Probing zero shot VLMs for hate meme detection: Opportunities, risks and interpretations

Anonymous ACL submission

Abstract

Multimedia content on social media is rapidly evolving, with memes gaining prominence as a distinctive form. Unfortunately, some malicious users exploit memes to target individuals or vulnerable communities, making it imperative to identify and address such instances of hateful memes. Extensive research has been conducted to address this issue by developing hate meme detection models. However, a notable limitation of traditional machine/deep learning models is the requirement for quality labeled datasets for accurate detection. Recently, the research community has witnessed the emergence of several vision language models (VLMs) that have exhibited outstanding performance across various tasks. In this study, we aim to investigate the efficacy of *open-source* VLMs in handling intricate tasks such as hate meme detection in a *completely zero-shot setting*. In particular, we systematically study various prompt strategies using zero-shot capabilities of VLMs to detect hateful/harmful memes. Next we use a novel superpixel based occlusion technique to obtain better interpretations of the misclassification results. Finally we show that these misclassified data points nicely cluster into well-defined topics thus naturally identifying the vulnerabilities of the VLMs and paving the way to better fabrication of safety guardrails in future. *Warning: Contains potentially offensive memes.*

1 Introduction

Several *large* vision language models (VLMs) have recently become available to the public. These models exhibit impressive performance across various tasks, including sentiment analysis (Kheiri and Karimi, 2023), visual question answering (Lan et al., 2023), and so on. However, the existing literature falls short in addressing how VLMs perform precisely in the context of hate meme detection (Van and Wu, 2023) and particularly in a zero-shot setting. The urgency for such systems stem from the exponential growth in multi-modal content on social media platforms with malevolent individuals severely exploiting memes as a tool to target various communities and propagate hate (Gomez et al., 2020). While manual moderation is nearly impossible, traditional machine learning models can also be not extensively trained for automatic moderation due to the severe lack of labeled hateful memes datasets that are diverse in terms of language, target groups and social setting. This gap in research underscores the need to explore and evaluate the effectiveness of zero-shot VLMs for identifying and mitigating the spread of such content in memes. Note that the zero-shot setting is important here since curating labeled hateful meme datasets that are socially, culturally and target-wise diverse is extremely difficult.

In this paper, for the first time, we systematically employ various prompt strategies and input instructions to assess the 'power' of well-known *open source* VLMs, including IDEFICS (Laurençon et al., 2023), LLAVA-1.5 (Liu et al., 2023), and INSTRUCTBLIP (Dai et al., 2023) in detecting hateful memes in a fully zero-shot setting. Note that we purposefully choose open source models since they do not come with a huge monetary cost unlike their commercial counterparts. We evaluate the outputs of these models for four well-known datasets covering hateful, misogynistic, and harmful memes. The central contributions of this paper are as follows.

(i) Systematic evaluation of classification capability of VLMs: We systematically study the effect of prompt strategies that we use to query these models to understand their strengths and vulnerabilities. In total we investigate as many as **32** prompts (8 prompt variations across 4 datasets) for each model. This is unlike what is typically done in a majority of studies where the model is queried using one or two prompt variants at most thus limiting the true potential of prompt engineering. Our prompts can be broadly categorized into the following types based on the input and output patterns: input variants can comprise (a) vanilla input, (b) input along with the definition of what is hateful/misogynistic/harmful, (c) input along with OCR text, (d) input along with definition and OCR text; output variants can be (a) vanilla output, (b) output along with an explanation. We observe that prompt variants that are most successful in eliciting correct responses vary across models and datasets; nevertheless, in many of the cases *OCR text alone* or *OCR text with definition* works well.

(ii) Interpretation of misclassified results: In order to understand the reasons for the misclassifications done by a model we present a **novel** superpixel based occlusion strategy to occlude different parts of an originally mispredicted meme. We note if these occlusions result in a change in the model prediction. If they indeed do, then one can conclude that the occluded parts play an important role in the decision making process of the model. This approach allows us to interpret the failure cases of the model and pinpoints to the regions of the memes that plays a key role in confusing the model predictions. Interestingly, we also find evidences of cases where the ground-truth annotations might themselves have been wrong, per our judgement, as opposed to the model predictions.

(iii) Typology of misclassifications: The final question that we ask in the paper is whether one can systematically organise the misclassifications of the model so that constructive suggestions could be brewed from them to re-engineer the safety guardrails of the VLMs. To this purpose, we cluster the misclassified memes using multi-modal topic modeling thereby inducing a *typology* of error patterns. Interestingly, this typology seems to highly align with the different kinds of superpixel based interpretations that we obtain. This typology can be thought of as the 'Achilles heel' of a VLM against which it needs to be safeguarded in future.

Overall, our study has a far larger scope than the standard objective of identifying the best all-purpose VLMs. It strives to rather choose the best prompt variant across different models using a thorough and principled prompt engineering approach. Further it lays a foundation to identify interpretable typological categories of hateful memes that the VLMs are most vulnerable to. These induced topics can be used to improve the performance of VLMs by implementing safety guardrails without fine-tuning the models repeatedly which typically comes with a huge compute cost.

