Parikh et al., 2019), LSTMs (Jha and Mamidi, 2017) or CNNs (Zhang and Luo, 2019). Sap et al. (2020) handles this problem as a style transfer task, by turning implicit bias in language into explicit statements. Chiril et al. (2020) explore BERT contextualized word embeddings complemented with both linguistic features and generalization strategies (i.e., replacement combinations) in order to distinguish reported sexist acts from real sexist messages.

One of our objectives in this shared task is to explore the use of generative models for the identification of types of sexism. NLP has experienced a surge in promising generative models such as GPT-3 (Brown et al., 2020; Ouyang et al., 2022), GPT-Neo (Black et al., 2021; Gao et al., 2020) and BART (Lewis et al., 2019). They have shown to possess a broad set of language pattern recognition abilities, which are employed during the forward-pass to adapt to any task on the fly, including text classification. This method of classification is called prompting (Brown et al., 2020), and while many variations exist, one particularly successful technique is priming (Webson and Pavlick, 2022). Also known as in-context learning or few-shot classification, it consists in prepending a limited amount of examples to the message to be predicted, additionally wrapping each one in a template. Since both the learning and prediction steps coincide, there is no requirement for further weight updates (Webson and Pavlick, 2022). To the best of our knowledge, their usage in the field of hate speech detection is limited. Chiu et al. (2021) examined the ability of GPT-3 to identify hate speech on the ETHOS dataset (Mollas et al., 2022). Their findings show that these models are not ideal for hate speech detection, as the average accuracy rates are between 50 and 70% in a binary setting.

3 Datasets

Table 1 shows the class statistics for Tasks A, B and C. The dataset for Task A is skewed towards the negative class. As the numbers for Task B show, derogation is the most frequent type of sexism, followed by animosity; prejudiced discussions and threats are the least frequent. These four classes are further divided for Task C, which zooms into different subtypes of sexism.

		train	dev	test
	Task A			
	Sexist	3,398	486	970
	Not Sexist	10,602	1,514	3,030
	Task B			
1	Threats	310	44	89
2	Derogation	1,590	227	454
3	Animosity	1,165	167	333
4	Prejudiced Discussion	333	48	94
	Task C			
1.1	Threats of harm	56	8	16
1.2	Incitement and encouragement of harm	254	36	73
2.1	Descriptive attacks	717	102	205
2.2	Aggressive and emotive attacks	673	96	192
2.3	Dehumanising attacks and overt sexual objectification	200	29	57
3.1	Casual use of gendered slurs, profanities, and insults	637	91	182
3.2	Immutable gender differences and gender stereotypes	417	60	119
3.3	Backhanded gendered compliments	64	9	18
3.4	Condescending explanations or unwelcome advice	47	7	14
4.1	Supporting mistreatment of individual women	75	11	21
4.2	Supporting systemic discrimination against women as a group	258	37	73

Table 1: Class distribution for the tasks A, B and C.

In addition to labelled data, 2M unlabelled posts were provided —1M from Gab and 1M from Reddit—, which were used to augment our training set through active learning (Hino, 2020).

4 Models Description

We experiment with hate-tuned Transformer-based models and generative models. We compare the former with their original counterpart: BERT (Devlin et al., 2019) and RoBERTa (Zhuang et al., 2021). All hate-tuned models are fine-tuned on our downstream task. We perform a minimum parameter selection tuning on the validation set (10% of the training set). We selected the highest performing learning rate $\in$ [1e−5, 2e−5, 1e−2]; batch size $\in [4, 8, 16, 32]$; epochs in range $[1 - 10]$. Appendix A includes the best parameters for each model.

4.1 Hate-Tuned Encoder Models

We experiment with:

twitter-roberta-base-hate (Barbieri et al., 2020): a RoBERTa-base model trained on 58M tweets and fine-tuned for hate speech detection with the TweetEval benchmark (Basile et al., 2019).

hateXplain (Mathew et al., 2021): a BERT model trained on Twitter and Gab hateful posts. Each post has 3 levels of annotation: a multi-class labeling —hate, offensive or normal—, the target community and the rationales (i.e. the span of the post on which the labelling decision is based).

hateBERT (Caselli et al., 2021): a re-trained BERT model for abusive language detection, trained on RAL-E, a large-scale dataset of Reddit comments in English from communities banned for being offensive, abusive, or hateful.

