Sexism Identification in Social Networks with Generation-based Language Models

Notebook for the EXIST Lab at CLEF 2024

Le Minh Quan[1,2,*], Dang Van Thin[1,2]

[1]University of Information Technology-VNUHCM, Quarter 6, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

This paper demonstrate our participation in the EXIST (sEXism Identification in Social neTworks) at CLEF 2024. We participate in 3 task: sexism identification (Task 1), sexism intention (Task 2), and sexism categorization (Task 3). We proposed an ensemble architecture using Large Language Models (LLMs) to tackle these tasks. Our approach aimed to emulate the human annotation process and gain more insights into fine-tuning LLMs for classification tasks. Our best performance model achieved 2nd on task 1, 1st on task 2 and 1st on task 3 with hard label result. Achieving 0.7826, 0.5677 and 0.6004 on F1 score respectively for each task.

Keywords

Llama 2, Large Language Model, LoRA, Fine-tuning LLM, Prompting Engineering, Social media, CLEF 2024, EXIST 2024

1. Introduction

Sexism is prejudice, stereotyping or discrimination base on one's sex or gender, typically against women and girls. This inequality and discrimination against women remains a pervasive issue in modern society and manifesting in online spaces. Sexism affects women in many facets of their lives, including domestic and parenting roles, career opportunities, body image, and life expectations. Moreover, online sexism can influence social media users, particularly teenagers, to develop sexist attitudes or view them as normal and acceptable. The growth of social media platforms like Twitter and Facebook has led to a significant rise in online sexism. This makes the need for automatic tools to identify online sexism even more critical.

Identifying sexism on social media is a challenging problem. Sexist messages can be overtly offensive and hateful, but they can also be subtle, disguised as humor or friendly posts. To address this problem, EXIST (sEXism Identification in Social neTworks) was established. EXIST 2024 at CLEF 2024 is the fourth edition of the EXIST challenge that focus at combating sexism on social media. EXIST aims to capture sexism in a broad sense, from explicit misogyny to other subtle expressions that involve implicit sexist behaviours [1, 2]. EXIST 2024 provides tasks to detect sexism behaviours and discourses, identify the intention of the author behind a sexist social media post and categorize the forms of sexism. These tasks are divided into tweet classification and meme classification, focusing on textual messages and image content (particularly memes) on Twitter. Identifying sexism content can also be conflicting due to differences in perspectives among annotators. To account for this problem, EXIST 2024 adopt the Learning With Disagreement (LeWiDi) [3] paradigm for both the development of the dataset and the evaluation of the systems. In the LeWiDi paradigm, models are trained with conflicting or diverse annotations instead of relying on a single "correct" label per sample. This results in two types of output: hard outputs and soft outputs.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

[*]Corresponding author.

20520709@gm.uit.edu.vn (L. M. Quan); thindv@uit.edu.vn (D. V. Thin)
https://nlp.uit.edu.vn/ (D. V. Thin)
0000-0001-8340-1405 (D. V. Thin)

In this paper, we employ Large Language Models to address the first three tasks of EXIST 2024. Below is a overview of the proposed tasks:

- **Task 1** is a binary classification task. The objective of this task is to decide whether or not a given tweet contains sexist expressions or behaviours.
- **Task 2** is a hierarchical multi-class classification task. For the tweets that have been predicted as sexist, the objective of this task is to classify each tweet according to the intention of the person who wrote it.
- **Task 3** is a hierarchical multi-label classification task. For the tweets that have been predicted as sexist, the objective of this task is to categorize them according to the types of sexism.

This paper presents our approach for the first three tasks of the EXIST 2024 shared task. In section 4, we introduce our proposed architecture and the models employed for experiments. Section 5 details the experimental setup, including the evaluation metrics used and the system setting. In section 6, we present and discuss the experiment result from both the development and evaluation phases. Finally, our conclusion based on the result will be in section 7.

2. Related Work

The scientific community has established numerous academic events and shared tasks about Sexism and Hate Speech overall. **HatEval 2019** focused on detection of hate speech against immigrants and women in Twitter [4]. **Homo-Mex 2024** focus on Hate Speech Detection Towards the Mexican Spanish Speaking LGBT+ Population [5]. **EDOS 2023** address the the limitations of Binary detection in sexism detection, and emphasizing the need to provide clear explanations for why something is sexist [6]. **DETESTS-Dis 2024** at IberLEF 2024 follow Learning with Disagreement paradigm to detect and classify racial stereotypes in Spanish social media [7]. These efforts show that sexism and hate speech detection still remain an active and challenging area of research.

Previous EXIST shared task (EXIST 2023) have attracted many research teams to participate. Participants have achieved good results and provided insightful research. Below are some works that achieved top results in the EXIST 2023 shared task.