2 Related works

Hate meme detection: A growing body of research in recent years focused on hate meme detection (Gomez et al., 2020; Kirk et al., 2021; Shang et al., 2021). Several datasets and models have been developed, encompassing various dimensions, including hateful content detection (Kiela et al., 2020), misogyny detection (Fersini et al., 2022), cyberbullying detection (Maity et al., 2022), harmful meme detection (Pramanick et al., 2021a,b), and many more (Chandra et al., 2021; Lin et al., 2024) including other languages (Das and Mukherjee, 2023).

Vision language models: IDEFICS (Laurençon et al., 2023), LLAVA-1.5 (Liu et al., 2023), INSTRUCTBLIP (Dai et al., 2023), Flamingo (Alayrac et al., 2022), and OpenAI GPT-4 (AI) are popular vision language models widely used for tasks like sentiment analysis (Kheiri and Karimi, 2023), question answering (Lan et al., 2023), and chatbot creation (AlZu'bi et al., 2024), hate meme detection (Van and Wu, 2023). However, exploring hate meme detection using VLMs is limited (Van and Wu, 2023; Lin et al., 2024), particularly in the context of different prompt scenarios, different model setups and thorough interpretation of results.

3 Datasets and metrics

Datasets: This section introduces the four datasets we have utilized to explore the performance of Vision-Language Models (VLMs). These datasets cover three types of memes: hate, misogyny, or harmful content (see Table 1 for details).

Dataset	Label distribution		Total
FHM	Hateful	250	500
	Not hateful	250	
MAMI	Misogynous	500	1000
	Not misogynous	500	
HARM-P	Harmful	173	355
	Not harmful	182	
HARM-C	Harmful	124	354
	Not harmful	230	

Table 1: Label distribution for each dataset.

(1) Facebook hateful memes (FHM): The FHM dataset introduced by Facebook AI (Kiela et al., 2020) is a collection of memes designed to help researchers develop tools for identifying and removing hateful content online. The dataset contains more than 10K memes labeled hateful and not-hateful, covering various targets, including race, ethnicity, religion, gender, sexual orientation, and disability. We use a random sample of 500 memes[1] in order to test the VLMs in the zero-shot setting.

(2) Multimedia automatic misogyny identification (MAMI): The MAMI (Fersini et al., 2022) dataset was created and shared as part of SemEval-2022 Task 5. Unlike the FHM dataset, the MAMI dataset focuses on identifying misogyny in online memes. The dataset contains 11K memes, of which 1K memes are in the test set, and we conduct all our experiments considering only the test set. Each meme has a binary label – 'misogynous' or 'not misogynous' – which we use for our experiments.

(3) Harmful memes: 'Harmful' is a more general term compared to 'offensive' and 'hateful'. While an offensive or hateful meme is harmful, not all harmful memes are necessarily hateful or offensive. We utilize the **HARM-P** (Pramanick et al., 2021b) (related to US politics) and **HARM-C** (Pramanick et al., 2021a) (related to COVID-19) datasets for our experiments. Both datasets contain more than 3.5K memes. For our study, we only consider the test sets. The original labels of both datasets have three classification labels: *not harmful*, *somewhat harmful*, and *very harmful*. To maintain consistency with our binary classification experiments, we have merged *somewhat harmful* and *very harmful* into a single category labeled as *harmful*.

Metrics: As we perform binary classification tasks, we measure the models' performance using **accuracy**, **macro F1 score**, and **area under the ROC curve** metrics.

4 Models

We ran our experiments on a total of **five** different models. All models are open source large VLMs. Due to resource constraints as well as to make a fair comparison, we ran our experiments using *8-bit quantization* (Liu et al., 2021) for all the VLMs.

IDEFICS: IDEFICS (Laurençon et al., 2023) which closely follows the architecture of Flamingo, is trained on open source datasets like OBELICS and LAION. It combines two frozen uni-modal backbones which are, LLaMA as the language model and OpenClip as the vision encoder. We used instruction fine-tuned IDEFICS 9B model with the checkpoint ***HuggingFaceM4/idefics-9b-instruct*** for our experiments.

[1]Note that for this dataset the test set was removed by the authors after the competition. We have therefore used the validation set to sample our data points.

LLAVA-1.5: LLAVA-1.5 (Liu et al., 2023) is an enhanced version of LLaVA. LLaVA combines LLaMA/Vicuna as the language model and CLIP as the vision encoder. Compared to LLaVA, LLAVA-1.5 has enhanced capabilities due to the addition of an MLP vision-language connector and integration of academic task-oriented data. We have used two different LLAVA-1.5 models with 7B and 13B parameters. The checkpoints of these models are ***llava-hf/llava-1.5-7b-hf*** and ***llava-hf/llava-1.5-13b-hf***.

INSTRUCTBLIP: INSTRUCTBLIP (Dai et al., 2023) is an instruction fine-tuned model that uses the same architecture as BLIP-2 with a small but significant difference. It uses frozen Flan-T5/Vicuna as the language model and a vision transformer as the image encoder. Extending BLIP-2, INSTRUCTBLIP proposes an instruction-aware Q-Former module. As additional inputs, the model takes instruction text tokens which interacts with the query embeddings via the self-attention layer of the Q-Former. We have used two different INSTRUCTBLIP models with Vicuna 7B and Flan-T5-xl as backbone language models. The checkpoints of these models are ***Salesforce/instructblip-vicuna-7b*** and ***Salesforce/instructblip-flan-t5-xl*** respectively.