4.2 Generative Models

Although the GPT-3 family currently represents the de-facto standard for generative models, it is not open source and is only accessible through its dedicated API, which not only limits the possibilities for fine-tuning, but also bills per token, making it notably expensive (Webson and Pavlick, 2022).

For this reason, we opt for GPT-Neo (Wolf et al., 2020; Black et al., 2021; Gao et al., 2020), a transformer model developed starting from EleutherAI's GPT-3 architecture replica and trained on the Pile, a large scale curated dataset for language generation. We experiment using both the 1.3B parameter model (`EleutherAI/gpt-neo-1.3B`) and the 2.7B parameter model (`EleutherAI/gpt-neo-2.7B`).

5 Experiments and Results

In this section we present the experiments performed for each of the tasks, along with the results on the development and test set.

5.1 Task A

Obfuscation of Identity Terms From the observation of previous tasks on misogyny detection (Fersini et al., 2020; Nozza et al., 2019; Muti and Barrón-Cedeño, 2020), and from a preliminary error analysis on our validation set, we noticed that identity terms might lead to biased model decisions. Identity terms tend to be associated with the positive class due to their high co-occurrence. To reduce those spurious correlations, we obfuscate all identity terms in the training set. Specifically, we replace all instances of identity terms (woman, girls, female, etc.) with a generic placeholder token; e.g., [THEM] for plural and [IT] for singular forms.[1] We then train our best-performing model — `roberta-hate` — on this obfuscated dataset and evaluate its performance on the un-obfuscated dev set. Table 2 shows the results. The performance drops by 0.005 compared to our best model, i.e. `roberta-hate`. Appendix C shows the confusion matrix for these two models. We manage to decrease the false positive rates, by limiting the spurious correlations with identity terms, at the expenses of an increase in false negatives. Given the unsatisfactory results, we discard this approach in the next steps.

Data Augmentation with External Resources Since the dataset is heavily imbalanced, we exploit the following task-related datasets annotated for misogyny or sexism to increase the size of our training set:

- **SBIC** (Sap et al., 2020) 150k social media posts with implied bias and offensiveness. The data comes from Reddit, Twitter and hate websites, such as Gab and Stormfront. We select those targeting women (3.7k posts), aiming to make our model more sensitive to implicit sexist statements.
- **AMI** (Fersini et al., 2018; Anzovino et al., 2018) 4.4k misogynous tweets of the two editions of Automatic Misogyny Identification targeting the English language.
- **The 'Call me sexist but' Dataset** (Samory, 2021) 2.1k sexist tweets collected by querying the phrase 'call me sexist but', which were subsequently removed, leaving only the remaining text. This dataset contains in addition 1.1k hostile sexist instances from Waseem and Hovy (2016) and 821 instances of benevolent sexism from Jha and Mamidi (2017).
- **Microaggressions** (Breitfeller et al., 2019) 1.3k gender-based posts from *microaggressions.com*, which collects self-reported microaggression episodes.
- **Incels.is** 1.1k posts that we bootstrapped from the *Incels.is* forum, annotated for misogyny.
- **Implicit Hate** (ElSherief et al., 2021) 6.4k implicitly hateful tweets, annotated for the target (e.g., race, religion, gender). We select the 65 posts targeting women.

[1] For example, `I hate women` would be transformed into `I hate [THEM]`.

model	strategy	dev	test
roBERTa-base	–	0.813	–
bert-base-unc.	–	0.781	–
HateXplain	–	0.791	–
HateBert	–	0.839	–
roberta-hate	–	**0.845**	0.835
roberta-hate	obfuscation	0.840	–
roberta-hate	data aug. (ext.)	0.820	–
roberta-hate	data aug. (int.)	0.830	–

Table 2: Macro F_1 for Task A. Strategy *data aug. (ext.)* refers to adding the positive instances from external datasets from similar tasks; *data aug. (int.)* refers to silver data produced via active learning. Our submitted system (roberta-hate) differs from the top-performing one by 0.04 points.

