QCon at SemEval-2023 Task 10: Data Augmentation and Model Ensembling for Detection of Online Sexism

Weston Feely*, **Prabhakar Gupta***, **Manas Mohanty, Timothy Chon, Tuhin Kundu, Vijit Singh, Sandeep Atluri, Tanya Roosta, Viviane Ghaderi, Peter Schulam, Heba Elfardy**
Amazon
{feelyw, prabhgup, mhmanas, timochon, tkundu, vijit, satluri, troosta, vsg, schulamp, helfardy}
@amazon.com

Abstract

The web contains an abundance of user-generated content. While this content is useful for many applications, it poses many challenges due to the presence of offensive, biased, and overall toxic language. In this work, we present a system that identifies and classifies sexist content at different levels of granularity. Using transformer-based models, we explore the value of data augmentation, use of ensemble methods, and leverage in-context learning using foundation models to tackle the task. We evaluate the different components of our system both quantitatively and qualitatively. Our best systems achieve an F_1 score of 0.84 for the binary classification task – aiming to identify whether a given content is sexist or not – and 0.64 and 0.47 for the two multi-class tasks that aim to identify the coarse and fine-grained types of sexism present in the given content respectively.

1 Introduction

The web provides tremendous value, in part, because of how easy it is to publish new content and for others to access it. This accessibility and openness, however, comes at a cost. Today's web with its plethora of user-generated content is rife with toxic, offensive, and obscene language. Therefore, it has become essential to moderate this content at scale. Removing the toxic content is even more critical for technologies that automatically index and surface web-content to end users, as well as generative language models that utilize this content. To address the need for content moderation, the natural language processing (NLP) and machine learning communities have put considerable effort into the problem of automatically detecting toxic language.

In this paper, we describe the system behind our team's submission to the SemEval-2023 "Explainable Detection of Online Sexism" shared task (Kirk et al., 2023). There are three tasks – Task A, B and C. Task A is a binary classification task and tasks B and C are multi-class classification tasks with 4 and 11 classes respectively.

The current dominant approach to detecting toxic language is to fine-tune sequence to sequence transformer (Vaswani et al., 2017) models (Hanu and Unitary team, 2020; Mutanga et al., 2020; Kim et al., 2022; Markov et al., 2022). Our system builds on this thread of research, and extends it by applying methods to ensemble deep learning models (Wenzel et al., 2020; Gontijo-Lopes et al., 2022; Wortsman et al., 2022). We ensemble transformers, where each member of the ensemble uses a different base architecture (i.e. BERT, RoBERTa, etc.), different training data, and different training hyperparameter configurations. In addition, we explore whether using in-context learning and weak-labels from foundation models, such as GPT-3 (Brown et al., 2020), further improve the performance.

Across Tasks A, B, and C, we find that our ensembles of fine-tuned transformers consistently reach higher macro F1 score on the test set than the best individual model that we would have chosen using macro F1 score on the development set. Consistent with the preliminary experiments in (Wortsman et al., 2022), however, the improvements we observe are relatively small. This encouraging result suggests that ensembling fine-tuned transformers has promise, but there is need for additional research into techniques for selecting members of the ensemble, and for computing the aggregated output. For in-context learning using GPT-3 (Brown et al., 2020), we find that the performance is competitive to the fine-tuned models, but it provides mixed results when used as a member of an ensemble.

2 Related Work

Work on automatically detecting abusive language on the web goes back to 2009, when Yin et al. (2009) proposed several features for detecting abu-

*Equal contribution.

sive user-generated content in "Web 2.0" applications. Yin et al. (2009) proposed two particularly interesting features: template patterns that capture common abusive phrases, and contextual features that capture similarities between the language being classified and "nearby" language (e.g. the text from a parent post on a message board). Template patterns of abusive phrases are no longer used as explicit features for abusive language classifiers, but have been used to create model "unit tests" to expose weaknesses of leading systems (Röttger et al., 2021). The issue of whether context is important for determining if a piece of text is offensive is still debated (Pavlopoulos et al., 2020).

Djuric et al. (2015) was the first to use word vector representations in an abusive language classifier, as opposed to discrete representations of words as in a bag of words approach. They used the Le and Mikolov (2014) model, but observed marginal improvements over a simpler bag of words approach (0.8 AUC from 0.79). Nobata et al. (2016) built on Djuric et al. (2015) and further experimented with word and paragraph embeddings combined with several "classical" features, such as character and token n-grams, features derived from dependency parses, and a suite of orthographic features. On the dataset used in Djuric et al. (2015), the approach in Nobata et al. (2016) further improved the AUC to 0.91. Surprisingly, however, the character n-grams alone achieved AUC of 0.90, suggesting that the word and paragraph embeddings offer only marginal improvements to the problem.