5 Prompts

This section presents the array of prompt variants employed in our work. A concise summary of representative examples for the prompt variants is provided in Appendix B, while detailed information for each variant is discussed below.

Input patterns: We run our experiments on four different input patterns, which are as follows.

Vanilla input: Following (Roy et al., 2023), we use a prompt template to instruct the model to classify the given meme into a label from a predefined `list_of_labels`. However, in our scenario, the `list_of_labels` is only restricted to binary labels. In addition, we supply two `example_outputs` (one label per line for positive and negative samples) to assist the models in generating appropriate answers. In our case, 'positive' denotes content deemed hateful, misogynistic, or harmful based on the dataset passed to the model.

(+) *Definition input*: For vanilla prompts, we assumed that VLMs are to some extent aware of the labels for classifying the input image. Here, we take a step further and add the `definition` of the labels as an additional context to the VLMs. Our intuition was similar to Roy et al. (2023), i.e., the `definition` can help the VLMs understand the classification tasks better. We picked and added one line of `definition` from the corresponding dataset for all `list_of_labels` (positive and negative in our case). We provide definitions of the labels for each dataset in Appendix A.

(+) ***OCR input***: In a meme, multi-modality, i.e., embedded text and image play very crucial role in the classification task, similar to the works (Pramanick et al., 2021a; Das and Mukherjee, 2023). We therefore add `ocr_extracted_text` in the vanilla prompt. Our intuition was that the models would further be better in understanding the contexts with this addition and would be more successful in classifying the input image meme as per the `list_of_labels`. We provide the `ocr_extracted_text` enclosed within three back-ticks for the model to easily distinguish it from other texts in the prompt.

(+) ***Definition & OCR input***: Here, we combine both `definition` and `ocr_extracted_text` with vanilla prompt and pass it as input prompt for our experiment. We use all intuitions discussed above in previous prompt variants and assume that this prompt would provide the models with deeper contexts for the classification task. Moreover, in this setup the order of the prompt text is the `definition` followed by the `ocr_extracted_text`.

Output patterns: We run our experiment on two different output patterns which are noted below.

Vanilla output: In this case, we prompt the model to generate as output only the correct class label from the `list_of_labels` corresponding to different datasets as mentioned in Table 1.

(+) ***Explanation output***: Adding to the above case of vanilla output, we prompt the model to further explain the raionale (within 30 words) based on which it made a prediction.

Thus we run a total of *eight* prompts for each dataset and for each model setup by running four input patterns × two output patterns.

6 Experimental setup

For all the models, we use a batch size of 1. We manually tune the temperature values and set them to 1.0 for the IDEFICS, LLaVA-1.5 7B and 13B models, and 0.8 for the InstructBLIP models.