We add in bulk only the positive instances to make the dataset more balanced. Table 2 reports the results of this experiment, showing a drop of 0.02 points compared to our best model. Appendix B includes an ablation study in which we add one dataset at a time, including both positive and negative instances. Whether we add additional positive instances in bulk, or one dataset at a time (see Appendix B), we see no improvements in the performance with respect to our best model, i.e. roberta-hate. The model does not seem to benefit from external data, probably because of the cross-domain shift (Twitter, Reddit, blogs), which confuses the model. This finding is in line with current research, claiming that hate speech detection models show low generalizability across datasets (Yin and Zubiaga, 2021).

Data Augmentation with Provided Resources Since the previous technique did not yield positive results, we employ data augmentation using the unlabelled data provided. We use the following approach. Let D_l be our labelled training set, D_u the unlabelled dataset, and m_r our best baseline (`roberta-hate`): (*i*) Train m_r on D_l. (*ii*) Predict the instances in D_u with m_r. (*iii*) Rank the instances in D_u according to the confidence of the prediction score returned by m_r. (*iv*) Add iteratively the top-k instances in D_u as silver data to D_l. We repeat until the performance on the validation set improves and re-train a new model on our newly-originated training set at the end. We set $k = 200$ and we manage to add $1k$ instances to our original dataset (after four iterations the performance has stopped improving).

model	strategy	dev	test
roBERTa-base	–	0.614	–
bert-base-uncased	–	0.570	–
roberta-hate	–	**0.638**	0.58
HateXplain	–	0.578	–
HateBert	–	0.606	–
Bart	zero-shot	0.280	–

Table 3: Macro F_1 for Task B. The submitted system (roberta-hate) differs from the top-performing system by 0.15 points.

As Table 2 shows, our model does not benefit from additional data, neither task-related, nor labelled via active learning. Hence, we do not consider such strategies for the rest of the tasks.

5.2 Task B

For Task B we experiment with the same models employed for Task A. In addition, we experiment with *bart-large-mnli* (Lewis et al., 2019), a generative model with a ready-made zero-shot sequence classifier head. Table 3 shows the results. As expected, the zero-shot model shows the worst performance. The top-performing model on the dev set is confirmed to be roberta-hate, therefore we use it to predict on the final test set. However, in the test set the performance drops by 0.06 points.

5.3 Task C

Starting from the best model for Task B, `roberta-hate`, we employ it for Task C as well. We develop two training strategies, once from scratch and once in a cascaded setting, following the broader category assigned by the Task B model. In the first approach (*all_categories*), the model has access to all eleven categories, whereas in the second approach (*subcategories*), we use four classifiers, one for each class predicted by the model for Task B. Table 4 shows the results. The model does not benefit from the pre-categorization of Task B, due to the noisy input, resulting in a 0.02 performance drop. To confirm that the errors are propagated by the imperfect Task B model, we perform an additional set of experiments on the test set with a perfect classifier for Task B instead of relying on the performance of a previous model. Using the same settings, the performance significantly increases by 0.25, thus confirming that the noisy input was swaying the model and that using separate classifiers for each subclass leads to increased accuracy for the predictions.

model	strategy	dev	test
GPT-Neo1.3	all categories	0.048	0.093*
GPT-Neo1.3	subcategories	0.120	-
GPT-Neo2.7	all categories	0.025	-
GPT-Neo2.7	subcategories	0.120	-
roberta-hate	all categories	0.332	0.315
roberta-hate	subcategories	0.316	-
GPT-Neo1.3	subcategories+	-	0.180*
roberta-hate	subcategories+	-	0.580*

Table 4: Macro F_1 for Task C. The submitted system (roberta-hate) differs from the top-performing system by 0.24 points. Rows marked with (*) are calculated after the release of the gold labels for the test dataset. Rows marked with (+) start from a perfect Task B classifier, which is in line with the top-performing system.

