Penta-nlp at EXIST 2024 Task 1-3: Sexism Identification, Source Intention, Sexism Categorization In Tweets

Notebook for the EXIST Lab at CLEF 2024

Fariha Tanjim Shifat[1,*,†], Fabiha Haider[1,*,†], Md Sakib Ul Rahman Sourove[1], Deeparghya Dutta Barua[1], Md Farhan Ishmam[1,3], Md Fahim[1,2,*] and Farhad Alam Bhuiyan[1,*]

[1]*Research and Development, Penta Global Limited, Bangladesh*
[2]*CCDS Lab, IUB, Bangladesh*
[3]*Islamic University of Technology, Bangladesh*

Abstract

Social media platforms contains a vast user base and offers ease with which information can be shared. This can adversely facilitate the spread of sexist content which is infeasible for human monitoring and filtering. This paper investigates the automated detection of sexism in tweets using Natural Language Processing (NLP) techniques. Sexist tweets can create a hostile online environment and perpetuate harmful stereotypes. Manual identification is impractical due to the vast amount of data. The research proposes a system utilizing machine learning models to analyze text, identify bias and discriminatory language patterns, and flag tweets for moderation. The fourth edition of EXIST shared task 2024 Tweets Dataset, containing labeled English and Spanish tweets, is used to train and evaluate the models. The system explores various approaches, including TF-IDF with different classifiers (SVM, XGB, RF), Long Short-Term Memory (LSTM) networks with and without attention mechanisms, and pre-trained transformer models (XLM-Roberta, mBERT, BETO). The effectiveness of different preprocessing techniques and the role of attention weights in identifying sexism are also explored. The paper outlines the methodology, experimental setup, and analysis of results, paving the way for further discussion on error analysis and conclusions in subsequent sections.

Keywords

Sexism identification, Tweets, Source intention detection, Sexism categorization, Multilingual Models, Natural Language Processing

1. Introduction

This research paper discusses topics related to specific content that may be sensitive or offensive, which some readers may find distressing. The intent is to analyze and understand the research work.

The commence and growth of internet have a profound impact on our social structure, the way we communicate, our relationships through the development of various social platforms. Twitter, one of such social platforms, has developed into a lively forum for sharing idea and discourse because of its succinct format and emphasis on real-time updates, with attractive features like hashtags, tags, etc. While such advancements of social platforms promotes connectivity and facilitates the spread of information it also tempts people to gain fame thorough views and likes, and throw inappropriate contents and comments in disguise of freedom of speech lacking empathy towards race, gender, religion [1]. Evidently sexism exists in Twitter in the form of sexist tweets sometimes intentionally while at other times unintentionally. These tweets and contents can range from blatant objectification and insults to more implicit bias and prejudice. Sexist tweets can create a hostile environment online, especially for women and other targeted groups. Identifying sexism in online platforms is crucial for a various number of reasons. Recognizing sexist content is essential to advancing equality and averting social harm. The propagation of detrimental stereotypes and biases through such information

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
*Corresponding authors.
†These authors contributed equally.
fariha.tanjim.shifat@gmail.com (F. T. Shifat); fabihahaider4@gmail.com (F. Haider); fahimcse381@gmail.com (M. Fahim)

contributes to gender inequality and has a detrimental effect on people's mental health and self-esteem [2]. We can promote an inclusive and respectful culture by addressing sexist content, making sure that everyone feels respected and secure.

The sheer volume of such sexist tweets makes it difficult to identify sexism manually, requiring the immediate need for automated solutions. Natural Language Processing (NLP) can emerge as a successful tool in recognizing such harmful contents and filtering them out [3]. It can prevent people to post contents that goes against community standards creating a safer and more acceptable platform. Using natural language processing (NLP) techniques, we can create automated systems that can recognize sexist tweets with accuracy. Text content can be analyzed by these technologies, which can also spot bias and discriminatory language patterns and flag tweets for moderation or additional review.

EXIST aims to capture sexism in a broad sense, from explicit misogyny to other subtle expressions that involve implicit sexist behaviours. EXIST is a series of scientific events and shared tasks on sexism identification in social networks [4], [5]. The EXIST 2024 Tweets Dataset contains more than 10,000 labeled tweets, both in English and Spanish. Based on the labeled data, the tasks were to identify sexist tweets among them, the underlying intention of the author, and if it contained sexism at multiple degrees. The challenge lies in the nuanced and context-dependent nature of language on social media. Tweets often use slang, sarcasm, or coded language that can obscure sexist intent, while URLs can lead to external content that may contain sexist material not immediately evident in the tweet itself. Mentions and hashtags can complicate analysis by linking tweets to broader conversations or by being used to target specific individuals. Emojis add another layer of complexity, as their meanings can vary widely depending on context and cultural interpretation. These factors make automated detection systems prone to errors, requiring sophisticated algorithms and often human oversight to accurately identify and address sexism in tweets.