Strategies		FHM			MAMI			HARM-C			HARM-P		
in	out	acc	mf1	auc	acc	mf1	auc	acc	mf1	auc	acc	mf1	auc
IDEFICS 9B													
vn	vn	53.2	48.84	53.2	50.5	34.96	50.5	62.99	53.64	54.42	50.42	49.68	50.76
def	vn	50.14	33.4	50	50	33.33	50	44.49	43.32	44.63	51.12	50.34	50.87
ocr	vn	58	57.64	58	53.2	42.58	53.2	64.31	**61.64**	**62.02**	**63.38**	**63.1**	**63.2**
def + ocr	vn	52.02	41.29	52.02	50.1	33.56	50.1	45.35	45.29	49.38	53.67	53.55	53.75
vn	ex	51.2	43.16	51.2	50.1	33.56	50.1	51.13	50.01	51.24	47.61	46.97	47.36
def	ex	50.6	34.65	50.6	50.9	38.91	50.9	35.04	28.66	47.95	50.14	46.55	50.83
ocr	ex	57.6	57.45	57.6	50.15	50.13	50.15	64.41	39.92	49.75	48.17	48.17	48.18
def + ocr	ex	49.8	38.15	49.8	49.4	36.69	49.4	51.84	43	43.94	53.39	47.22	52.64
LLaVA-1.5 13B													
vn	vn	55.95	52.27	55.83	62.3	58.09	62.3	53.95	53.76	57.13	54.93	54.32	55.26
def	vn	57.96	57.46	57.81	60.84	60.63	60.82	54.76	54.53	56.93	54.79	53.95	54.96
ocr	vn	54.8	52.59	55.1	55.22	51.38	55.08	61.61	56.88	56.78	59.57	58.62	59.36
def + ocr	vn	58.57	58.33	58.57	**67.56**	**67.55**	**67.6**	58.63	58.07	60.87	56.12	55.61	56.29
vn	ex	56.61	55.89	56.61	61.92	61.9	61.92	55.81	45.09	46.94	54.31	49.73	53.65
def	ex	50.51	36.89	50.23	62.59	62.58	62.59	42.86	40.59	53.22	50.28	42.57	51.37
ocr	ex	57.5	57.5	57.52	64.16	63.97	64.22	54.05	51.65	52.18	58	56.09	57.45
def + ocr	ex	49.7	36.46	49.42	63.03	62.21	62.8	43.55	40.74	54.41	50	41.23	51.14
LLaVA-1.5 7B													
vn	vn	50	33.33	50	50.8	35.25	50.8	64.97	39.38	50	51.27	33.89	50
def	vn	52.8	46.79	52.8	50.82	41.81	50.58	**67.35**	58.12	58.39	52.46	43.25	51.89
ocr	vn	53.31	46.32	53.38	53.4	41.17	53.4	65.25	40.25	50.4	51.27	33.89	50
def + ocr	vn	55.6	50.39	55.6	62.7	60.44	62.7	65.25	59.93	59.69	54.93	52.7	54.41
vn	ex	50.4	36.18	50.4	55.1	48.37	55.1	64.97	39.38	50	51.55	34.53	50.29
def	ex	55	53.91	55	54.7	46.12	54.7	48.02	47.28	56.65	49.86	47.52	50.43
ocr	ex	51.2	41.45	51.2	52.7	40.89	52.7	64.97	39.38	50	51.55	35.03	50.3
def + ocr	ex	**60**	**59.98**	**60**	63.6	63.48	63.6	60.45	59.03	60.27	54.08	54.07	54.08
InstructBLIP Vicuna 7B													
vn	vn	54.14	38.59	49.27	46.86	31.91	49.18	44.25	40.16	55.24	43.98	33.42	48.64
def	vn	51.12	34.55	50.41	49.74	34.44	50.29	49.12	48.65	50.84	48.63	41.05	48.1
ocr	vn	50.1	33.73	50.2	48.37	33.94	48.85	65.44	59.86	59.65	48.48	46.96	47.85
def + ocr	vn	50.21	34.87	50.01	51.49	38.19	51.49	64.13	52.84	54.33	51.49	44.12	52.87
vn	ex	48.38	38.06	49.05	50.35	35.2	50.4	46.63	41.27	41.27	44.84	44.21	44.69
def	ex	49.68	33.19	50	51.43	49.85	51.56	46.88	46.77	51.04	50.35	50.27	50.41
ocr	ex	49.12	34.41	49.75	47.39	47.34	47.56	65.42	55.19	55.91	49.5	45.17	48.7
def + ocr	ex	53.06	44.37	52.72	54.39	52.52	53.97	65.6	51.25	53.61	54.09	49.82	53.4
InstructBLIP Flan-T5 xl													
vn	vn	50.2	33.78	50.2	56.67	48.6	56.79	64.12	41.88	50.09	51.27	36.32	50.07
def	vn	50.2	35.13	50.2	59.9	56.67	59.9	64.97	39.38	50	51.27	33.89	50
ocr	vn	51.2	39.22	51.2	55.9	47.75	55.9	65.16	40.21	50.4	51.27	33.89	50
def + ocr	vn	52.6	42.33	52.6	52.1	39.59	52.1	65.54	42.52	51.18	51.98	36.68	50.63
vn	ex	60.78	60.76	60.76	55.34	50.7	56.32	60.19	41.53	47.07	49.12	46.76	53.79
def	ex	51	40.86	51	53.14	34.92	49.91	64.2	39.1	50	51.46	33.98	50
ocr	ex	60	58.17	59.72	61.13	59.39	59.87	61.7	38.16	49.43	49.33	33.03	49.33
def + ocr	ex	57.94	55.58	58.02	55.01	45.67	54.63	65.04	39.41	50	51.56	34.02	50

Table 2: *Overall results* - Accuracy, Macro-F1 and AUC-ROC score for 4 datasets and 5 models across 8 prompt variants per (model, dataset) combination. Greyed out cells signify that (model, prompt) combination is unable to classify for at-least 90% cases for the corresponding dataset. Best (model, prompt) combination per (model, dataset) combination is highlighted in light blue. Best (model, prompt) combination over each dataset is marked in yellow. **in**: Prompt input, **out**: Prompt output, **vn**: vanilla, **def**: definition, **ocr**: OCR text, **ex**: explanation.

The temperature parameter controls how random the generated output would be. However, with lower temperatures, we observed inferior performance of these models. As noted earlier, we experiment with eight different prompts on four datasets, studying them across five models. In short, we run **32** prompts per model and **160** prompts across all five models. All the models are coded in Python using the PyTorch library. We utilize 2xT4 GPUs from Kaggle, providing a total of 15GB memory on each GPU with a usage limit of 30hrs/week. Further setup details are provided in Appendix C. We present the detailed results in the following section.