- **Kelkar et al.** [8] evaluated multiple different approach, ranging from classic classification methods like Multinomial Naive Bayes and Linear Support Vector Classifiers to deep learning methods like Multi-Layer Perceptrons, XGBoost, and LSTMs with attention. They also explored multiple embedding methods. Their research concluded that training models on both languages yielded better results compared to training them on separate languages.
- **Erbani et al.** [9] used three separate BERT models to fine-tune task 1, 2, and 3. After fine-tuning, the models were "frozen" and then connected together. Fully connected layers are added at the beginning of the models. This allows the models to share their "different views" of the data from their individual training and thus achieve better performance than using single models.
- **Paula et al.** [10] employed a combination of two BERT models: Multilingual BERT (mBERT) and Cross-lingual Language Model RoBERTa (XLM-RoBERTa). Each model generated its own prediction. The final prediction label was then chosen based on the prediction probability. Their experiments demonstrated that this ensemble approach significantly improved model performance. This research highlights the potential of utilizing ensemble architectures with even larger base models for tackling text classification problems.
- **Tian et al.** [11] introduced a novel cascade architecture using two large language models (LLMs) based on the GPT architecture: GPT-NeoX and BERTIN-GPT-J-6B. The first model is fine-tuned on the competition data, this model handles simpler classifications. The second model is sequentially fine-tuned on various “Hate speech” data and then fine-tuned on the competition data, this larger model tackles more complex samples. A confidence checker system identifies data points that are challenging for the smaller model and send them to the larger model. This architecture

helps the model save computational costs, increase classification speed while still having higher performance than conventional ensemble models.

- **Vallecillo-Rodríguez et al.** [12] explored two methods to improve model. The first method is tested on transformer models including mDeBERTa and XML-RoBERTa, with data augmentation by repeating the tweet six times corresponding to six annotator that labeling each tweet sample. The second method studies how to integrate annotator information into the training process. The research team applied the multi-modal architecture called "Transformer With Tabular". Text feature (tweets) are combined with the encoded annotator's metadata to create a special tensor. This tensor will go through a classification model to output the final prediction. Their findings revealed that while annotator information did not significantly impact performance on Task 1 (likely due to its binary nature), it showed promise for Tasks 2 and 3, which involved more complex classifications.

The approaches shown above span from traditional methods such as support vector machine (SVM), multi-layer perceptron (MLP) to Deep Learning architecture like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. Transformer-base architecture like BERT and XLM also widely used. Notably, all these proposed methods rely on text classification. Meanwhile, generative models are growing significantly in both computer vision (e.g., Diffusion Models) and natural language processing (e.g., Large Language Models). In this paper, we will not only employ Medium-sized Language model like XLM-RoBERTa for classification tasks, but also leverage the power of Large Language Model like mT5 and Llama 2.

3. Dataset overview

The dataset is provided by the EXIST shared task, containing Spanish and English tweets from Twitter. Dataset structure following Learning with disagreement paradigm (LeWiDi) [3], each Tweet sample is annotated by six annotators with diverse socio-demographic characteristics. Consequently, instead of a single gold label, the labels consist of six labels annotated by each annotator. The information of each annotator is provided, including gender, age, ethnicity, study level and country of origin. Table 1 shows the statistics for the tweet data, combining both English and Spanish text.

Table 1
Tweets Dataset statistic

Information	Training	Development	Test
N.o samples	6920	1038	2076
Max length	795	586	825
Min length	29	41	38
Average length	176	180	176
Number of tokens	1168350	177972	347100
Number of vocabulary	53472	12045	20642

Table 2 shows the statistics for the labels of each task, using the official gold labels provided by the shared task. Task 1 categorizes tweets as sexist or not sexist *(yes, no)*. Task 2 classify the intention of the author who wrote the sexist tweet *(direct, reported, judgemental)*. Task 3 categorize the types of a sexist tweet, which can have one or more label *(ideological inequality, sexual violent, objectification, stereotyping dominance, misogyny non-sexual violent)*. Task 2 and Task 3 follow a two-level hierarchical structure, as illustrated in Figure 1:

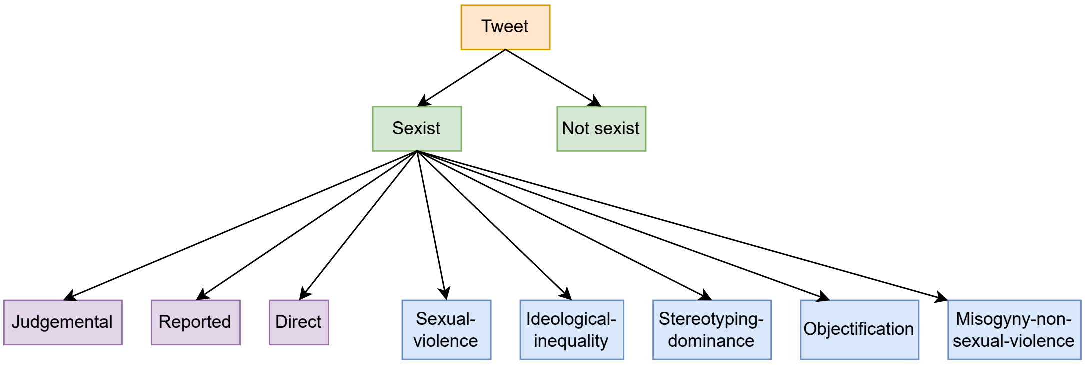

Figure 1: Hierarchical Sexism Classification.

Table 2
hard labels statistics

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

4. Approach

4.1. Architecture

Base architecture: Our architecture, illustrated in Figure 2, consists of six independent models, referred to as Component Models. These models share the same model type and training setting. This ensemble approach emulates the human annotation process, where each model represents a different annotator with potentially varying biases. Our proposed framework comprises five main components:

1. **Preparing dataset**: Each data sample includes metadata of six different annotator. We split the dataset by annotator, resulting in six separate dataset. Each dataset contains the same tweets but with metadata specific to the corresponding annotator.
2. **Data processing**: The sub-samples undergo pre-processing. For LLM models, prompt engineering is applied.
3. **Fine-tuning the Component Models**: Each Component Model undergoes a separate fine-tuning process using its own data picked from one of the six separate datasets. the component models have to have the same type, whether it is XLM-RoBERTa or mT5 or Llama 2, and have the same system setting.
4. **Post processing**: The prediction outputs from the six Component Models are collected and post-processed to the required submission format
5. **Final output**: The Component Predictions are converted into two type of outputs, soft label and hard label:
    - **Soft labels**: To generate soft labels, we calculate probability distribution for each possible classes. In tasks 1 and 2, the sum of these probabilities must be 1.0. For task 3, the sum of probability does not necessarily need to be 1.0 because of the multi-label structure.

- **Hard labels**: To generate the hard labels from the predicted outputs, we follow the probabilistic threshold used in the official gold label. For tasks 1 (mono-label), the class annotated by more than 3 annotators is selected. For tasks 2 (mono-label), the class annotated by more than 2 annotators is selected. Finally, for tasks 3 (multi-label), the class annotated by more than 1 annotator are selected. In cases where no class meets the threshold, a random class is chosen.

Integrating Hierarchical architecture: Task 2 and Task 3 follow hierarchical structure. To address these tasks, we modified our architecture such that Component Models for Tasks 2 and 3 only make predictions for sub-samples classified as sexist. We rely on the Predictions make by Component Models of Task 1 to get the classified labels *(YES, NO)*. Our base architecture incorporates a hierarchical structure, as shown in Figure 3

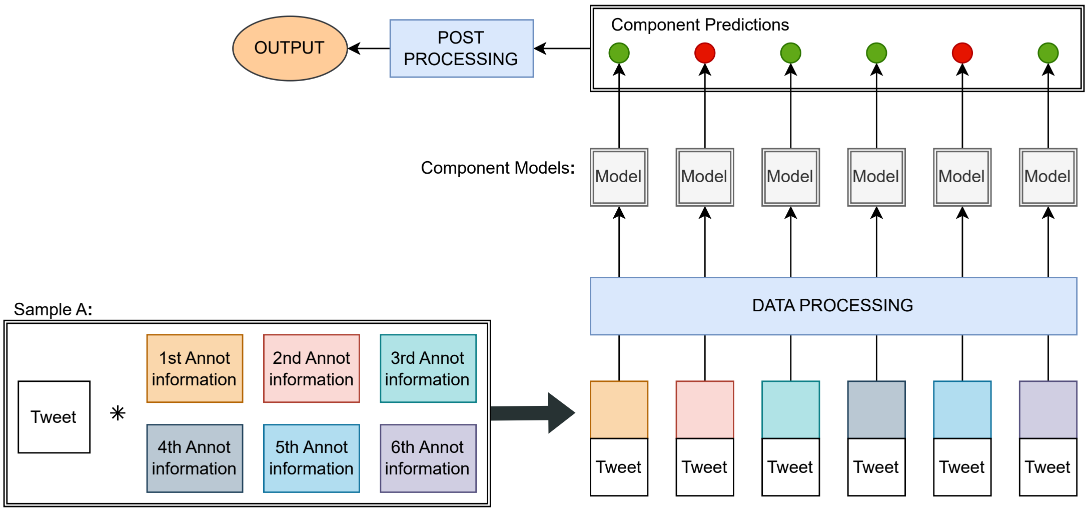

Figure 2: Overall Ensemble architecture proposed for EXIST 2024 shared task.