Given the low scores obtained and the significant amount of subcategories for Task C, we attempt to approach the task using generative models to leverage their high ability of contextual understanding. We employ priming techniques to generate the predictions, devising prompts with examples and labels extracted from the main dataset. Additionally, we attempt to explore the effect of different prompting scenarios on the performance of generative models, following the same settings previously used with `roberta-hate`. We also experiment with including either one or two examples per category for the *subcategories* setting to understand whether providing more in-context data improves model performance. Appendix D shows the prompt structure.

The temperature value for the GPT-Neo models is set empirically. In our case, being restricted among 11 classes meant the model should not be too creative, but setting it too low might confuse it. A range of $[0.5, 1]$ is commonly adopted for generative tasks, but we need to adapt it for classification tasks, which are more restrictive. We have first experimented with a value of 0.1 on a small set of prompts (Brown et al., 2020; Webson and Pavlick, 2022), which did not produce meaningful predictions. By increasing it to 0.3 we managed to obtain sensible outputs, we thus set the temperature to 0.3 for all of our experiments. Table 4 summarizes the results obtained using the different prompt settings. The only results we show for the *subcategories* setting are obtained using the prompt containing two examples, because it emerged that by providing a single example the model would often generate additional random categories, such as *'1.3'*, despite including the complete list of categories within the prompt itself. While the results on both *subcategories* settings might look promising, the results are actually misleading because the output prediction is always the last category provided in the examples, mimicking a repetitive pattern rather than actually generating a meaningful prediction. Only the *all_categories* prompt appears to be able to generate actual predictions and we experiment with it on the test set as well. The results are on par with those from random predictions, suggesting that such a fine-grained classification is difficult to predict using in-context learning with our generative models, in spite of their strong NLU capabilities.

6 Error Analysis

We conduct an error analysis for Task A to understand patterns of misclassification.[2] We observe all misclassified instances manually. With the help of the NLTK library, we retrieve frequent words in misclassified instances. In false positive cases in Task A we found identity terms (e.g., women) and sexually-abusive terms such as *rape* and *f%ck*, used without the intentionality of harming, like in the case of reports of sexist acts or non-offensive slurs. Since such terms are frequently used in sexist instances, the model gets confused when they occur in non-sexist instances, resulting in being labeled as sexist. This suggests that intentionality must be considered when discriminating between actually sexist and reported sexist acts, as stressed by Chiril et al. (2020). The obfuscation of identity terms led to a decrease in false positive rates, at the expenses of a lower recall, resulting in more false negatives. This is expected, as for instance the sentence 'I hate women' would be likely identified by a standard model, but the sentence 'I hate [THEM]' would likely not be classified as sexist.

7 Conclusions

In this paper we presented our submission to the EDOS shared task (Kirk et al., 2023). We experimented with hate-tuned Transformer-based models for Task A, B and C, and generative models for Task B and C. For Task A, we adopted model-agnostic strategies such as the obfuscation of identity terms and data augmentation, with and without active learning. For all three tasks, our best model

[2]We do not include Task B and Task C due to the high variability of our results.

is always a vanilla `roberta-hate`. For Task A, the model does not benefit from additional data, neither task-related annotated data, nor silver data produced via active learning. The obfuscation of identity terms does not negatively impact the performance, but does not help either, although we manage to decrease the false positive rates, at the expense of a lower recall. For Task C, deep learning models do not have enough samples to learn from: a hybrid linguistically-informed system might thus be preferable for this kind of task and it is our intention to try it in the future.