Early word embedding approaches (Djuric et al., 2015; Nobata et al., 2016) learned task specific word vectors on relatively small corpora, on the order of hundreds of thousands of texts. This may explain the underwhelming performance observed in these studies. Large "foundation models" have improved this approach. Nikolov and Radivchev (2019) was one of the first to show that fine-tuning BERT for hate speech detection can significantly improve over standard baselines like logistic regression and support vector classifiers. The Nikolov and Radivchev (2019) work showed that BERT improves performance the most for more fine-grained classifications of hate speech, meaning going beyond the basic problem of binary classification of the content. There have since been several proposed extensions of the standard transformer fine-tuning process. For instance, Tran et al. (2020) shows that learning task-specific vocabularies for transformers can marginally improve performance.

Although fine-tuned transformers have successfully "cracked" many text classification tasks, the problem of hate speech detection remains difficult. Progress on the problem is tricky because the judgement of what is offensive is inherently subjective. This has led to an active thread of ongoing research focused on understanding the roles that data collection, annotation guidelines, and annotator biases play on the quality of hate speech classifiers (see, e.g., (Dixon et al., 2018; Fortuna et al., 2020; Röttger et al., 2022; Garg et al., 2022).

Our system builds on ideas from the emerging literature on methods for creating ensembles of deep, "high-capacity" models (Wenzel et al., 2020; Gontijo-Lopes et al., 2022; Wortsman et al., 2022). The majority of experimental work in the space of model ensemble has been in the area of computer vision. Our paper contributes additional data to the open question of whether ensembling of deep models can successfully solve problems in the area of natural language.

3 Data Selection

For each classification task, our approach involves ensembling different fine-tunings of pre-trained Transformer (Vaswani et al., 2017) based models. These models vary in the selection of: base models, hyper-parameter configurations, and the data used to fine-tune each model. Moreover, they make use of weak-labels generated by prompting GPT-3 (Brown et al., 2020) using in-context learning. In this section we go into the details of the data we used for model training.

For all model trainings, we include the corpus of gold-labeled training data provided for the shared task (Kirk et al., 2023). For tasks A, B, and C, we randomly sample 10% of the gold data, and use it as a validation set to perform hyper-parameter tuning. We choose the best model based on the performance on the corresponding task's official development set. In addition to the gold data, we leverage a subset of the unlabeled Reddit and Gab data released by the task organizers. We apply silver labeling to this data and augment our original training data with it. To further augment the training dataset, we use two existing public datasets: "EXIST" (Rodríguez-Sanchez et al., 2021) and "Call Me Sexist" (Samory, 2021). Tables 1, 2, and 3 show the statistics of the gold, silver, and public datasets utilized for Tasks A, B and C respectively.

	Train					Dev.	Test
	Gold	Silver (Reddit)	Silver (Gab)	Call Me Sexist	EXIST	Gold	Gold
Non-Sexist	10,602	3,467	3,476	3,398	11,822	1,514	3,030
Sexist	3,398	3,533	3,524	486	1,809	486	970

Table 1: Distribution of gold, silver and public datasets used for Task A.

	Train		Dev.	Test
	Gold	Silver (Reddit)	Gold	Gold
1. Threats	310	1,000	44	89
2. Derogation	1,590	1,500	227	454
3. Animosity	1,165	2,000	167	333
4. Prejudice	333	1,000	48	94

Table 2: Distribution of gold and silver datasets used for Task B.

3.1 Public Datasets

For Task A, we include training data samples from the publicly available datasets – (1) "Call Me Sexist" (Samory, 2021) (2) "EXIST" (Rodríguez-Sanchez et al., 2021). We randomly sample 5,000 samples from Call Me Sexist dataset to include in models (G, I, J) and include the entire EXIST corpus for models (L, M) in our Task A results Table 4.

3.2 Silver Labeling

To label the provided unlabeled data, we use a weighted sum (weights $= (0.75, 0.25)$) of normalized scores from two weak classifiers – (1) RoBERTa and (2) Sentence Transformer, where a higher score indicates a higher likelihood of samples belonging to the given class. We select a fixed number of highest scored samples for each class according to the task to generate additional silver-labelled training dataset. Additionally, we create an offensive term list for Task A to filter out some samples from unlabelled corpus to get better silver-labelled data.

For Task A, we use both the unlabeled Reddit and Gab datasets, whereas for Tasks B and C we only use samples from the unlabeled Reddit dataset; we experimented with Gab dataset for both tasks but observed lower performance when using it. For Task A, we add 5,000 positive and 5,000 negative samples each from both Reddit and Gab datasets, increasing the training data size by 20,000 samples. For Task C, we added 500 samples for each of the 11 classes only from the Reddit dataset, increasing the training data size by 5,500 samples.