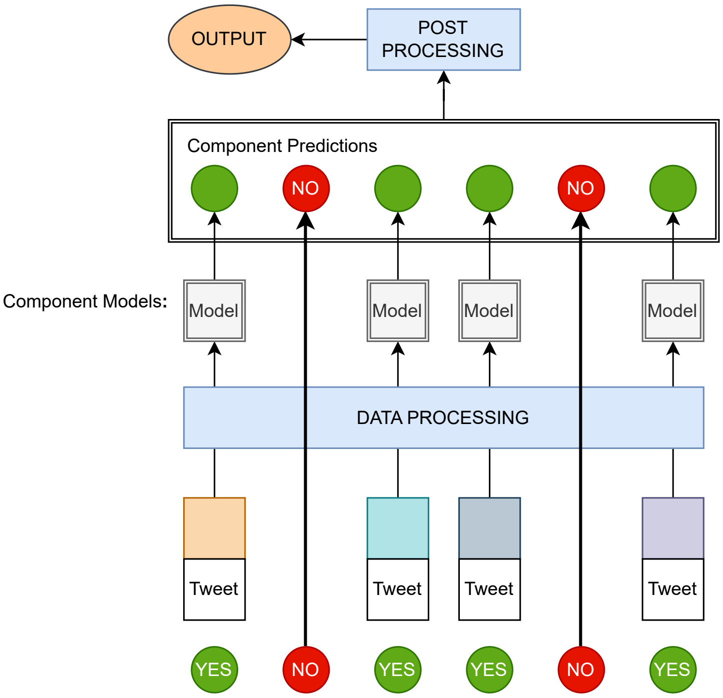

Figure 3: Ensemble architecture with Hierarchical for Task 2 and Task 3 of EXIST 2024 shared task.

4.2. Model Overview

In this paper, we leverage two types of language models: large language models (LLMs) with Multilingual T5 (mT5) and Llama 2 and a smaller transformer-base language model XLM-RoBERTa.

XLM-RoBERTa [13]: XLM-RoBERTa (XLM-R) is a Multilingual language model base on RoBERTa. It leverages the Transformer architecture and is trained using a multilingual masked language modeling objective. XLM-R has been shown to outperform multilingual BERT (mBERT) on various cross-lingual benchmarks. Our approach using XLM-RoBERTa is described below:

- **Pre-processing**: For XLM-RoBERTa, we apply multiple pre-processing methods to the tweet dataset as follows.
    1. **Link conversion**: We use regex to match URLs from tweet and convert them to "URL" token.
    2. **Mention conversion**: We convert user mentions (@username) in tweets to "USER" tokens.
    3. **Hashtag conversion**: We convert hashtags into separate words (For example, "#DeepLearning" will be converted to "Deep Learning").
    4. **label conversion**: For tasks 1 and 2, labels are converted to numerical representations. For multi-label task 3, labels are converted to binary representations.
- **Integrating metadata**: To integrate annotator metadata to the input, we add a [SEP] token to separate each metadata and the tweet. For example:
    - **Original data**: **Tweet**: "Lo sentimos, el meme aún está en construcción."; **Annotator's information**: *gender*: female, *age*: 23-45, *ethicities*: White or Caucasian, *education level*: Bachelor's degree, *country*: Spain.
    - **Processed data**: "Lo sentimos, el meme aún está en construcción.[SEP]female[SEP]23-45[SEP]White or Caucasian[SEP]Bachelor's degree[SEP]Spain"
- **Fine-tuning**: We fine-tune the pre-trained XLM-RoBERTa using the Trainer API from HuggingFace's library. We fine-tune each task separately with different parameter setting.
- **Post Processing**: Post-processing involves converting the output label to the submission format. This step transforms the numerical or binary label output by models to original natural language format.