7 Results

In this section, we present the results of our experiments. In Table 2 we show the results for the four datasets across the five models. Each block in the table corresponds to a particular (model, dataset)

Strategy		Models				
in	**out**	**IDEFICS 9B**	**LLAVA-1.5 13B**	**LLAVA-1.5 7B**	**I-BLIP V**	**I-BLIP F**
vn	**vn**	43.46	55.47	35.26	NA	42.2
def	**vn**	NA	57.86	45.78	NA	45.36
ocr	**vn**	52.34	53.7	41.02	40.14	42.38
def + ocr	**vn**	40.4	**62.03**	56.84	NA	40.21
vn	**ex**	40.52	55.89	41.95	NA	NA
def	**ex**	37.53	50.03	48.29	NA	NA
ocr	**ex**	49.84	59.26	39.83	NA	NA
def + ocr	**ex**	39.72	49.57	60.46	NA	NA

Table 3: *Leaderboard* - Weighted macro F1 score for each (model, prompt) combination averaged across all 4 datasets. Overall best score is underlined and highlighted. Best scores across each prompt strategy are underlined. **I-BLIP V:** INSTRUCTBLIP Vicuna 7B, **I-BLIP F:** INSTRUCTBLIP Flan-T5-xl.

combination and covers the results for eight prompt pattern combinations. Since we use the generation capability of VLMs for prediction, we observe that in some prompt variants, certain (model, prompt) combinations did not classify the input meme amongst the `list_of_labels` and diplomatically bypassed the query with an irrelevant answer. This led to a decrease in support to infer the results accurately grounded on correct labels in the dataset. In the table, we have greyed out the cases which did not generate a correct label for at least 90% of the data points. Examples of ambiguous outputs are provided in Appendix D.

Overall results: From Table 2, we observe that INSTRUCTBLIP models are not able to correctly predict the labels out of `list_of_labels` and generate ambiguous answers for quite a large number of prompt variants. Their generated output did not conform with the expected output format specified in input prompt. We also observe that IDEFICS performs best with only OCR as input. Overall, LLAVA-1.5 13B emerges to be the best model with OCR and definition as input and vanilla output. It was also the most stable in terms of the responses generated out of all the VLMs considered across different prompt variants. LLAVA-1.5 7B, worked best with explanation as output, when the input prompt was definition and OCR text.

Leaderboard : Since engineering solutions are always in the 'quest for the best', we propose a quantitative metric to organize the (model, prompt) combinations into a leaderboard. The idea is that the top combinations on this leaderboard should generalize well across the four datasets combined. For each prompt variant considered over all models, we calculate a weighted average macro F1 score depending on the number of samples in each of the datasets by the formulation: $\frac{\sum_{\mathcal{D}}(f_{\mathcal{D}})*|\mathcal{D}|}{\sum_{\mathcal{D}}|\mathcal{D}|}$. Here $f_{\mathcal{D}}$ is the macro F1 for the dataset $\mathcal{D}$. The results are shown Table 3. We did not consider those prompt variants in our calculation which did not produce results for at least 90% of the data points i.e., the grey entries in Table 2. For such cases, we mark **NA** in Table 3. Based on the above results, we conclude LLAVA-1.5 13B to be the best model with definition and OCR text as input and vanilla as output. Further, in 7 out of 8 prompt variants LLAVA-1.5 13B outperforms all other models across the datasets combined. The only variant where LLAVA-1.5 7B beats LLAVA-1.5 13B is definition and OCR text as input and explanation as output.

8 Error analysis

In the previous section we found that LLAVA-1.5 13B (with definition + OCR text as input and vanilla as output) is at the top of the leaderboard. We therefore investigate the cases of misclassification for this setting by comprehensively evaluating a total of 799 misclassified memes across considered datasets; 202 from *FHM*, 321 from *MAMI* and 276 from *HARM P+C* datasets. In particular we attempt to obtain an explanation of ***parts in the meme*** that confuses the model resulting in the mispredictions (section 8.1). In addition, we induce a ***typology of the error cases*** to systematically organise the vulnerable points of the model (section 8.2).

8.1 Occlusion based result interpretation

Using the SLIC algorithm (Süsstrunk et al., 2012) we first segment the misclassfied memes into superpixels. The algorithm automatically segments the images into ***5 - 12 superpixels*** depending on the size of the image. We control the size of each superpixel so that it is neither too small nor too big. Next the region circumscribing each of these superpixels are occluded one at a time by white patches and the model (i.e., LLAVA-1.5 13B with definition and OCR text as input and vanilla as output) is queried again for its predictions. We present a ***case-by-case manual analysis*** of the outputs obtained.

CASE 1: Original meme misclassified as positive (i.e., hateful, misogynistic or harmful corresponding to the dataset) and at least one occluded version resulted in the correct prediction (i.e., negative). Table 4 presents some representative examples for this case from each dataset. ***FHM dataset***: Majority of the memes are made up of two images stacked together. These memes put humans and animals (apes/gorilla/goat) in the same frame. Further cer-