Our approach involved training different Machine Learning (ML) models to capture the sexist pattern in the contents and choose the one that performs the best on the validation dataset. The machine learning models included TF-IDF+SVM [6], TF-IDF+XGB [7], TF-IDF+RF [8], LSTM [9], LSTM+Attention [10], XLM-Roberta [11], mBert [12], BETO [13]. We yielded results for different ways of preprocessing the data and further finetuned the models based on best results. Additionally, we attempted to find out the provoking words that contributed in the semantic meaning of sexism reflected through the attention weights of those words. Taking into accounts the attention weights of the sentence representation helped the same model to yield the best result for Task 1, Task 2 and Task 3. The results show that the Bert based models are well trained to capture the pattern of sexism when considering the attention layer. The performance metrics used were accuracy and F1 score and we could obtain an accuracy of 84.80%, 72.06% and 88.25% and F1 score of 84.76%, 51.43% and 54.77% for Task 1, Task 2 and Task 3, respectively. To do better modeling and include different explainability results, we adopt different training and explainable experiments from EDAL, ITPT, HateXplain [14, 15, 16] papers. In the remainder of this paper, we present the Problem Description in Section 2, Methodology in Section 3, Experimental Setup in Section 4, Result Analysis in Section 5, Error Analysis in Section 6 and Conclusion in Section 7.

2. Problem Description

2.1. Task Descriptions

In this work, we have addressed tasks 1, 2, and 3. The task descriptions are listed as per the guidelines.

1. **Task 1 (Sexism Identification in Tweets):** Given a tweet, this subtask aimed to classify whether a tweet contains sexist expressions or behaviors. The tweet can be sexist itself, describe a sexist situation or critic sexist behavior. It is a binary classification problem. We needed to label 'YES' or 'NO' to the tweets.

2. **Task 2 (Source Intention in Tweets):** This subtask aims to classify each tweet according to the intention of the person who wrote it. This is a multi-class classification problem. The classes are as follows:
    - **DIRECT:** The intention is to write a message that is sexist itself.
    - **REPORTED:** The author intends to report or describe a sexist situation or event suffered by a woman or women in the first or third person.
    - **JUDGEMENTAL:** The author intends to be judgemental since the tweet describes sexist situations or behaviors to condemn them.
    - **NO:** The tweet is detected as not sexist in subtask 1.
3. **Task 3:** This subtask categorizes the tweets according to the type of sexism. This is a multi-label classification problem with 5 labels. So more than one class can be assigned to each tweet. The labels are as follows:
    - **IDEOLOGICAL-INEQUALITY:** The tweet discredits the feminist movement, rejects inequality between men and women, or presents men as victims of gender-based oppression.
    - **STEREOTYPING-DOMINANCE:** The tweet expresses false ideas about women that suggest they are more suitable to fulfill certain roles (mother, wife, family caregiver, faithful, tender, loving, submissive, etc.), or inappropriate for certain tasks (driving, hard work, etc.), or claims that men are somehow superior to women.
    - **OBJECTIFICATION:** The tweet presents women as objects apart from their dignity and personal aspects or assumes or describes certain physical qualities that women must have to fulfill traditional gender roles (compliance with beauty standards, hypersexualization of female attributes, women's bodies at the disposal of men, etc.).
    - **SEXUAL-VIOLENCE:** The tweet includes or describes sexual suggestions, requests for sexual favors, or harassment of a sexual nature (rape or sexual assault).
    - **MISOGYNY-NON-SEXUAL-VIOLENCE:** The tweet expresses hatred and violence towards women, different from that with sexual connotations.
    - **NO:** When none of the 5 labels are assigned to the tweet.

2.2. Dataset Statistics

The dataset includes over 10,000 tweets both in Spanish and English. The train, dev, and test sets contain 6064, 934, and 2076 tweets respectively. These numbers are after discarding the non-labeled samples in the gold standard. The distribution between both languages has been somewhat balanced to tackle the issue of biases. The exact figures are in Table 1.

Table 1
Tweets Dataset, containing training, development and test splits. The dataset contained both English and Spanish language.

Data Splits	Total Samples	Language Wise Samples	
		EN	ES
Train	6064	2870	3194
Dev	934	444	490
Test	2076	978	1098

Task 1 is a binary classification problem with 'YES' and 'NO' labels. Task 2 is a multi-class classification problem with 3 sexism sources and one 'NO' label. Task 3 is a multi-label classification problem with 5 different labels and a 'NO' label when none of the labels are assigned. The distribution is not balanced. Each tweet is labeled by six different annotators. The gold label is the average of the labels. The exact figures of the distribution is in Table 2.

Table 2
Class or Label-wise distribution in the dataset for Task 1, Task 2 and Task 3.