RoBERTa We fine-tune a RoBERTa (large) (Liu et al., 2019) model using only the gold training data provided for the task A and C.

Sentence Transformer We use a sentence transformer model (Reimers and Gurevych, 2020) that maps sentences to a 384-dimensional dense vector space, to generate a similarity score between each labeled training sample and the unlabeled sample. Using these similarity scores, for each unlabeled sample we generate an average class score for all classes for a given task. Based on the task, we select the highest and lowest scored samples to add to the training data.

Offensive Term List For Task A, we utilize a manually-created list of offensive terms. We collect counts of how often terms from this manually-created list occur in both the sexist and non-sexist samples in the Task A gold training data, and take the top terms when ranked by their ratio of sexist to non-sexist mentions. Additionally, only terms that appeared more than five times were included for robustness. For gendered slurs, both singular and plural variants of the same nouns were included regardless of the frequency in the gold training data. This offensive term-list was used to select unlabeled training data samples from Reddit and Gab to include as silver labeled training data for Task A. The 5,000 silver labeled samples we include when training Model L in our results Table 4 includes only samples which contained at least one term from this term list.

4 Models

As mentioned earlier, we use an ensemble of Transformer-based models (Vaswani et al., 2017) for the three tasks. To this end, we fine-tuned: BERT (base-uncased) (Devlin et al., 2019), RoBERTa (base and large) (Liu et al., 2019), XLM-RoBERTa (large) (Conneau et al., 2020), ELECTRA (base) (Clark et al., 2020), BERTweet (large) (Nguyen et al., 2020), MiniLM (L12-H384) (Wang et al., 2020), ALBERT (Lan et al., 2020).

Additionally, we utilized in-context learning to prompt GPT-3 (Brown et al., 2020) to generate predictions for the development and test set examples for Task A. We follow a similar approach to (Chiu

	Train		Dev.	Test
	Gold	Silver (Reddit)	Gold	Gold
1.1 Threats of harm	56	500	8	16
1.2 Incitement and encouragement of harm	254	500	36	73
2.1 Descriptive attacks	717	500	102	205
2.2 Aggressive and emotive attacks	673	500	96	192
2.3 Dehumanising attacks and overt sexual objectification	200	500	29	57
3.1 Casual use of gendered slurs, profanities and insults	637	500	91	182
3.2 Immutable gender differences and gender stereotypes	417	500	60	119
3.3 Backhanded gendered compliments	64	500	9	18
3.4 Condescending explanations or unwelcome advice	47	500	7	14
4.1 Supporting mistreatment of individual women	75	500	11	21
4.2 Supporting systemic discrimination against women as a group	258	500	37	73

Table 3: Dataset distribution for Task C.

and Alexander, 2021).[1] The prompt format we used was the word "SENTENCE" followed by a randomly selected training set example, then the word "LABEL" followed by either the word "non-sexist" or "sexist", to indicate the label of the preceding example. We included K randomly sampled positive training set examples and K randomly sampled negative training examples for each prompt.[2] At the end of the prompt, we included a development or test set example followed by "LABEL" with no following text, prompting for an example completion with either the word "non-sexist" or "sexist" from the model. We used these prompt completions as predicted labels, which are included in our results Tables 4 and 5.

4.1 Fine-tuning

In our approach, we fine-tune models for two ensembles: one ensemble of models for Task A and another ensemble for Task C. For Task B, the fine-grained labels from Task C to the coarse-grained labels of Task B. For example, a *"1.1 Threats of harm"* Task C label would map to the *"1. Threats, plans to harm and incitement"* Task B label.

For Task A we use a weighted binary cross entropy (BCE) loss function. The loss is equally weighted between the positive and the negative samples present in the training data for each fine-tuned model.

For multi-class tasks, B and C, we take inspiration from related work on hierarchical modeling tasks using a shared loss function (Wu et al., 2017), and treat both tasks as a single model with two objectives. We capture the hierarchical structure of the coarse and fine-grained labels for Task A and B in this single model's loss function. We create a composite loss computed of the sum of (1) the standard cross entropy (CE) loss on all 11 Task C outputs, and (2) the cross entropy of the Task C outputs projected onto the 4 outputs for Task B, as shown in equation 1, where $Y_{B|C}$ is the true class and $\hat{Y}_{B|C}$ is the predicted class. For example, we compute the model's prediction for *"1. Threats, plans to harm and incitement"* as the sum of Pr(*1.1 Threats of harm*) and Pr(*1.2 Incitement and encouragement of harm*). The key intuition is that the model's output should be consistent at both the coarse and the fine-grained levels, and that enforcing this consistency will improve the model's performance in Task C. This loss function introduces an additional hyper-parameter β that weights the Task C loss against the Task B loss in the composite calculation.[3]

$$\mathcal{L}_C = \beta \cdot CE(Y_B, \hat{Y}_B) + CE(Y_C, \hat{Y}_C) \quad (1)$$

4.2 Ensembling

As shown in (Wenzel et al., 2020), ensembling different models with different hyperparameter settings outperforms using a single model. Therefore, we explore the performance of different ensembles of models by measuring the performance of all model combinations in the power set $\mathcal{P}(S)$, for the set of models S for each of the tasks. For each task, the predictions for a given ensemble of models is based on the majority vote for each sample. Each ensemble, as well as the individual models, were evaluated on the development set for each task, and the best performing setup was used to label the held out test sets.