References

Maria Anzovino, Elisabetta Fersini, and Paolo Rosso. 2018. Automatic identification and classification of misogynistic language on twitter. In *International Conference on Applications of Natural Language to Information Systems*, pages 57–64. Springer.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Luke Breitfeller, Emily Ahn, David Jurgens, and Yulia Tsvetkov. 2019. Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts. 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 1664–1674, Hong Kong, China. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Sabur Butt, Noman Ashraf, Grigori Sidorov, and Alexander Gelbukh. 2021. Sexism identification using bert and data augmentation - exist2021. *CEUR Workshop Proceedings*, 2943:381–389. Publisher Copyright: © 2021 CEUR-WS. All rights reserved.; null ; Conference date: 21-09-2021.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Patricia Chiril, Véronique Moriceau, Farah Benamara, Alda Mari, Gloria Origgi, and Marlène Coulomb-Gully. 2020. He said "who's gonna take care of your children when you are at ACL?": Reported sexist acts are not sexist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4055–4066, Online. Association for Computational Linguistics.

Ke-Li Chiu, Annie Collins, and Rohan Alexander. 2021. Detecting hate speech with gpt-3. *arXiv:2103.12407*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 67–73, New York, NY, USA. Association for Computing Machinery.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2018. Overview of the Evalita 2018 task on automatic misogyny identification (AMI). In *EVALITA Evaluation of NLP and Speech Tools for Italian: Proceedings of the Final Workshop 12-13 December 2018, Naples*, pages 59–66. Torino: Accademia University Press.

Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2020. AMI@EVALITA2020: Automatic misogyny identification. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*. CEUR.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Daniel García-Baena, Miguel Ángel García Cumbreras, Salud María Jiménez Zafra, and Manuel García-Vega. 2022. Sinai at exist 2022: Exploring data augmentation and machine translation for sexism identification. In *IberLEF@SEPLN*.

Hideitsu Hino. 2020. Active learning: Problem settings and recent developments. *CoRR*, abs/2012.04225.

Akshita Jha and Radhika Mamidi. 2017. When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 7–16, Vancouver, Canada. 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*, Toronto, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2022. ETHOS: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*, 8(6):4663–4678.

Arianna Muti and Alberto Barrón-Cedeño. 2020. UniBO@AMI: A Multi-Class Approach to Misogyny and Aggressiveness Identification on Twitter Posts Using AlBERTo. In (Fersini et al., 2020).

Debora Nozza, Claudia Volpetti, and Elisabetta Fersini. 2019. Unintended bias in misogyny detection. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 149–155.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

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, Laura Plaza, Julio Gonzalo, Paolo Rosso, Miriam Comet, and Trinidad Donoso. 2021. Overview of exist 2021: sexism identification in social networks. *Procesamiento del Lenguaje Natural*, 67(0):195–207.

Francisco Rodríguez-Sánchez, Jorge Carrillo de Albornoz, Laura Plaza, Adrián Mendieta-Aragón, Guillermo Marco-Remón, Maryna Makeienko, María Plaza, Julio Gonzalo, Damiano Spina, and Paolo Rosso. 2022. Overview of exist 2022: sexism identification in social networks. *Procesamiento del Lenguaje Natural*, 69(0):229–240.

Mattia Samory. 2021. The 'call me sexist but' dataset (cmsb). . Data File Version 1.0.0, https://doi.org/10.7802/2251.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.

Heather Savigny. 2020. *Sexism and Misogyny*. John Wiley & Sons, Ltd.

Sima Sharifirad, Borna Jafarpour, and Stan Matwin. 2018. Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs. In

Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pages 107–114, Brussels, Belgium. Association for Computational Linguistics.

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, San Diego, California. Association for Computational Linguistics.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenjie Yin and Arkaitz Zubiaga. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, 7.

Ziqi Zhang and Lei Luo. 2019. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, 10(5):925–945.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Best Parameters per Model and Task

Table 5 shows the hyperparameters that led to the best performance for each model on each task.

model	task	lr	bs	epoch
roBERTa-base	A	1e-5	16	4
roBERTa-base	B	1e-5	8	5
bert-base-uncased	A, B	2e-5	16	4
roberta-hate	A, B	1e-5	16	5
roberta-hate	C	1e-5	16	6
HateXplain	A	2e-5	16	8
HateXplain	B	2e-5	16	5
HateBert	A, B	2e-5	16	5

Table 5: Best hyperparameters per model and task, as fine-tuned on the development set (lr=learning rate, bs=batch size).