Class/Labels	Samples	
	Train	Dev
Task 1		
NO	3376	479
YES	2697	455
Task 2		
NO	3367	479
DIRECT	1294	204
REPORTED	459	83
JUDGEMENTAL	376	75
Task 3		
IDEOLOGICAL-INEQUALITY	1113	212
STEREOTYPING-DOMINANCE	1423	241
OBJECTIFICATION	1103	183
SEXUAL-VIOLENCE	675	123
MISOGYNY-NON-SEXUAL-VIOLENCE	856	158
NO	3367	479

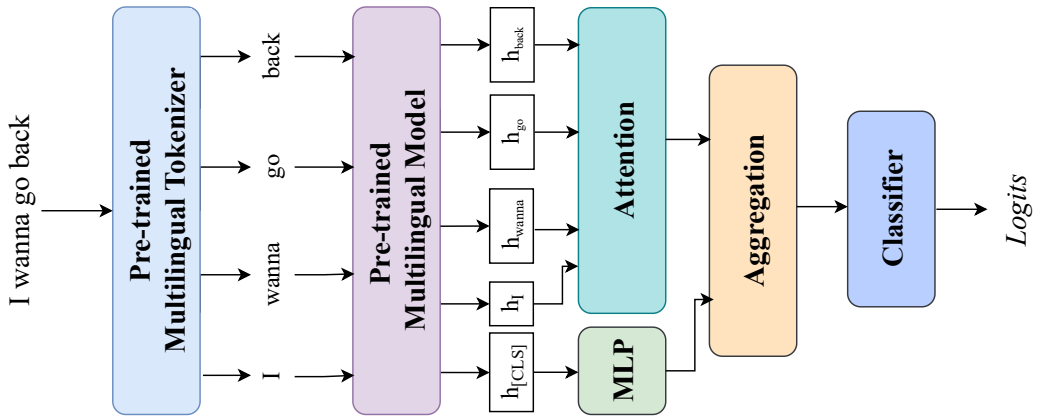


Figure 1: Model Architecture. Single sentence is passed through the tokenizer to separate the tokens. They are then passed to the model which gives the encoded format of those tokens. Encoded tokens except CLS are passed to the attention module whereas CLS is passed to the MLP layer which then is aggregated using various methods and then this representation is used for classification. The logits are yield by the classifier.

3. Methodology

In this section, we will outline our methodology. Given the multilingual nature of the dataset encompassing both English and Spanish texts, we employ a multilingual pretrained model. We further refine this model through fine-tuning on the dataset. Our model architecture comprises five key components: i) Pretrained model backbone, ii) Sentence Representation from CLS token, iii) Attention-based Context Vector, iv) Feature Aggregation Module, and v) Classifier Head

3.1. Pretrained Multilingual Model as Backbone

An input sentence S is passed into the pretrained multilingual tokenizer to obtain the tokens of the sentence $S = \{t_{[CLS]}, t_1, t_2, \ldots, t_n, t_{[SEP]}\}$, where t_i represents the i-th token and $t_{[CLS]}$ & $t_{[SEP]}$ are

the special tokens. The tokens are passed into the pretrained multilingual model to achieve contextual representations for each token t_i, denoted as $H = \{h_{[CLS]}, h_1, h_2, \ldots, h_n, h_{[SEP]}\}$, where h_i represents the contextual representation of token t_i. Specifically, we extract the last layer hidden representations of the pretrained model, which are further fine-tuned during the dataset training.

3.2. Sentence Representation from CLS Token

To get the representation for the entire sentence, we use the representation of [CLS] token $h_{[CLS]}$. This representation is passed into a Single Layer Perception to get the enhanced representation.

$$h^{'}_{CLS} = W_{CLS} \cdot h_{\text{CLS}} + b_{CLS}$$

3.3. Attention-based Context Vector

As the tasks are natural language understanding tasks where individual words hold distinct predictive significance, we integrate an attention-based network to ascertain word importance. By accounting for the significance of each word, we construct a context vector for the sentence. We only consider the representations of the words and exclude the representations of the special tokens.

Once contextual representations H are obtained for a sentence S, an additional attention layer is added to compute learnable attention scores α_i for each token t_i in H, and its calculation is as follows:

$$\alpha_i = \textit{softmax}(W \cdot h_i + b),$$
$$i = 1, 2, \ldots, n$$

This results in a set of *attention_scores* $= \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ corresponding to the tokens in sentence S. These attention scores collectively represent the overall attention distribution across the sentence, indicating the relative importance or relevance of each token to the context of the entire sentence. After finding attention scores for each token, we find the context vector for the sentence S by multiplying the contextual representations of token t_i with its attention score α_i.

$$c = \sum_{i=1}^{n} \alpha_i \cdot h_i$$

3.4. Feature Aggregation Module

In this section, we consolidate various representations of the sentence S. We obtain two distinct representations: one based on the CLS token $h^{'}_{CLS}$ and the other based on attention-based context vector c. The aggregation of these representations is performed using two distinct techniques considering the two following techniques:

- **Concat-based Aggregation:** For concat-based aggregation, we just simply concatenate $h^{'}_{CLS}$ and c to get the aggregated representation z as follows:

  $$z\_\text{concat} = \text{concat}[h^{'}_{CLS}, c]$$

  Then z_concat is passed into single layer MLP layer to get the final combined representation z where $z = \text{MLP}(z\_\text{concat})$

- **Element wise Addition based Aggregation:** In this aggregation, we combine $h^{'}_{CLS}$ and c by summing them element wise as follows

  $$z_{suumed} = h^{'}_{CLS} + c$$

  Then z_suumed is passed into single layer MLP layer to get the final combined representation z where $z = \text{MLP}(z\_\text{suumed})$

3.5. Classifier Head

After finding the aggregated feature representation z, it is fed into a classification layer. The representation is the logits z is employed for the classification process by the following:

$$z^{'} = W \cdot z + b$$

Finally, we calculate the Cross-Entropy (CE) loss based on $z^{'}$ with the ground truth.