[1] The exact model version we used was "text-davinci-003".

[2] We experimented with K = 5, 10 and 20 and achieved the best results using 5 examples.

[3] We determined β empirically based on the performance on the development set. Specific values can be found in Appendix A.

	Model	F_1 score		
		Non-sexist	Sexist	Macro
	Majority Baseline	0.86	0.0	0.43
(A)	BERT (L)	0.90	0.66	0.78
(B)	ELECTRA (S)	0.91	0.69	0.80
(C)	RoBERTa (B)	0.92	0.74	0.83
(D)	RoBERTa (L)	0.92	0.70	0.81
(E)	RoBERTa (L)	0.93	0.76	0.84
(F)	RoBERTa (L)	0.92	0.74	0.83
(G)	RoBERTa (L)	0.91	0.72	0.82
(H)	RoBERTa (L)	0.93	0.75	0.84
(I)	RoBERTa (L)	0.92	0.76	0.84
(J)	RoBERTa (L)	0.92	0.73	0.82
(K)	RoBERTa (L)	0.92	0.75	0.83
(L)	RoBERTa (L)	0.93	0.76	0.85
(M)	RoBERTa (L)	0.92	0.72	0.82
(N)	XLM-R (L)	0.92	0.72	0.82
(O)	GPT3 (Prompting)	0.85	0.61	0.73
	ALL (A-O)	0.93	0.77	0.85
	(E)+(F)+(H)+(I)+(L)+(O)	**0.94**	**0.80**	**0.87**

Table 4: **Development set** results for Task A. The final row is for the best performing ensemble on the development set of Task A.

	Model	F_1 score		
		Non-Sexist	Sexist	Macro
	Majority Baseline	0.86	0.0	0.43
(A)	BERT (L)	0.90	0.75	0.77
(B)	ELECTRA (S)	0.91	0.71	0.81
(C)	RoBERTa (B)	0.92	0.75	0.84
(D)	RoBERTa (L)	0.92	0.71	0.81
(E)	RoBERTa (L)	0.92	0.76	0.84
(F)	RoBERTa (L)	0.92	0.75	0.84
(G)	RoBERTa (L)	0.91	0.73	0.82
(H)	RoBERTa (L)	**0.93**	0.76	0.84
(I)	RoBERTa (L)	0.91	0.74	0.82
(J)	RoBERTa (L)	0.91	0.72	0.82
(K)	RoBERTa (L)	0.91	0.74	0.83
(L)	RoBERTa (L)	0.92	0.75	0.83
(M)	RoBERTa (L)	0.91	0.70	0.80
(N)	XLM-R (L)	0.91	0.72	0.81
(O)	GPT3 (Prompting)	0.85	0.61	0.73
	ALL (A-O)	**0.93**	**0.77**	**0.85**
	(E)+(F)+(H)+(I)+(L)+(O)	0.92	**0.77**	0.84

Table 5: **Test set** results for Task A. The final row is the test-set score for the ensemble that performed best on the development set of Task A.

5 Experiments

As mentioned earlier, we explore fine-tuning individual models for each task. Tables 4, 6, and 8 report the performance of each individual model, as well as the best ensemble for each of Tasks A, B, and C respectively on the development sets. Tables 5, 7, and 9 report the performance of the best (i.e. submitted) ensemble on the held-out test sets of the three tasks. We report the macro F_1 average of all classes for each task – since it is the official metric for all tasks – as well as the per-class F_1 score. As a baseline for comparison with our individual models, we include a row with results from always predicting the most frequent class in the training data for a given task – "Majority Baseline" and as a baseline for comparison with our best ensembles, we include results for a simple ensemble of all models trained for each task "ALL".

Task A: Binary Sexism Detection For Task A, the best performing ensemble includes five fine-tunings of RoBERTa (Large), and predictions obtained from prompting GPT-3, as described in Section 4.

- Models E and F were trained on the gold training data set.
- Model H was trained on both the gold training data set as well as 5,000 silver labeled Reddit and Gab samples.
- Model I was trained on the gold training data set, 5,000 silver labeled Reddit and Gab samples (which here were filtered to include only samples with terms from our curated term-list), and 5,000 samples from the "Call Me Sexist" corpus.
- Model L was trained on the gold training data set and the entire EXIST corpus.
- Model O are predictions from prompting GPT-3.