Multilingual T5 [14]: Multilingual T5 (mT5) is a LLMs and a multilingual variant of T5. It was pre-trained on a new Common Crawl dataset containing 101 languages. T5 uses a basic encoder-decoder Transformer architecture [15] and pre-trained on a masked language modeling "span-corruption" objective, where the entire spans of tokens are selected for corruption at once. T5 excel at multiple task such as question answering, summarization, translation. Our approach using mT5 is described below:

- **Pre-processing**: Our experiments show that for mT5 in this work, performing better with tweet in their original format. This mean keeping elements like user mentions, links, and emojis rather than converting them to specific tokens or removing them. We only convert hashtags to separate words.
- **Prompting**: We format the input by adding a task prefix before the tweet and the section indicator before the annotator's metadata. For the task prefix, we use short phrases such as "Classify" or "Multi-label Classify". To indicate a section containing the annotator's information, we use phrases like "Context" or "Information". Here is the example for prompts used for each task:
    1. **Prompt:** *Classify:* Lo sentimos, el meme aún está en construcción *Information:* male, 46+, White or Caucasian, Master's degree **Response:** YES
    2. **Prompt:** *Multiclass Classify:* Lo sentimos, el meme aún está en construcción *Context:* male, 46+, White or Caucasian, Master's degree **Response:** JUDGEMENTAL

3. **Prompt:** *Classify sexism types of the following tweet:* Lo sentimos, el meme aún está en construcción *Information of annotator:* male, 46+, White or Caucasian, Master's degree **Response:** OBJECTIFICATION, SEXUAL-VIOLENCE

- **Fine-tuning**: We leverage the Hugging Face library's Trainer API to fine-tune the pre-trained mT5 model for each task separately. Each task utilizes different parameter settings to optimize performance.

Table 3
Llama 2's Prompt engineering for Task 1.

Prompt:	[INST] Imagine you are a person with the following characteristics: **female**, **23-45** years old, **White or Caucasian** ethnicity, **Bachelor's degree**, and residing in **Spain**. Now classify the sentiment of a tweet: "**Lo sentimos, el meme aún está en construcción**" ## If the tweet contains any form of sexism or describes situations involving discrimination against women, classify it as "YES". ## If the tweet does not exhibit prejudice, underestimate, or discriminate against women, classify it as "NO". Answer only YES or NO. Answer: [/INST]
Response:	YES

Table 4
Llama 2's Prompt engineering for Task 2.

Prompt:	[INST] Imagine you are a person with the following characteristics: **female**, **23-45** years old, **White or Caucasian** ethnicity, **Bachelor's degree**, and residing in **Spain**. Now classify the sentiment of a tweet: "**Lo sentimos, el meme aún está en construcción**" ## If the intention of the tweet is to write a message that is sexist by itself, classify as "DIRECT" ## If the intention of the tweet is to report or describe a sexist situation or event suffered by a woman or women in first or third person, classify as "REPORTED". ## If the intention of the tweet is to be judgemental and the tweet describes sexist situations or behaviors with the aim of condemning them, classify as "JUDGEMENTAL". Answer only DIRECT, REPORTED, or JUDGEMENTAL. Answer: [/INST]
Response:	REPORTED

LLaMA 2 [16]: Llama 2 is a large language model developed by Meta AI and is the successor to their original Llama model. Llama 2 uses the standard transformer architecture and applies pre-normalization using RMSNorm. The model employs the SwiGLU activation function, grouped-query attention (GQA), and rotary positional embeddings. Both the context length and the pre-training corpus size have been increased compared to the previous model. Unlike other models in this paper, Llama 2 was primarily trained on English data. Llama 2 surpasses its predecessor in various tasks, including Reasoning, Coding, and Knowledge. Our approach using Llama 2 is described below:

- **Pre-processing**: Like mT5, Llama 2 performs better with tweet in their original format. Therefore, we only convert hashtags to separate words.
- **Prompting**: Our observation shows that providing more information, such as the label's description to the prompt, leads to better model performance. Additionally, having distinct separators between each part of the prompt helps to clarify the instructions and makes them easier for the model to understand. A special token [INST] is utilized to separate the input prompt and answer segments. Prompts used for each task are shown in Table 3, 4, 5.
- **Fine-tuning**: We fine-tune the pre-trained Llama 2 with the LoRA method from HuggingFace's library. Each task are fine-tuned separately, with different parameter settings.

Table 5
Llama 2's Prompt engineering for Task 3.

Prompt:	[INST] Imagine you are a person with the following characteristics: **female**, **23-45** years old, **White or Caucasian** ethnicity, **Bachelor's degree**, and residing in **Spain**. Now classify the sentiment of a tweet: "**Lo sentimos, el meme aún está en construcción**" ## If the tweet rejects equality between men and women, classify as "IDEOLOGICAL-INEQUALITY". ## If the tweet implies that men are superior to women, classify as "STEREOTYPING-DOMINANCE" ## If the tweet objectifies women, classify it as "OBJECTIFICATION" ## If the tweet contains sexual suggestions, or harassment of a sexual nature, classify as "SEXUAL-VIOLENCE". ## If the tweet expresses hatred and violence towards women without being sexual in nature, classify as "MISOGYNY-NON-SEXUAL-VIOLENCE". ## Answer only these category IDEOLOGICAL-INEQUALITY, STEREOTYPING-DOMINANCE, OBJECTIFICATION, SEXUAL-VIOLENCE, MISOGYNY-NON-SEXUAL-VIOLENCE. You can choose more than one category if applicable. Answer: [/INST]
Response:	IDEOLOGICAL-INEQUALITY, STEREOTYPING-DOMINANCE