4. Experimental Setup

4.1. Model Selection

For the tasks, our initial approach involved a thorough exploration of various model types to determine the most appropriate one. Our investigation led us to examine three distinct categories: Machine Learning (ML) Models, Deep Learning Models, and Transformer-based Pretrained Models. For machine learning models, we considered Support Vector Machine (SVM), Random Forest (RF), and XGBoost. Employing TF-IDF as our feature extractor, we processed each sentence through this mechanism before inputting it into the ML models for classification. For task 3, we consider Logistic Regression instead of XGBoost.

In our deep learning methodologies, we leverage both LSTM [17] and LSTM + Attention models. As for transformer-based approaches, we've explored XLM-RoBERTa [18], mBERT [19], and BETO [20]. The outcomes are detailed in Table 3 for the development set. These model selections were driven by the presence of two distinct languages within the dataset. The total unique words for TF-IDF experiments were 39613 after deleting the punctuations, auxiliaries, and spaces. For LSTM experiments, we have used embedding dim = 50, hidden units = 64 of the LSTM layer, and output dim = 256 of a fully connected layer with a learning rate of 0.001. The experiments were done in 20 epochs. For the transformer-based models, we fine-tuned the pre-trained models. We use 10 epochs with a batch size of 32

Table 3
Comparing the Performance of Different Models in Validation Dataset for Task 1 and Task 2

Model	Performance Metric			
	Task 1		Task 2	
	Dev Acc↑	Dev F1↑	Dev Acc↑	Dev F1↑
ML Models				
TF-IDF + SVM	70.88	69.73	61.12	26.70
TF-IDF + XGB	76.55	76.26	66.11	37.24
TF-IDF + RF	68.09	66.40	61.12	26.52
Deep Learning Models				
LSTM	71.73	73.14	64.44	31.92
LSTM + Attention	74.95	74.82	65.28	32.87
Transformer based Models				
XLM-RoBERTa	**83.94**	**83.94**	**73.13**	**56.55**
mBERT	81.05	81.04	71.58	53.16
BETO	80.73	80.72	70.99	54.27

Table 3 presents the results of different models for Task 1 and Task 2 datasets, while Table 4 displays the model performance specifically for Task 3 on the development set. Analysis of Table 3 reveals that classical ML models exhibit competitiveness against deep learning counterparts. Notably, employing TF+IDF with XGBoost yields superior results to deep learning models. Incorporating the attention module leads to a notable 3% enhancement in accuracy for Task 1 and 1% improvement Task 2. Additionally, fine-tuning pretrained models demonstrates substantial improvements ranging from 6-9%.

Table 4
Comparing the Performance of Different Models in Validation Dataset for Task 3

Model	Performance Metric Task 3	
	Dev Acc↑	Dev F1↑
ML Models		
TF-IDF + LogisticRegression	85.89	60.85
TF-IDF + SVM	84.85	55.79
TF-IDF + RF	84.85	55.79
Transformer based Models		
XLM-RoBERTa	87.69	47.76
mBERT	**87.86**	**48.21**
BETO	87.41	47.13

Among these, XLM-Roberta demonstrates optimal performance for both Task 1 and Task 2, thereby being selected as the final model for further experimentation. Turning to Task 3, as indicated in Table 4, mBERT marginally outperforms XLM-RoBERTa. Consequently, mBERT is chosen as the final model for Task 3.

4.2. Preprocessing

The datasets retrieved from Twitter contain additional information such as usernames and URLs alongside the original posts and comments. To gauge their impact, we conducted experiments using various preprocessing methods. These techniques included removing usernames, URLs, punctuation, and emojis. The outcomes of these experiments are presented in Table 5. From the table, we can see that removing url from the tweets improves the model performance for task 1 and task 3. For task 2, no preprocessing is helpful.

Table 5
Effect of different preprocessing in dev set.