Tasks B and C For both Task B and Task C we train one set of models A-H on the Task C labels using the loss described in Section 4.1, with the predicted labels for Task C mapped to their corresponding labels for Task B. All the models are trained on gold training data set and silver labeled Reddit samples.

For Task B the best performing ensemble includes models A, B, C, D, E which are fine-tunings of BERT, ELECTRA, RoBERTa (Base), RoBERTa (Large) and, RoBERTa (Large), respectively. For Task C the best performing ensemble includes models B, D, E, F which are fine-tunings of ELECTRA, RoBERTa (Large), RoBERTa (Large), and, XLM-RoBERTa respectively. We also include scores from GPT3 (Prompting) as Model I in Table 6. See Appendix A.

6 Discussion

Task A Test set results for Task A show that our submission of the best-performing ensemble on the development set, which has an F_1 score of 0.84, is slightly outperformed by an ensemble of all mod-

		F_1 score				
Base Model		**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Macro**
Majority Baseline		0.0	0.64	0.0	0.0	0.16
(A)	BERT (B)	0.66	0.71	0.58	0.56	0.63
(B)	ELECTRA (L)	0.70	0.70	0.56	0.52	0.62
(C)	RoBERTa (B)	0.61	0.66	0.55	0.64	0.62
(D)	RoBERTa (L)	**0.75**	0.73	0.62	0.70	0.70
(E)	RoBERTa (L)	0.69	0.71	0.60	0.63	0.66
(F)	XLM-R (L)	0.62	0.69	0.57	0.55	0.61
(G)	BERTweet (L)	0.70	0.73	0.56	0.60	0.65
(H)	MiniLM (L12)	0.54	0.57	0.58	0.58	0.57
(I)	GPT3 (Prompting)	0.47	0.58	0.43	0.00	0.37
ALL (A-I)		0.63	0.73	0.64	0.64	0.66
(A)+(B)+(C)+(D)+(E)		0.72	**0.75**	**0.69**	**0.72**	**0.72**

Table 6: **Development set** results for Task B. The final row is for the best performing ensemble on the development set of Task B.

		F_1 score				
Base Model		**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Macro**
Majority Baseline		0.0	0.64	0.0	0.0	0.16
(A)	BERT (B)	0.67	0.68	**0.62**	0.46	0.61
(B)	ELECTRA (L)	0.71	**0.71**	0.52	0.51	0.61
(C)	RoBERTa (B)	0.62	0.66	0.55	0.50	0.58
(D)	RoBERTa (L)	**0.74**	0.69	0.52	**0.56**	0.62
(E)	RoBERTa (L)	**0.74**	0.70	0.59	0.51	**0.64**
(F)	XLM-R	0.66	0.65	0.50	0.53	0.58
(G)	BERTweet	0.73	**0.71**	0.54	0.48	0.62
(H)	MiniLM	0.50	0.53	0.54	0.50	0.52
ALL (A-H)		**0.74**	**0.71**	0.61	0.52	**0.64**
(A)+(B)+(C)+(D)+(E)		0.72	0.69	0.61	0.55	**0.64**

Table 7: **Test set** results for Task B. The final row is the test-set score for the ensemble that performed best on the development set of Task B.

els we trained for Task A, which has an F_1 score of 0.85. Here, the best ensemble on the development set degrades in performance from 0.87 F_1 on the development set to 0.84 F_1 on the test set, suggesting this ensemble may have overfit to the development set. We also see that the simple average of predictions from all models we trained for Task A gives us a result that does not degrade from the development set to the test set.

When considering our individual Task A models, we see little to no drop in performance (0.0 to 0.2 macro F_1 difference) and models B, C, F even improve on this test set. GPT-3 prompting also shows no difference in performance on development and test, although performance is lower than for most of our individually trained Task A models. When comparing with the individual models we trained, we see that our best ensemble on the development set out-performs our best individual model (L) for this task. Submitting the best ensemble on the development set rather than this best individual model leads to an improvement in both precision (from 0.75 to 0.77) and macro average F_1 (from 0.83 to 0.84).

Tasks B and C Results for Tasks B and C show a stronger drop in performance when comparing the development set results to the test set results. This trend applies to not just our best performing ensemble on development, but to all individual models we trained. Despite this fact, our submitted ensemble of models which performed best on the development set also achieves the best performance on the test set, compared to both a simple average of all models we trained and compared to individual model performance.