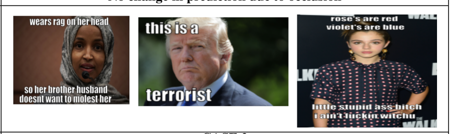
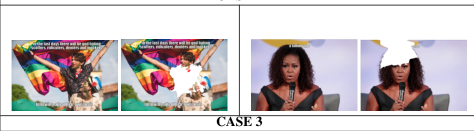
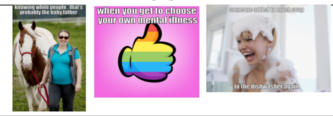
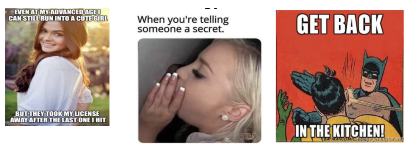
Dataset	Misclassified To	Change in prediction due to occlusion	No change in prediction due to occlusion
FHM	Hateful	 CASE 1	 CASE 2
	Not-Hateful	 CASE 3	 CASE 4
MAMI	Misogynistic	 CASE 1	 CASE 2
	Not-Misogynistic	 CASE 3	 CASE 4
HARM-C & P	Harmful	 CASE 1	 CASE 2
	Not-Harmful	 CASE 3	 CASE 4

Table 4: Occlusion based predictions. The occlusion is implemented by making a given superpixel white. **Note:** We have hidden explicit nudity in the memes wherever present using black boxes.

tain memes have embedded text containing profane words. However, in most cases the overall theme of the meme is not hateful. Occlusion resulted in correct predictions due to the removal of these confusing regions from the meme where the model was misfocusing. That said, our manual inspection indicates that some memes are indeed wrongly annotated as not hateful and the predictions of the model for the original meme are arguably correct. ***MAMI dataset***: Majority of the memes contain perturbed faces of women with weird makeups or portray men either with (i) women or with (ii) embedded text containing words like *'women', 'girlfriend', 'girl'*. Moreover, many memes are made up of multiple images stacked together. However, the overall theme of the meme is not misogynistic. When occlusion removes the perturbed faces of women or words from the embedded text, the focus of the model is no longer misdirected thus leading to correct predictions. ***HARM P+C dataset***: Here again most of the memes are composed of stacked images. Further many of these memes have long text with small font size embedded on them. Such images are even hard for human judges to label. Owing to this very complex nature of the memes, there in no regular pattern indicating why occluding certain parts of the image results in the correct prediction. This is one case where the occlusion based prediction changes are insufficient in explaining the performance gap of the models and more research is needed in the future.

CASE 2: Original meme misclassified as positive and none of the occluded versions resulted in the correct prediction (see Table 4 for some representative examples). ***FHM dataset***: Surprisingly, we find that a major portion of the memes are indeed hateful and seem to be incorrectly annotated as not hateful. Common targets include religion, gender, race and politicians. Amongst religion, *'Islam'* is mostly targeted while *'Hitler'* and *'Trump'* are the most targeted politicians. None of the occlusions resulted in a change in the predictions which further reinforces the possibility that the data might be wrongly annotated. ***MAMI dataset***: Majority of the memes pose nudity, vulgarity, feminism amongst other attacks on women. Embedded texts have vulgar words like *'bra', 'va**na', 't*ts', 's*xy', 'a*s'* targeting women. These memes indeed portray explicit misogyny and as per our analysis, model correctly classifies it as misogynistic and this decision does not get reverted due to occlusion. Here again, we conclude that annotations themselves are incorrect. ***HARM P+C dataset***: Here too we manually observe that most of the memes are indeed harmful and are possibly incorrectly annotated. The predictions of the model seem to be correct and occlusions do not change the predictions.

CASE 3: Original meme misclassified as negative and at least one occluded version resulted in the correct prediction (i.e., positive). Please see Table 4 for some representative examples. ***FHM dataset***: In this group, most of the memes have very small font size of the embedded text. Further the image has multiple objects or numerous color variations. This confuses the model leading to wrong predictions. Occlusion of these confusing regions allowed the model to focus on the parts of the image important for correct classification. ***MAMI dataset***: In most of the cases, image portrays nudity or other forms of vulgarity. In some memes, the embedded text contains the word *'MILF'* targeting women. Occlusion brings the focus of the model to these disturbing elements of the image leading to the correct prediction. ***HARM P+C dataset***: Majority of the memes contain the image of *'Trump'* or mention the words *'Trump', 'Covid-19'* or *'Corona'*. Length of embedded texts are very large in these memes which possibly confuses the model. Occlusion helps to bring back the focus of the model to the correct regions resulting in correct predictions.

CASE 4: Original meme misclassified as negative and none of the occluded versions resulted in the correct prediction (see Table 4 for some representative examples). ***FHM dataset***: Majority of the memes contain implicit hate. Individually neither the image nor the embedded text in the memes portray anything harmful. Most text have words like *'dishwater', 'sandwich maker', 'girl', 'wive', 'girlfriend'*. The images in these memes have cheerful faces of women with no vulgarity. When both the image and text are taken together they portray hate and, quite naturally, the model has difficulty in identifying this implied semantics even when parts of the image are occluded. ***MAMI dataset***: Once again these memes seem to bear implicit misogynistic content. Words like *'dishwater', 'sandwich maker', 'kitchen'*, and those referring to implicit body shaming appear in the embedded text. The model does not seem to have the requisite reasoning ability to infer the correct class of the meme and occlusion naturally does not come to any help. ***HARM P+C dataset***: A large majority of memes in this group portray fake conversations amongst political leaders. These conversations are implicitly harmful and thus the model misclassifies both the original and the occluded memes.