Task	Model	Preprocessing	Performance Metric	
			Dev Acc↑	Dev F1↑
Task 1	XLM-RoBERTa	No preprocessing	83.94	83.94
		Username Removed	83.73	83.73
		URL Removed	**84.80**	**84.76**
		Punctuation and Emoji Removed	84.37	84.36
Task 2	XLM-RoBERTa	**No preprocessing**	**72.06**	**51.43**
		Username Removed	71.94	52.32
		URL Removed	71.11	52.00
		Punctuation and Emoji Removed	71.11	53.82
Task 3	mBERT	No preprocessing	87.86	38.21
		Username Removed	87.76	40.94
		URL Removed	**88.25**	**54.77**
		Punctuation and Emoji removed	87.86	48.72

4.3. Settings

For the hyper parameter settings, we also investigated with different values of them. We did ablation studies on learning rate with two different values [1e-5, 2e-5], on batch size with values [16, 32] and on random seed with values [0, 42]. The experimented results are reported in the Table [6, 7, 8]. Considering

Table 6
Effect of different learning rates in dev set.

Task	Learning Rate	Performance Metric	
		Dev Acc↑	Dev F1↑
Task 1	2×10^{-5}	**84.90**	**83.90**
	1×10^{-5}	84.80	84.76
Task 2	2×10^{-5}	**72.06**	**51.43**
	1×10^{-5}	70.87	51.93
Task 3	2×10^{-5}	**88.25**	**54.77**
	1×10^{-5}	85.49	52.36

Table 7
Effect of different batch sizes in dev set.

Task	Batch Size	Performance Metric	
		Dev Acc↑	Dev F1↑
Task 1	**32**	**84.90**	**83.90**
	16	82.01	81.92
Task 2	32	72.06	51.43
	16	**73.25**	**54.42**
Task 3	**32**	**88.25**	**54.77**
	16	85.35	51.69

Table 8
Effect of different seed values for model perforamce in dev set.

Task	Random Seed	Performance Metric	
		Dev Acc↑	Dev F1↑
Task 1	**42**	**84.90**	**83.90**
	0	84.05	84.03
Task 2	**42**	**73.25**	**54.42**
	0	71.70	52.32
Task 3	**42**	**88.25**	**54.77**
	0	85.39	54.52

those results, we set a learning rate of 2e-5, random seed = 42 and batch size 32 = for tasks 1 & 3 and 16 for task 3 for our model. We use AdamW optimizer in our experiments with betas = (0.9, 0.99).

All experiments were conducted using Python (version 3.12) and PyTorch, leveraging the free NVIDIA Tesla P100 GPU provided by Kaggle. For the transformer based models we consider Huggingface *transformers* library. All the transformer based models were run for 10 epochs.

4.4. Evaluation Metrics

When assessing the effectiveness of the models, we consider different performance metrics. Mainly we focus on Accuracy, Macro-F1, and ICM for our performance as our evaluation metircs.

4.4.1. Accuracy

Accuracy measures the proportion of correctly classified instances among all instances. It is calculated by dividing the sum of true positives (correctly predicted positive instances) and true negatives (correctly predicted negative instances) by the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

4.4.2. Macro F1 Score

The F1 score is the harmonic mean of precision and recall. In the macro F1 score, each class is given equal weight, and the mean of these F1 scores across all classes is calculated. Macro F1 Score is calculated as follows:

$$\text{F1-Score}_i = \frac{2}{\frac{1}{Precision_i} + \frac{1}{Recall_i}}$$

$$\text{Macro-F1} = \frac{1}{N}\sum_{i=1}^{N} \text{F1-Score}_i$$

4.4.3. ICM

The ICM metric functions as a similarity measure employed to assess the likeness between model-generated outputs and the actual ground truth in classification tasks. It does so by comaring the Information Content of catrgories presented through its features with the help of a Similarity function. It extends the principles of Pointwise Mutual Information, a common metric used for evaluating relationships between words. When all parameters in the ICM formula are set to 1, it becomes equivalent to PMI. This means ICM can capture similar relationships as PMI but with more flexibility. ICM helps evaluate how well a system's output aligns with the ground truth by comparing the information content of the categories they represent. (ICM) is adopted for all tasks and evaluation types (hard-hard, hard-soft, soft-soft). **ICM-soft** is an extenstion of ICM for evaluating hierarchical multi-label classification tasks where there might be disagreements in the ground truth. It can can handle both soft system outputs and soft ground truth.

5. Result Analysis

Table 9 exhibits the results of various sentence representation extractors employing different aggregation techniques across three tasks on the validation dataset. Regarding task 1, employing the CLS token-based representation yields an accuracy of 84.90% and an F1 score of 83.90%. Transitioning to context vector representation through an attention feature extractor results in a slight increase in the F1 score. Furthermore, combining both types of representation through addition-based aggregation shows a 2% enhancement in performance across the board. If we combine them using concat based aggregation techniques we see a small improvement in F1 score but not like the addition based aggregation one

Table 9
Effect of different aggregation processes of the CLS token in dev set.