When comparing our performance between Tasks B and C, we see that our approach of training models for Task C using a combined loss for Task B and C does allow us to achieve good results on Task B as well, when mapping the Task C fine-grained predicted labels to Task B coarse-grained labels. For these tasks, we also explored a wider range of models compared to Task A, which added to the diversity of model predictions we could select from for our ensemble.

		F_1 score											
Base Model		**1.1**	**1.2**	**2.1**	**2.2**	**2.3**	**3.1**	**3.2**	**3.3**	**3.4**	**4.1**	**4.2**	**Macro**
Majority Baseline		0.0	0.0	0.35	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.03
(A)	BERT (B)	0.33	0.66	0.60	0.54	0.31	0.57	0.53	0.53	0.00	0.22	0.58	0.44
(B)	ELECTRA (L)	0.44	0.59	0.61	0.51	0.45	0.55	0.45	0.32	0.00	0.27	0.54	0.52
(C)	RoBERTa (B)	0.37	0.52	0.58	0.53	0.30	0.58	0.38	0.27	0.11	0.39	0.59	0.42
(D)	RoBERTa (L)	0.56	**0.71**	0.61	0.50	0.43	0.61	0.51	0.53	0.22	**0.50**	**0.66**	0.53
(E)	RoBERTa (L)	0.38	0.69	0.62	0.51	0.33	0.63	0.45	0.15	0.18	0.35	0.61	0.45
(F)	XLM-R (L)	0.56	0.59	0.56	0.47	0.40	0.55	0.49	0.40	0.00	0.30	0.53	0.44
(G)	BERTweet (L)	0.50	0.68	0.61	0.53	0.38	0.60	0.46	0.15	**0.22**	0.38	0.61	0.47
(H)	MiniLM (L12)	0.56	0.46	0.45	0.28	0.24	0.58	0.45	0.25	0.13	0.44	0.51	0.40
(I)	ALBERT v2 (XL)	0.50	0.50	0.47	0.43	0.31	0.49	0.44	0.40	0.00	0.36	0.43	0.39
ALL (A-I)		0.37	0.68	0.65	**0.57**	0.37	0.64	0.53	0.38	0.00	**0.50**	0.60	0.48
(B)+(D)+(E)+(F)		**0.59**	0.69	**0.66**	0.56	**0.51**	**0.67**	**0.54**	**0.67**	0.18	**0.50**	0.65	**0.56**

Table 8: Per-class and macro-F1 **development set** results for Task C. The final row is for the best performing ensemble on the development set of Task C.

		F_1 score											
Base Model		**1.1**	**1.2**	**2.1**	**2.2**	**2.3**	**3.1**	**3.2**	**3.3**	**3.4**	**4.1**	**4.2**	**Macro**
Majority Baseline		0.0	0.0	0.35	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.03
(A)	BERT (B)	0.43	0.59	0.54	0.52	0.37	**0.65**	0.50	0.23	0.00	0.24	0.39	0.41
(B)	ELECTRA (L)	0.46	0.65	**0.60**	0.55	0.43	0.57	0.37	0.00	0.00	0.38	0.49	0.41
(C)	RoBERTa (B)	0.33	0.53	0.52	0.51	0.40	0.57	0.39	0.26	**0.20**	0.37	0.45	0.41
(D)	RoBERTa (L)	**0.59**	**0.68**	0.57	0.54	0.41	0.51	0.52	0.23	0.00	**0.49**	**0.55**	0.46
(E)	RoBERTa (L)	0.46	0.67	0.56	**0.58**	0.41	0.62	0.52	0.22	0.00	0.36	0.50	0.44
(F)	XLM-R (L)	0.35	0.59	0.54	0.50	0.34	0.47	0.45	0.28	0.17	0.27	0.48	0.40
(G)	BERTweet (L)	0.48	0.68	0.57	0.57	0.46	0.59	0.46	0.21	0.00	0.21	0.51	0.43
(H)	MiniLM (L12)	0.21	0.45	0.48	0.25	0.27	0.57	0.49	0.00	0.00	0.27	0.45	0.31
ALL (A-I)		0.48	0.66	0.59	0.56	**0.47**	0.63	**0.54**	0.28	0.00	0.39	0.50	0.46
(B)+(D)+(E)+(F)		0.53	**0.68**	0.59	0.57	0.44	0.62	0.53	**0.29**	0.00	0.38	0.53	**0.47**

Table 9: Per-class and macro-F1 **test set** results for Task C. The final row is the test-set score for the ensemble that performed best on the development set of Task C.