8.2 Typology of the error cases

While the previous section allowed us to obtain reasons for misclassification using the occlusion approach, it is largely manual. In this section we present an automatic method to induce the cases we observed earlier.

As a first step, for each dataset, we organise the misclassified data points into two groups – (a) misclassified as positive (hateful/misogynistic/harmful) and (b) misclassified as negative. Next for the data points for each group of each dataset we first obtain embeddings of the meme image + OCR text using the ***clip-ViT-B-32*** model. We then run **multimodal BERTopic** [2] on each group with number of clusters varying between 2 and 3 depending on the number of data points in the group. In the rest of this section we present the results obtained for each group.

Misclassified as positive - *FHM dataset*: We observe that the topics in the first cluster (Figure 1) contains the word *'gorilla'*. Nevertheless, we did not find images or induced topic words with profanity in this cluster. The second and third image clusters and the induced topic words cover most of our observations discussed in CASE 2. Some instances of CASE 1 are also observed. ***MAMI dataset***: All three clusters (Figure 2) largely correspond to CASE 1 discussed in previous subsection. None of the clusters correspond to CASE 2 discussed previously which contained nudity/vulgarity. ***HARM P+C dataset***: We observe in Figure 3 that the first and second image clusters contain majority of the memes with multiple images stacked together and have very long texts. This is analogous to CASE 1. Some of the topic words obtained in these clusters are *'quarantine', 'coronavirus', 'virus', 'china', 'corona'* which possibly confuses the model predictions. The third cluster is analogous to CASE 2 and depicts images which are indeed harmful.

Misclassified as negative - *FHM dataset*: Interestingly, in the first cluster, we get almost the same instances that we found in CASE 3 (Figure 4) where

[2] https://maartengr.github.io/BERTopic/getting_started/multimodal/multimodal.html

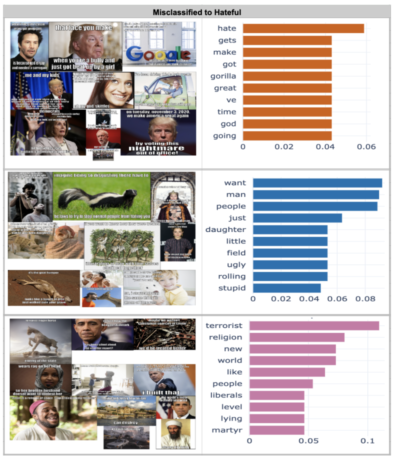


Figure 1: Misclassification to hateful memes in FHM dataset.

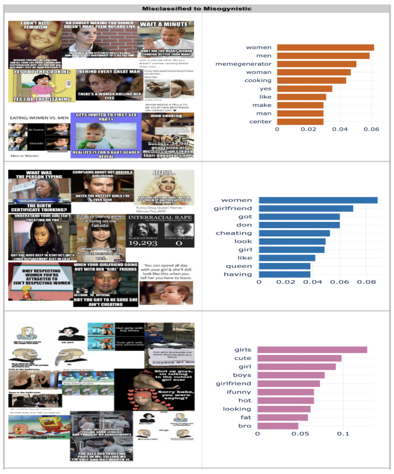


Figure 2: Misclassification to misogynistic memes in MAMI dataset.

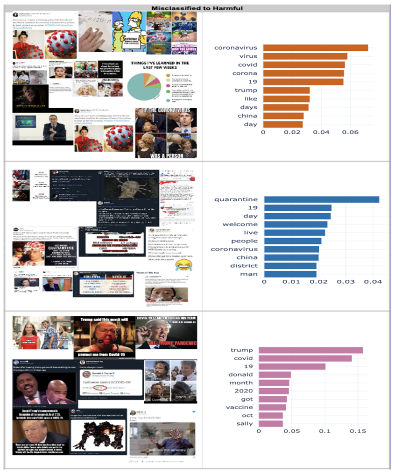


Figure 3: Misclassification to harmful memes in HARM C+P dataset.

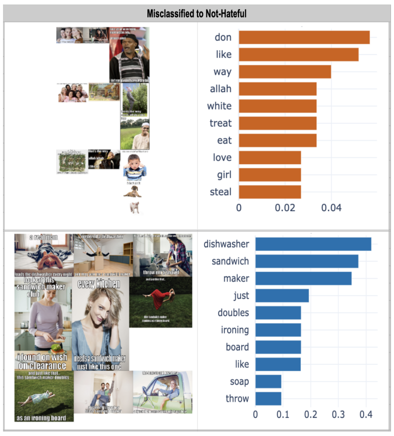


Figure 4: Misclassification to not hateful memes in FHM dataset.

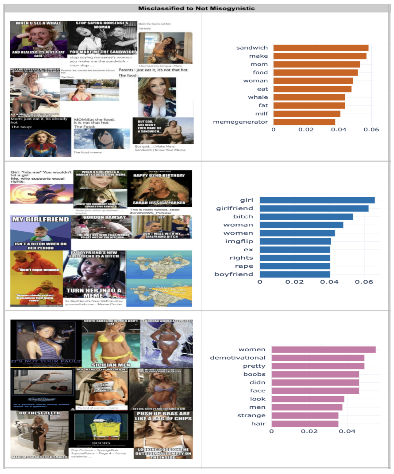


Figure 5: Misclassification to not misogynistic memes in MAMI dataset.

