Sexism Detection with LLM

Ali Akbarhalvaei, Ali Loloee
Master's Degree in Artificial Intelligence, University of Bologna
{ ali.akbarhalvaei, ali.loloeejahromi }@studio.unibo.it

Abstract

This assignment tackles the Explainable Detection of Online Sexism (EDOS) task by designing a binary classification framework using large language models (LLMs). Leveraging a dataset created for sexism detection, we implemented the mentioned prompting template where the LLM assumes the role of an annotator tasked with providing concise classifications ("YES" or "NO") for sexist content. Two state-of-the-art LLMs, Mistral V3 (Mistral AI, 2023) and Phi 3.5 (Microsoft, 2024) Instruct, were evaluated under zero-shot and few-shot settings. Both Phi 3.5 Mini Instruct and Mistral V3 achieved 59% accuracy in zero-shot and 50% in few-shot learning. The models often generated more than one word but they succeeded at generating a relevant response for classification problem all the time.

1 Introduction

In this assignment, we evaluated multiple LLMs, including Mistral V3, Phi 3.5 Mini Instruct, and Llama V3.1, for their performance on the Explainable Detection of Online Sexism (EDOS) task. Following the provided prompting templates, the models were tasked with classifying text as sexist or non-sexist, responding with concise "YES" or "NO" answers. Both zero-shot and few-shot paradigms were tested, with the latter incorporating two labeled examples per class.

Our experiments assessed metrics such as accuracy, fail ratio, adherence to instructions, and number of predictions for each class. Specific challenges, such as verbose responses. These findings underline the need for advanced prompt engineering and fine-tuning to improve reliability and explainability in socially sensitive tasks.

2 System description

This section outlines the pipeline we followed to address the Explainable Detection of Online Sexism (EDOS) task using large language models (LLMs). The process consists of four key stages:

1. **Model Loading and Quantization:**
   First, we loaded the model and because of the limitation of the GPU memory we used 4-bit quantization usng BitsAndBytes.

2. **Dataset Preparation and Prompting:**
   The provided dataset was preprocessed using the provided prompting template, and the labels were converted into binary values (1 for 'sexist' and 0 for 'not sexist'). For few-shot learning, 2 labeled examples per class were included in the prompts to provide context during inference.

3. **Inference (Response Generation):**
   During this step, the processed text prompts were passed to the loaded LLMs to generate responses. The models were tasked with classifying each input text into one of two categories ("sexist" or "not sexist").

4. **Post-processing and Binary Output Extraction:**
   The model consistently predicted 'YES' or 'NO' immediately after the 'ANSWER:' tag; however, it occasionally continued generating additional words after the answer. At this stage, we extracted only the initial response ('YES' or 'NO') to evaluate the LLMs' performance on the classification task.

3 Experimental setup and results

In this section, we describe the experimental setup and present the results obtained for both zero-shot and few-shot learning using Phi 3.5 Mini Instruct and Mistral V3 models. Additionally, we analyze how well each model adheres to the instructions provided in the prompting template, particularly focusing on instances where models generated

additional words beyond "YES" or "NO." We refer to this variable as **extra_words**.

For example, in the zero-shot setting, Mistral V3 generated the following response:

> **ANSWER:** YES How about this text: "W

In this case, the model successfully classified the input text as sexist ("YES"), but it failed to adhere to the instructions by adding irrelevant content. While the classification itself was correct, the response violated the task's requirements for brevity.

Both models showed the same performance in Accuracy and Fail-ratio. Fail ratio is always zero in these models. Thus, we need other measures to take into consideration. The measures are how often the models fail to follow the instruction to only produce one word and no other words, and also the tendency of the models to produce one answer more than the other class.

Model	Setting	Acc (%)	Fail Ratio (%)	extra words (%)
Phi	Zero-shot	59	0	78
Phi	Few-shot	50	0	3
Mistral	Zero-shot	59	0	1.3
Mistral	Few-shot	50	0	18.1

Table 1: Performance Metrics for Zero-shot and Few-shot Learning.

Model Setting	Yes Count	No Count
Phi Mini Zero-Shot	267	33
Phi Mini Few-Shot	654	346
Mistral V3 Zero-Shot	267	33
Mistral V3 Few-Shot	671	329

Table 2: Yes and No counts for different LLM settings.

4 Discussion

In this section, we analyze the results of our experiments and the errors.

4.1 Error Analysis

The quantitative results highlight some key differences in the performance of Phi 3.5 Mini Instruct and Mistral V3 in zero-shot and few-shot learning setups:

- **Extra Words:** In the zero-shot setup, Mistral failed to follow the instruction to produce only one word (either "YES" or "NO") just 1.3% of the time, whereas Phi failed 78% of the time, often continuing to generate additional text after providing the correct answer. However, in the few-shot setup, Phi significantly improved, failing to follow instructions only 3% of the time, while Mistral's failure rate increased to 18%. These results indicate that Mistral adhered better to instructions in the zero-shot setting, whereas Phi performed better in the few-shot setting.

- **Unbalanced Responses:** Both models exhibited a tendency to produce more "YES" answers compared to "NO" answers, as illustrated in Table 2. This imbalance could indicate an inherent bias in the models or their interpretation of the dataset. Further fine-tuning or adjustments to the prompting template may be necessary to address this behavior and achieve a more balanced response distribution.

Overall, the quantitative metrics suggest that while Phi performs better in instruction adherence, both models accuracy and fail-ratio are similar.

5 Conclusion

Based on our experiment, two models are good at answering a classification problem with fail raio of 0.

Based on these findings, the following directions could be pursued to address the observed limitations and improve performance:

- **Advanced Prompt Engineering**: Developing more robust prompts that explicitly discourage verbose responses could further enhance adherence to task-specific instructions.

- **Fine-tuning LLMs**: Fine-tuning the models on a task-specific dataset could improve their ability to align with instructions and achieve higher accuracy.

In summary, while general-purpose LLMs can effectively tackle socially sensitive tasks like sexism detection, this study highlights the importance of prompt design, model fine-tuning, and error analysis in achieving robust and reliable performance. Future work should focus on addressing these challenges to develop more accurate and explainable AI systems.

6 Links to external resources

- **EDOS Dataset**: GitHub repository for small part of the EDOS dataset
- **Hugging Face Transformers Library**: Official Hugging Face Transformers Library

References

Microsoft. 2024. Phi-3.5-mini-instruct.

Mistral AI. 2023. Mistral-7b-instruct-v0.3.