- **Low-rank Adaptation**: To fine-tune Llama 2, we used Parameter-Efficient Fine-Tuning (PEFT) method called LoRA. LoRA (Low-Rank Adaptation for Large Language Models) is a popular technique to fine-tune Large pre-trained models such as LLMs and diffusion models. LoRA allows us to train some dense layers in a neural network indirectly by optimizing rank decomposition matrices of the dense layers change during adaptation instead, while keeping the pre-trained weights frozen [17]. In short, LoRa reduce the number of trainable parameters, making the training process faster and less computational cost, while maintaining strong performance on downstream tasks.

5. Experimental Setup

5.1. Evaluation Metric

We evaluate our model's performance using a combination of metrics for all tasks and all types of evaluation (soft label and hard label):

- **Hard Label Evaluation**:
  - **ICM (Information Contrast Measure)**: ICM is a similarity function that generalizes Pointwise Mutual Information (PMI), and can be used to evaluate system outputs in classification problems by computing their similarity to the ground truth categories [18].
  - **F1-score**: F1-score measure of the harmonic mean of precision and recall.
- **Soft Label Evaluation**:
  - **ICM Soft**: A modified ICM metric that accepts both soft system outputs and soft ground truth assignments for soft label evaluation.
  - **Cross Entropy**: Cross Entropy measures the difference between the true label distribution and the predicted probability distribution. Cross Entropy only applicable for Tasks 1 and 2.

5.2. System Setting

We use the PyTorch framework and HuggingFace's Transformers library [19] for our system. We conducted fine-tuning on various Language models and their variants for each task:

- **Llama 2**
  - **Model Variants**: We use Llama 2 7B Chat model. This is the fastest and lightest model in the Llama 2 family and has been fine-tuned specifically for dialogue.
  - **Training setting:** We used a learning rate of 2e-4 and a batch size of 4. We used the Parameter-Efficient Fine-Tuning (PEFT) method LoRA and the AdamW optimizer [20]. We only fine-tune task 1 for 1 epoch. Task 2 and task 3 were fine-tuned for 5 epochs.
  - **LoRA setting:** For Causal Language Modeling, we configured LoRA with an attention dimension "r" of 8, an alpha parameter "LoRA_alpha" of 16, and a dropout probability "lora_dropout" of 0.05. For target modules, all trainable modules of Llama 2 were included: gate_proj, up_proj, down_proj, q_proj, k_proj, v_proj, and o_proj.With LoRA, the required training time and resources are significantly reduced. The number of training parameters is reduced from approximately 7 billion to approximately 20 million.
  - **Processing unit:** A100 80G GPU
- **XLM RoBERTa**
  - **Model Variants**: We conduct experiments on two XLM-RoBERTa model sizes: Base and Large
  - **Training setting:** We used a learning rate of 2e-5 and a batch size of 8. For the optimizer, we used AdamW. For regularization, we apply Weight Decay (L2 regularization) of 0.01. We set Warm-up steps to 100. This allows the model to slowly adjust to the data and mitigate the risk of divergence in the initial stages. We fine-tuned the model for 10 epochs for Task 1, and 20 epochs for Task 2 and Task 3.
  - **Processing unit:** A100 80G GPU for XLM-R Large, P100 16G GPU for XLM-R Base
- **Multilingual T5**
  - **Model Variants**: We conduct experiment on three mT5 model sizes: Small, Base and Large
  - **Training setting:** We used a learning rate of 3e-4, a batch size of 16, and the AdamW optimizer for fine-tuning. We fine-tuned all three Tasks for 15 epochs.
  - **Processing unit:** A100 80G GPU for mT5 Large, P100 16G GPU for mT5 Base and mT5 Small.

6. Result and Discussion

6.1. Development Phase

This section presents the model's performance on the development dataset. We compare the performance of different models and variants. Table 6 details our results on hard label, while Table 7 details our results on soft label. To ensure a fair comparison of Task 2 and Task 3 results, we used the true labels from the dataset for the first hierarchical level instead of predictions from Task 1. Therefore, these results are only used to compare model performance and should not be considered the actual performance on data.

Table 6
Hard label results on development set.