6.1 Error Analysis

Since the Task C categories are the most fine-grained, we investigated for which Task C categories our model under-performed by comparing the ensemble predictions on the development set with the task C development set gold labels. A confusion matrix from the development set is reported in Figure 1. Table 10 show examples of where the data was mislabeled for each. The majority of the development set errors stemmed from the failure of the model to distinguish between:

- Labels 2.2 (Aggressive and emotive attacks) and 3.1 (Casual use of gendered slurs, profanities and insults)
- Labels 2.1 (Descriptive attacks) and 2.2 (Aggressive and emotive attacks)
- Labels 2.1 (Descriptive attacks) and 3.2 (Immutable gender differences and gender stereotypes)

2.2 (Aggressive and emotive attacks) vs. 3.1 (Casual use of gendered slurs, profanities and insults) This was the most frequent error category on the development set for our best ensemble of models. Our model often confused the two labels due to the subtlety of casual versus aggressive use of gendered slurs. Documents with both labels may contain such slurs, but the implications of those slurs can be quite different. For example, in Table 10, the third example shows the speaker's intention and feeling by directly calling someone a slur. This is different from examples labeled as 3.1 (Casual use of gendered slurs, profanities and insults), where the writers are not intentionally attacking the women but rather casually expressing sexism through their use of slurs. For example, in the fourth example in Table 10 the speaker is not directly attacking women but rather grouping all women into loyal and disloyal, an unfair artificial categorization, which is emphasized with the speaker's use of a gendered slur. We again could improve our models by including additional silver labeled samples that contain these gendered slurs, and manually placing them into two groups of examples with casual mentions of gendered slurs or direct, intentional use of slurs.

rewire_id	Gold	Pred	Text
12120	2.1	2.2	*Treat a ***** like a lady, and treat a lady like a *****.*
12306	2.2	2.1	*How do block people from following me? A bunch of nasty women are following me on gab.i'm disgusted!!*
13595	2.2	3.1	*That,Äôs is more likely your meth ***** mom,Äôs job. Go **** yourself, Nazi ***.*
17438	3.1	2.2	*I'd rather have a loyal friend than a dumb *****
16460	3.2	2.1	*Do you want answers or not? You asked trp and we answered. Just because you don't like the answer doesn't make it false - that's a woman's line of thinking.*

Table 10: **Development set** examples from our error analysis. The predicted labels are generated using best performing ensemble from Table 8 (*B, D, E, F*). Gendered slurs in example text have been omitted and replaced with ****.

2.1 (Descriptive attacks) vs. 2.2 (Aggressive and emotive attacks) This was the second most frequent error category on the development set for our best ensemble of models. As stated on the shared task website label schema, 2.1 (Descriptive attacks) is for descriptive attacks that disparage women through generalizations, which differs from 2.2 (Aggressive and emotive attacks) where the speaker expresses negative sentiment towards women in the document. Many of these 2.2 (Aggressive and emotive attacks) documents contain the first person pronoun "I", expressing how the speaker is feeling. A future direction to improve the ability of our model to distinguish a sentence that displays direct negative emotions from the speaker, versus a generalized negative emotion, is to include additional silver labeled training data that contain both "I"-statements, and such attacks on women.

2.1 (Descriptive attacks) vs 3.2 (Immutable gender differences and gender stereotypes) This was the third most frequent error category on the development set for our best ensemble of models. This case is unique in the sense that the majority of cases were documents labeled 3.2 being mislabeled as 2.1 (Descriptive attacks), rather than the opposite. For example, in Table 10, the fifth example

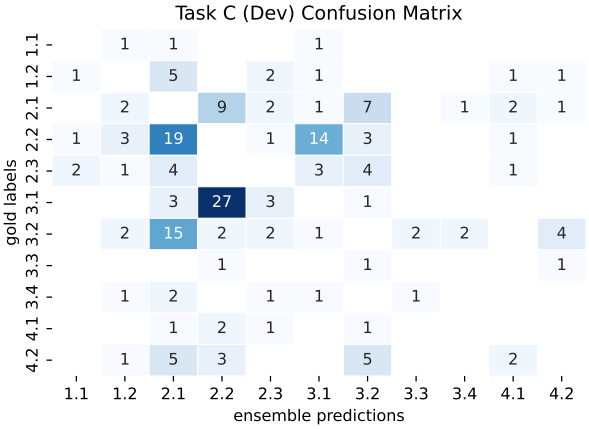

Figure 1: Confusion matrix for Task C development set.

has the phrase *"that's a woman's line of thinking"*. Here, the intention of the speaker is not directly attacking one or more women, but rather expressing an attitude about how women think, which is a gendered stereotype. This statement may not be obviously sexist to all readers. Much of the intention of the sentences in the 2.1 (Descriptive attacks) labeled data (as in our second example in Table 10) is to directly attack women through sexist generalizations. We believe our model was not able to distinguish these two fine grained cases because it could not distinguish if a statement was directly or indirectly sexist.