Task	Model	Sentence Repr Extractor	Aggregation Method	Performance Metric	
				Dev Acc↑	Dev F1↑
Task 1	XLM-RoBERTa	CLS	-	84.90	83.90
	XLM-RoBERTa	Attention	-	84.15	83.94
	XLM-RoBERTa	Attention + CLS	Addition	**86.30**	**86.28**
	XLM-RoBERTa	Attention + CLS	Concat	84.90	84.88
Task 2	XLM-RoBERTa	CLS	-	72.06	51.43
	XLM-RoBERTa	Attention	-	72.77	52.65
	XLM-RoBERTa	Attention + CLS	Addition	73.13	54.65
	XLM-RoBERTa	Attention + CLS	Concat	**74.20**	**57.87**
Task 3	mBERT	CLS	-	88.12	54.77
	mBERT	Attention	-	87.97	51.50
	mBERT	Attention + CLS	Addition	84.40	53.50
	mBERT	Attention + CLS	Concat	**88.24**	**54.82**

For task 2, we experience similar trend in the performance where attention based sentence representation performs better than CLS token based sentence extractor. If we combine both CLS and

context vector getting from attention through addition based aggregation techniques, 1% improvement in accuracy and 2% improvement in F1 score. While we consider concat based aggregation techniques for combining both sentence level representation, the performance is further improved by 1.2% in accuracy and around 3% improvement in F1 score. For task 3, we also get better result than the CLS based baseline while we aggregate both sentence representations using concatenation.

Table 10
EXIST test results for submitted predictions with the best performing configuration along with CLS token aggregation on the dev set.

Task	Model [Agg. Type]	Hard-Hard	Rank	ICM-Hard	ICM-Hard Norm	Macro F1
Task 1	XLM-RoBERTa + Attention + CLS [Addition]	ALL	29	47.79	74.02	75.08
		EN	29	46.01	73.48	72.09
		ES	29	48.04	74.02	77.33
Task 2	XLM-RoBERTa + Attention + CLS [Concat]	ALL	9	20.89	56.79	48.56
		EN	13	12.03	54.16	44.31
		ES	10	27.34	58.54	51.96
Task 3	mBERT + Attention + CLS [Concat]	ALL	14	-25.97	43.97	43.79
		EN	17	-26.92	43.40	41.56
		ES	16	-27.41	43.88	44.60

The top-performing model is chosen to generate predictions for the test dataset across all three tasks. The performance results on the test dataset are detailed in Table 10, showcasing the ICM-Hard, ICM-Hard Norm, and Macro-F1 scores as our performance metrics. Our best-performing model achieves an F1 score of 75.01 for all tweets, with 72.09 and 77.33 F1 scores recorded for English and Spanish tweets, respectively. For task 2, the score is 48.56, while task 3 reaches 43.79 across all tweet data. The table shows that our model can predict Spanish tweets more effectively than English tweets.

6. Error Analysis

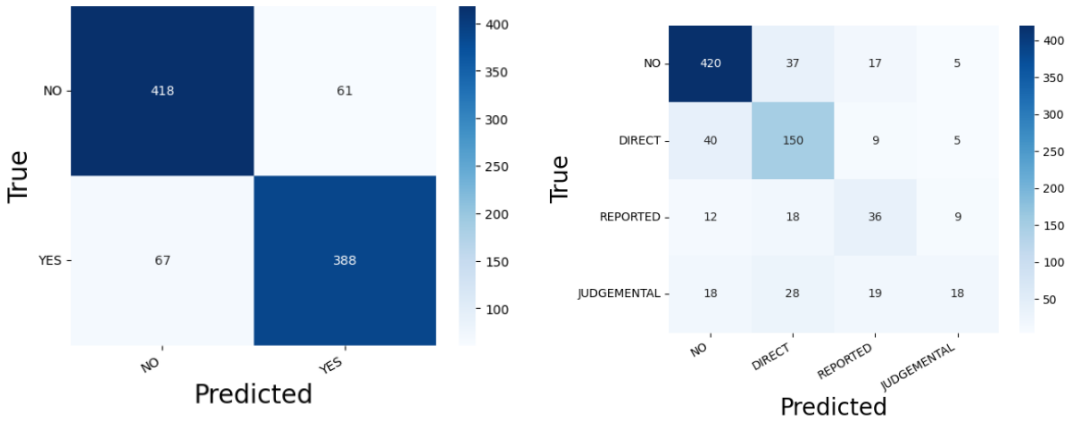
6.1. Confusion Matrix

The evaluation of our approach for Task 1 is done on the development dataset. The confusion matrix shown in 2(a) represents the performance of the classification. The number of True Positive is the number of correctly predicted positive cases. The model correctly predicted 388 cases as "Yes" (True Yes). True Negative cases is the number of correctly predicted negative cases. The model correctly predicted 418 cases as "NO" (True No). False Positive is the number of incorrectly predicted positive cases. The model incorrectly predicted 61 cases as "YES" when they were actually "NO" (False YES). False Negative (FN) is the number of incorrectly predicted negative cases. The model incorrectly predicted 67 cases as "NO" when they were actually "YES" (False NO). The model produced 806 correct classification as opposed to 128 misclassification.