Model	Task 1			Task 2			Task 3		
	ICM	ICM Norm	F1-score	ICM	ICM Norm	F1-score	ICM	ICM Norm	F1-score
XLM-R base	0.4402	0.7202	0.8134	0.7396	0.7312	0.6650	0.9214	0.7052	0.7042
XLM-R large	0.5095	0.7549	0.8365	0.8041	0.7514	0.6902	1.0068	**0.7242**	**0.7235**
mT5 small	0.1686	0.5843	0.7222	0.4869	0.6522	0.5313	0.7669	0.6708	0.6679
mT5 base	0.2038	0.6019	0.7339	0.5403	0.6689	0.5493	0.0893	0.5199	0.5066
mT5 large	0.4314	0.7158	0.8098	0.6270	0.6960	0.5897	0.9017	0.7008	0.6950
Llama 2	**0.6235**	**0.8119**	**0.8746**	**0.9583**	**0.7996**	**0.7471**	**0.9602**	0.7138	0.7116

Table 7
Soft label evaluating results.

Model	Task 1			Task 2			Task 3	
	ICM Soft	ICM Soft Norm	Cross Entropy	ICM Soft	ICM Soft Norm	Cross Entropy	ICM Soft	ICM Soft Norm
XLM-R base	0.6335	0.6023	1.4940	0.4352	0.5349	1.9023	0.2396	0.5127
XLM-R large	0.9706	0.6568	1.2474	**0.7208**	**0.5578**	1.9068	**0.5924**	**0.5313**
mT5 small	-0.1815	0.4706	1.1619	-0.0628	0.4949	1.8849	-0.5319	0.4718
mT5 base	-0.0013	0.4997	**1.1495**	0.2413	0.5193	1.8241	-3.2413	0.3282
mT5 large	0.5845	0.5944	1.1795	0.4844	0.5388	**1.7933**	0.0349	0.5018
Llama 2	**0.9980**	**0.6612**	1.5768	0.6728	0.5539	1.9209	0.3020	0.5160

On hard label evaluation, Llama 2 achieves the best results for both Task 1 (ICM Norm: 0.8119, F1-score: 0.8746) and Task 2 (ICM Norm: 0.7996, F1-score: 0.7471). This suggests its strong text understanding capabilities effectively translate to classification tasks. For Task 3, Llama 2 performed on par with XLM-R large, with scores differing by only about 1% on both ICM Norm and F1-score. mT5 models consistently underperformed across all three tasks. Notably, even mT5 large's results were significantly lower than other models. This suggests that mT5 might not be well-suited for these tasks, possibly due to the complexity of the input sequences, which combine tweet and annotator information.

Llama 2 achieved the best results on Task 1 for soft labels. However, XLM-R large outperformed all models on Tasks 2 and 3. Llama-2's high performance on hard labels but not on soft labels suggests it might not be learning annotator's information or, more likely, is ignoring it due to the prompt construction. XLM-R, on the other hand, is not reliant on prompt engineering and likely processes the entire input, leading to more diverse soft outputs. Notably, all three models achieved low ICM Soft Norm scores ranging from 40% to 66%. This indicates that capturing the complexity of the LeWiDi paradigm distribution might be a challenging problem.

6.2. Evaluation Phase

Task 1: As shown in Table 8, in the Hard-hard evaluation method, Llama 2 achieved the best result (2th place) with an ICM Norm of 0.7994 and an F1-score of 0.7826 for the positive class. XLM-R large came in 7th place with an ICM Norm of 0.7898, and mT5 large achieved the lowest with an ICM Norm of 0.6952. In the Soft-soft evaluation method. XLM-R large achieved the best result, ranking 4th place with an ICM Soft Norm of 0.6490.

Table 8
Submission results on Task 1

Model	Hard - hard				Soft-soft			
	ICM	ICM Norm	F1-score (YES)	Ranking	ICM Soft	ICM Soft Norm	Cross Entropy	Ranking
XLM-R large	0.5766	0.7898	0.7823	7	**0.9291**	**0.6490**	**1.2637**	**4**
mT5 large	0.3884	0.6952	0.7292	40	0.4594	0.5737	1.2164	20
Llama 2	**0.5957**	**0.7994**	**0.7826**	**2**	0.8316	0.6333	1.6727	7

Task 2: Table 9 showcases our model performance on two evaluation methods. In the Hard-hard evaluation, our method using Llama 2 achieves 1st rank with ICM Norm at 0.6320 and F1-score at 0.5677. XLM-R large at 6th rank with ICM Norm at 0.5926, and mT5 large falls short with ICM Norm at 0.4555. While Llama 2 excelled in the Hard-hard evaluation, its performance faltered in the Soft-soft evaluation. XLM-R large emerged as the leader, ranking 7th with an ICM Soft Norm at 0.3513.