We see errors for category 3.2 (Immutable gender differences and gender stereotypes) as some of the most challenging for our models to overcome, due to how stereotypes around gender differences may be stated indirectly: these documents may not include any explicit profanity or gendered slurs, or other obvious hallmarks of sexism in the text. Additionally, examples of class 3.2 (Immutable gender differences and gender stereotypes) are less frequent in the gold training and development data. In order to make improvements to our model, we believe collecting additional annotated data using the annotation guidelines written by the shared task organizers is the right direction.

7 Conclusion

In this paper, we described our team's submission to SemEval-2023's "Explainable Detection of Online Sexism" shared task, that aims to identify whether or not a given content is sexist (Task A), as well as the broad (Task B) and fine-grained (Task C) class of sexist content present in the given text (e.g., prejudice, animosity, derogation, etc.). We find that utilizing a combination of data augmentation – both

through publicly available datasets as well as silver labeling – and ensembling different transformer-models gives us the best performance. In addition, we also explored using in-context learning using GPT-3, and found that it yields mixed results, helping us achieve better performance for Task A but not for Tasks B or C.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ke-Li Chiu and Rohan Alexander. 2021. Detecting hate speech with GPT-3. *CoRR*, abs/2103.12407.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. 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*, pages 67–73.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International World Wide Web Conference*, pages 29–30.

Paula Fortuna, Juan Soler, and Leo Wanner. 2020. Toxic, hateful, offensive or abusive? what are we really classifying? an empirical analysis of hate speech datasets. In *Proceedings of the 12th language resources and evaluation conference*, pages 6786–6794.

Tanmay Garg, Sarah Masud, Tharun Suresh, and Tanmoy Chakraborty. 2022. Handling bias in toxic speech detection: A survey. *ACM Computing Surveys*.

Raphael Gontijo-Lopes, Yann Dauphin, and Ekin Dogus Cubuk. 2022. No one representation to rule them all: Overlapping features of training methods. In *International Conference on Learning Representations*.

Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

Jiyun Kim, Byounghan Lee, and Kyung-Ah Sohn. 2022. Why is it hate speech? masked rationale prediction for explainable hate speech detection. *arXiv preprint arXiv:2211.00243*.

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*, Toronto, Canada. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*. OpenReview.net.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Todor Markov, Chong Zhang, Sandhini Agarwal, Tyna Eloundou, Teddy Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2022. A holistic approach to undesired content detection in the real world. *arXiv preprint arXiv:2208.03274*.

Raymond T Mutanga, Nalindren Naicker, and Oludayo O Olugbara. 2020. Hate speech detection in twitter using transformer methods. *International Journal of Advanced Computer Science and Applications*, 11(9).

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 9–14. Association for Computational Linguistics.

Alex Nikolov and Victor Radivchev. 2019. Nikolov-radivchev at semeval-2019 task 6: Offensive tweet classification with bert and ensembles. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 691–695.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International World Wide Web Conference*, pages 145–153.

John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4296–4305.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4512–4525. Association for Computational Linguistics.

Francisco J. Rodríguez-Sanchez, 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. *Proces. del Leng. Natural*, 67:195–207.

Paul Röttger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective nlp tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 175–190.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. Hatecheck: Functional tests for hate speech detection models. 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 41–58.

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

Thanh Tran, Yifan Hu, Changwei Hu, Kevin Yen, Fei Tan, Kyumin Lee, and Se Rim Park. 2020. Habertor: An efficient and effective deep hatespeech detector. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7486–7502.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. 2020. Hyperparameter ensembles for robustness and uncertainty quantification. *Advances in Neural Information Processing Systems*, 33:6514–6527.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR.

Cinna Julie Wu, Mark Tygert, and Yann LeCun. 2017. A hierarchical loss and its problems when classifying non-hierarchically. *PLoS ONE*, 14.

Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian D Davison, April Kontostathis, and Lynne Edwards. 2009. Detection of harassment on web 2.0. In *Proceedings of the Content Analysis in the Web 2.0 (CAW2.0) Workshop at WWW2009*.

A Appendix

A.1 Training Hyper-parameters

We chose the Adam optimizer (Kingma and Ba, 2015) with $(\beta_1, \beta_2) = (0.9, 0.98)$ and an initial learning rate of $5e$-5, gradient accumulation of 4 and different batch sizes for different base models. For regularization, we use weight decay of $1e$-2. To train the model for Task A, we use binary cross-entropy criterion weighted with respect to the proportion of positive and negative samples in the training data set. For Task B and C we use the loss described in Section 4.1 Equation 1.

We train each model for 20 epochs with early stopping and a patience of 3 epochs without improvement on validation set loss.

For Task B and C we introduced the hyperparameter β when combining the loss functions for both

tasks (as explained in Section 4.1). All models we trained for Tasks B and C use $\beta = 0.25$ except for model (E) which uses $\beta = 0.1$.