Similarly, the confusion matrix is shown in 2(b) shows the result of the evaluation done on the development dataset for Task 2. As shown in the figure, out of 479 true NO labels, 420 cases were correctly predicted as NO, 150 were correctly predicted as DIRECT out of 204 true DIRECT cases, 36 were correctly predicted as REPORTED out of 65 true REPORTED cases and 18 were correctly predicted as JUDGEMENTAL out of 83 true JUDGEMENTAL cases. The model produced 624 correct classification as opposed to 217 misclassification.

6.2. Attention Heatmap

We have calculated the Layer Integrated Gradient attributions for few specific input tweets using Captum [21]. The attributions explain how each input element of a tweet contributes to the model's prediction for the target class. It means that the attention provided by the model to each tokens. So we name it attention heatmap. The darker the color in the cell of a token, the higher its attribution value



(a) Task 1 (b) Task 2

Figure 2: Confusion matrix with all classes for dev set with best performing configuration of each task.

and the higher its contribution to the target. We have considered both English and Spanish tweet for the experiment and creating the heatmap. The corresponding labels are also listed in the Figure 3. We can infer that for 'NO' label the most attentive tokens among the 4 tweets are *last, abuse, ada, in* and for 'YES' label the most attentive tokens among the 4 tweets are *economy, s, a, LAS*. This experiment was done considering the best performing model in the task 1.

Attention heatmap for tokens of a sample tweet	Lang	Label
0 _@ s age _ rana e _This _world _wouldn ' t _last _a _day _without _women 😴	EN	NO
0 _Israel ' s _hat red , _in tolerance , _and _abuse _is _dis gust ing .	EN	NO
0 _Con _25 _Leon es 🦁 _y _500 _más _en _las _grada s 🗣️ ‼ # E ko Fu ego 🔥 # La Man ada	ES	NO
0 _@ cris 7 ian _Yo _no _te _he _pregunta do _ja ja _# no tall men _Se _ve ía _venir .	ES	NO
0 _What _a _day _without _women _would _really _do _to _the _economy	EN	YES
0 _@ Bo dy sat nav _She ' s _not _a _nice _old _bu ddy . _She ' s _a _blood thir sty _old _bin t	EN	YES
0 _@ Ho t Em ma 3 _Sab roso _ culo ... _para _reven tar lo _a _ver gaz os _corazón _ 🔥	ES	YES
0 _NO _INS UL TES " 0 , 2 _segundos _des pu es " _P EDA ZO _GU ARRA _P UTA _V AGA _G ILI POL LAS	ES	YES

Figure 3: Attention heatmap for task 1 of a few samples of the dev set with XLM-RoBERTa and the best performing configuration.

6.3. Most Attentive Tokens

We also extracted the most attentive tokens for each category in the validation dataset for further analysis. We determine these tokens by extracting the highest average self-attention scores for each token across the entire category-wise validation dataset. Transformer-based models compute attention scores to gauge the relationship between $word_i$ and $word_j$ across various layers and heads. Considering a transformer-based model with L layers and H heads per layer, and a sequence length of n for a given sentence S, the resulting attention matrix has dimensions of $L \times H \times n \times n$. To compute the average attention score for token_i within sentence S, we follow this calculation:

$$\text{Avg\_Attn\_Score}_{i_S} = \frac{1}{L \cdot H \cdot n} \sum_{l=1}^{L} \sum_{h=1}^{H} \sum_{t=1, t \neq i}^{n} \text{Attention}_{i,t,h,l}$$

To find category-wise average attention score for token$_i$ on the validation dataset, we take the average of Avg_Attn_Score$_{i_S}$ for those sentences where token$_i$ appears in. After calculating the token attention scores, we arrange them in descending order and select the top K words. In Table 11, we present the category-wise most attentive tokens predicted by the top-performing model for both Task 1 and Task 2.

Table 11
List of most attentive tokens for tasks 1 and 2 for each class with the best performing configuration for each task.

Task	Label	List of Most Attentive Tokens [Top 10]
Task 1	NO	volant, slut, cum, hot , summer peligro, constante, staff, development , practic
	YES	penis, forever, death, economy , ika GPS, tomorrow, exam, gangbang, room
Task 2	NO	pun, wall, question, falso, auténtico modo, concert, syon, yes, making
	DIRECT	where, GPS, tomorrow, exam, room attention, foot, cla, without, uni
	REPORTED	wenr, school, bath, dom, GAL ILE, ingu, ebla, dia, ACIÓN
	JUDGEMENTAL	saben, ándose, cocina, make, sense want, mother, ape, ici, gas

The table presents a detailed breakdown of the most attentive tokens categorized by the top-performing model across two distinct tasks. In Task 1, where the classification pertains to identifying sexist content, the model identifies tokens such as "volant," "slut," and "cum" as prominent indicators for non-sexist content, while tokens like "penis," "forever," and "gangbang" are highlighted for identifying sexist content. Task 2, which involves various categories like "NO," "DIRECT," "REPORTED," and "JUDGEMENTAL," exhibits a diverse range of most attentive tokens. For instance, in the "NO" category, tokens like "pun," "wall," and "question" stand out, while "DIRECT" category tokens include "where," "GPS," and "tomorrow." Additionally, tokens such as "wenr," "school," and "saben" are emphasized in the "REPORTED" and "JUDGEMENTAL" categories, respectively. This comprehensive analysis sheds light on the model's attention mechanism and its discernment of different types of content within each task.