Table 9
Submission results on Task 2

Model	Hard - hard				Soft-soft			
	ICM	ICM Norm	F1-score	Ranking	ICM Soft	ICM Soft Norm	Cross Entropy	Ranking
XLM-R large	0.2847	0.5926	0.5289	6	**-1.8462**	**0.3513**	**2.4123**	**7**
mT5 large	-0.1368	0.4555	0.4182	30	-2.0149	0.3377	2.3892	10
Llama 2	**0.4059**	**0.6320**	**0.5677**	**1**	-2.9080	0.2657	2.7595	18

Task 3: Our results are presented in Table 10. In the Hard-hard evaluation, XLM-R large comes in

at a close second to 1st place of Llama 2, with ICM Norms of 0.5822 and 0.5862, respectively. XLM-R maintains its lead in the Soft-soft evaluation, ranking 7th with an ICM Soft Norm of 0.3143.

Table 10
Submission results on Task 3

Model	Hard - hard				Soft-soft		
	ICM	ICM Norm	F1-score	Ranking	ICM Soft	ICM Soft Norm	Ranking
XLM-R large	0.3540	0.5822	0.6042	2	**-3.5160**	**0.3143**	**7**
mT5 large	-0.1090	0.4747	0.5286	12	-3.5438	0.3129	8
Llama 2	**0.3713**	**0.5862**	**0.6004**	**1**	-4.5913	0.2576	13

7. Conclusion

This paper describes our approach utilizing the pre-trained Large Language Models (LLMs) for the classification tasks in the EXIST 2024 Shared task at CLEF 2024. We employed ensemble architectures to generate both soft and hard predictions for all three tasks. Our experiments highlight the strong text understanding capabilities of the Large Language Model Llama 2, allowing it to tackle various classification tasks with high accuracy on hard-hard evaluation method. However, the complexities of the LeWiDi paradigm distribution, which involves understanding diverse cultural and social perspectives presented a challenge. In future work, we will delve deeper into prompting techniques, such as Chain-of-Thought Prompting to encourage LLMs to consider the diversity of human perspectives.

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

[1] 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.

[2] 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.

[3] E. Leonardelli, G. Abercrombie, D. Almanea, V. Basile, T. Fornaciari, B. Plank, V. Rieser, A. Uma, M. Poesio, Semeval-2023 task 11: Learning with disagreements (lewidi), in: The 61st Annual Meeting Of The Association For Computational Linguistics, 2023.

[4] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, M. Sanguinetti, 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, 2019, pp. 54–63.

[5] H. Gómez-Adorno, G. Bel-Enguix, G. Sierra, S.-T. Andersen, S. Ojeda-Trueba, T. Alcántara, M. Soto, C. Macias, H. Calvo, Overview of homo-mex at iberlef 2024: Homo-mex: Hate speech detection in online messages directed towards the mexican spanish speaking lgbtq+ population, Procesamiento del Lenguaje Natural 73 (2024).

[6] H. Kirk, W. Yin, B. Vidgen, P. Röttger, Semeval-2023 task 10: Explainable detection of online sexism, in: Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), 2023, pp. 2193–2210.

[7] L. Chiruzzo, S. M. Jiménez-Zafra, F. Rangel, Overview of IberLEF 2024: Natural Language Processing Challenges for Spanish and other Iberian Languages, Procesamiento del Lenguaje Natural 73 (2024).

[8] S. Ravi, S. Kelkar, A. K. Madasamy, Lstm-attention architecture for online bilingual sexism detection (2023).

[9] J. Erbani, E. Egyed-Zsigmond, D. Nurbakova, P.-E. Portier, When multiple perspectives and an optimization process lead to better performance, an automatic sexism identification on social media with pretrained transformers in a soft label context, Working Notes of CLEF (2023).

[10] A. F. M. de Paula, G. Rizzi, E. Fersini, D. Spina, Ai-upv at exist 2023–sexism characterization using large language models under the learning with disagreements regime (2023).

[11] L. Tian, N. Huang, X. Zhang, Efficient multilingual sexism detection via large language models cascades, Working Notes of CLEF (2023).

[12] M. E. Vallecillo-Rodríguez, F. del Arco, L. A. Ureña-López, M. T. Martín-Valdivia, A. Montejo-Ráez, Integrating annotator information in transformer fine-tuning for sexism detection, Working Notes of CLEF (2023).

[13] 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, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020.

[14] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mt5: A massively multilingual pre-trained text-to-text transformer, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 483–498.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[16] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv e-prints (2023) arXiv–2307.

[17] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

[18] E. Amigo, A. Delgado, Evaluating extreme hierarchical multi-label classification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 5809–5819.

[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface's transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).

[20] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).