B Ablation Study

This Appendix contains the results of the ablation study performed to observe the impact of adding each dataset individually to the original training set for task A. All models were trained with roberta-hate, with the original training set plus the instances of each dataset. The hyperparameters are as in Table 5. As Table 6 shows, the datasets that less impacted the performance were Microaggressions (Breitfeller et al., 2019) and Implicit Hate (ElSherief et al., 2021), which had the smallest number of instances.

To better understand the impact of each dataset individually, we train a model with the original training set plus the same number of instances across all external datasets. We select the dataset with the least number of instances (Incels.is - 1.1k) and we select 1.1k instances from all datasets to be added to the original training set.[3] Column dev_sampled in Table 6 shows the results. The outcome changes only when adding data from SBIC and AMI, observing an improvement of 0.2 points in both cases. Limiting the number of external instances leads to an improvement in the performance, confirming that more data is not always the better in this cross-domain setting. Moreover, we perform another experiment with the aim of selecting only potentially good instances for the model to learn from, among the external datasets. We train on the official training set, we predict on all external datasets, we select only the instances predicted correctly by such model and we train another model by adding all those instances in bulk. The outcome remains unaltered so we do not report the results for this experiment.

C Confusion Matrix

Here we show the confusion matrix of roberta-hate and roberta-hate+obfuscation. With the obfuscation of identity terms, we manage to decrease the number of false positives, at the expense of more false negatives. As a result, even if this method helps to reduce spurious correlations with identity terms, we cannot neglect the significant drop in recall.

[3]We exclude Implicit Hate because it has only 65 instances.

dataset	dev	dev_sampled
SBIC	0.78	0.80
AMI	0.80	0.82
Call me sexist but	0.81	0.81
Microaggressions	0.82	0.82
Incels.is	0.81	0.81
Implicit Hate	0.82	–

Table 6: Macro F_1 score on the development test for Task A showing the impact of each dataset in the data augmentation process.

RoBERTa-hate

	Positive	Negative
Positive	1412	102
Negative	123	363

R-h + Obfuscation

	Positive	Negative
Positive	1421	93
Negative	135	351

D Prompts for Task C

This appendix contains the 3 prompts used for the Task C experiments: the *all_categories* prompt (Figure 1), the *subcategories* prompt with two examples (Figure 2) and the *subcategories* prompt with one example (Figure 3).

```
Predict the category for the last message
based on the category types in the examples.
Choose one among the following categories:
4.1 supporting mistreatment of individual women
2.3 dehumanising attacks & overt
sexual objectification
2.2 aggressive and emotive attacks
1.2 incitement and encouragement of harm
4.2 supporting systemic discrimination against
women as a group
1.1 threats of harm
3.1 casual use of gendered slurs, profanities,
and insults
3.3 backhanded gendered compliments
3.4 condescending explanations or
unwelcome advice
2.1 descriptive attacks
3.2 immutable gender differences and
gender stereotypes

Examples:
###
Message: <message_1>
Category: <category_1>
###
Message: <message_2>
Category: <category_2>
###
...
###
Message: <message_to_predict>
Category:
```

Figure 1: Example prompt for the *all_categories* setting using all available categories.

```
Predict the category for the last message
based on the category types in the examples.
Choose one among the following categories:

1.1 threats of harm
1.2 incitement and encouragement of harm

Examples:
###
Message: <message_1>
Category: <category_1>
###
Message: <message_2>
Category: <category_1>
###
Message: <message_3>
Category: <category_2>
###
Message: <message_4>
Category: <category_2>
###
Message: <message_to_predict>
Category:
```

Figure 2: Example prompt for the *subcategories* setting using two examples per category.

```
Predict the category for the last message
based on the category types in the examples.
Choose one among the following categories:

1.1 threats of harm
1.2 incitement and encouragement of harm

Examples:
###
Message: <message_1>
Category: <category_1>
###
Message: <message_2>
Category: <category_2>
###
Message: <message_to_predict>
Category:
```

Figure 3: Example prompt for the *subcategories* setting using one example per category.