7. Conclusion

The EXIST challenge is designed to promote research on automated sexism detection and modeling in online environments, with a particular focus on Twitter. In this paper, we performed extensive research and conducted thorough experiments to achieve this objective, employing advanced techniques in natural language understanding and machine learning. Specifically, we enhanced existing machine learning models by incorporating an attention layer and a CLS token, which emphasize the words that contribute to the context of sexism. Our findings demonstrate the effectiveness of our approach and models on both the training and validation datasets.

However, as the training is done on Spanish and English language the model might not be proficient at identifying sexist content in other languages. This can lead to misclassification of sexist Tweets.

Acknowledgements

This project has been sponsored by Penta Global Limited Bangladesh. We would like to express our deepest gratitude to Penta Global for for their financial support.

References

[1] J. B. Walther, Social media and online hate, Current Opinion in Psychology 45 (2022) 101298.

[2] A. Elbarazi, How social media affects people's ideas on sexist behaviours and gender-based violence (2023). doi:10.19080/GJIDD.2023.12.555838.

[3] Z. Al-Makhadmeh, A. Tolba, Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach, Computing 102 (2020) 501–522.

[4] L. Plaza, J. Carrillo-de-Albornoz, V. Ruiz, A. Maeso, B. Chulvi, P. Rosso, E. Amigó, J. Gonzalo, R. Morante, D. Spina, Overview of EXIST 2024 – Learning with Disagreement for Sexism Identification and Characterization in Social Networks and Memes, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), 2024.

[5] L. Plaza, J. Carrillo-de-Albornoz, V. Ruiz, A. Maeso, B. Chulvi, P. Rosso, E. Amigó, J. Gonzalo, R. Morante, D. Spina, Overview of EXIST 2024 – Learning with Disagreement for Sexism Identification and Characterization in Social Networks and Memes (Extended Overview), in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum, 2024.

[6] S. M. H. Dadgar, M. S. Araghi, M. M. Farahani, A novel text mining approach based on tf-idf and support vector machine for news classification, in: 2016 IEEE International Conference on Engineering and Technology (ICETECH), IEEE, 2016, pp. 112–116.

[7] G. Ozogur, M. A. Erturk, Z. Gurkas Aydin, M. A. Aydin, Android malware detection in bytecode level using tf-idf and xgboost, The Computer Journal 66 (2023) 2317–2328.

[8] V. Sundaram, S. Ahmed, S. A. Muqtadeer, R. R. Reddy, Emotion analysis in text using tf-idf, in: 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2021, pp. 292–297.

[9] J. Nowak, A. Taspinar, R. Scherer, Lstm recurrent neural networks for short text and sentiment classification, in: Artificial Intelligence and Soft Computing: 16th International Conference, ICAISC 2017, Zakopane, Poland, June 11-15, 2017, Proceedings, Part II 16, Springer, 2017, pp. 553–562.

[10] X. Bai, Text classification based on lstm and attention, in: 2018 Thirteenth International Conference on Digital Information Management (ICDIM), IEEE, 2018, pp. 29–32.

[11] X. Ou, H. Li, Ynu@ dravidian-codemix-fire2020: Xlm-roberta for multi-language sentiment analysis., in: FIRE (Working Notes), 2020, pp. 560–565.

[12] L. Khan, A. Amjad, N. Ashraf, H.-T. Chang, Multi-class sentiment analysis of urdu text using multilingual bert, Scientific Reports 12 (2022) 5436.

[13] M.-I. Limaylla-Lunarejo, N. Condori-Fernandez, M. R. Luaces, Requirements classification using fasttext and beto in spanish documents, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2023, pp. 159–176.

[14] M. Fahim, M. S. Shahriar, M. R. Amin, Hatexplain space model: Fusing robustness with explainability in hate speech analysis (????).

[15] M. Fahim, Aambela at blp-2023 task 2: Enhancing banglabert performance for bangla sentiment analysis task with in task pretraining and adversarial weight perturbation, in: Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), 2023, pp. 317–323.

[16] M. Fahim, A. A. Ali, M. A. Amin, A. M. Rahman, Edal: Entropy based dynamic attention loss for hatespeech classification, in: Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation, 2023, pp. 775–785.

[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[18] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, arXiv preprint arXiv:1911.02116 (2019).

[19] J. Libovickỳ, R. Rosa, A. Fraser, How language-neutral is multilingual bert?, arXiv preprint arXiv:1911.03310 (2019).

[20] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: PML4DC at ICLR 2020, 2020.

[21] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, O. Reblitz-Richardson, Captum: A unified and generic model interpretability library for pytorch, CoRR abs/2009.07896 (2020). URL: https://arxiv.org/abs/2009.07896. arXiv:2009.07896.