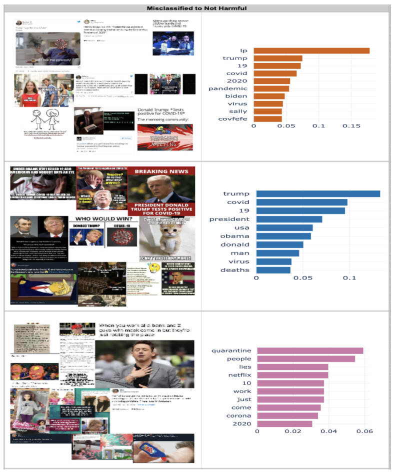


Figure 6: Misclassification to not harmful memes on HARM C+P dataset.

the images have irregular shapes and very small sized embedded text. The second cluster covers the point we discussed in CASE 4 and identifies topic words like *'sandwich'*, *'dishwater'*, *'soap'* with relevant associated images. ***MAMI dataset***: We observe that the first cluster (Figure 5) mostly resembles the CASE 4 discussed earlier. The second and third clusters contain images and topic words corresponding to CASE 3. ***HARM P+C dataset***: The two clusters in Figure 6 do not seem to be fully analogous to any of the cases. However, the second cluster/topic words partially resembles CASE 3. Overall we believe the above two subsections together provide invaluable insights into what are the systematic error patterns that VLMs are vulnerable to. These insights can be directly used in developing safety guardrails as opposed to expensive repeated fine-tuning.

9 Conclusion

We present a comprehensive study of popular open source VLMs for hateful meme detection, considering eight different prompt variants. For this study, we utilize four datasets covering various hate dimensions and observe that model performance varies based on datasets and prompts used. Furthermore, we also propose an approach to select the best model and prompt combination that generalizes well over considered datasets. Finally we present a systematic method to induce a typology of the errors committed by such VLMs which could have a long-term impact on how safeguarding approaches should be built in future.

10 Limitations

Our work has a few limitations. First, we conducted our experiments on English meme datasets and did not assess the model's capability for multilingual hate meme detection. Second, although we experimented with various prompt settings to identify misclassification patterns, these prompt variants are not exhaustive, and numerous other variants could be explored. Despite this, we are confident that our range of prompts can unveil the actual performance of VLMs in hate meme detection as they cover various broad meta-aspects. Third, we did not use hate meme datasets tailored explicitly for this task by fine-tuning the VLMs. In future, we plan to address these limitations.

11 Ethical statement

Our analysis refrains from attempting to trace users involved in disseminating hate, and we do not intend to harm any individuals or target communities. All experiments were thoroughly conducted using datasets crafted from prior research. Our primary focus was to assess the efficacy of large VLMs in hate meme detection, aiming to pinpoint potential areas for future enhancement.

References

Open AI. GPT-4 — openai.com. https://openai.com/research/gpt-4. [Accessed 15-02-2024].

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Shadi AlZu'bi, Ala Mughaid, Fatima Quiam, and Samar Hendawi. 2024. Exploring the capabilities and limitations of chatgpt and alternative big language models. In *Artificial Intelligence and Applications*, volume 2, pages 28–37.

Mohit Chandra, Dheeraj Pailla, Himanshu Bhatia, Aadilmehdi Sanchawala, Manish Gupta, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. "subverting the jewtocracy": Online antisemitism detection using multimodal deep learning. In *Proceedings of the 13th ACM Web Science Conference 2021*, pages 148–157.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Mithun Das and Animesh Mukherjee. 2023. Banglaabusememe: A dataset for bengali abusive meme classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15498–15512.

Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. 2022. Semeval-2022 task 5: Multimedia automatic misogyny identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 533–549.

Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2020. Exploring hate speech detection in multimodal publications. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1470–1478.

Kiana Kheiri and Hamid Karimi. 2023. Sentimentgpt: Exploiting gpt for advanced sentiment analysis and its departure from current machine learning. *arXiv preprint arXiv:2307.10234*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in Neural Information Processing Systems*, 33:2611–2624.

Hannah Kirk, Yennie Jun, Paulius Rauba, Gal Wachtel, Ruining Li, Xingjian Bai, Noah Broestl, Martin Doff-Sotta, Aleksandar Shtedritski, and Yuki M Asano. 2021. Memes in the wild: Assessing the generalizability of the hateful memes challenge dataset. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 26–35.

Yunshi Lan, Xiang Li, Xin Liu, Yang Li, Wei Qin, and Weining Qian. 2023. Improving zero-shot visual question answering via large language models with reasoning question prompts. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4389–4400.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M Rush, Douwe Kiela, et al. 2023. Obelisc: An open web-scale filtered dataset of interleaved image-text documents. *arXiv preprint arXiv:2306.16527*.

Hongzhan Lin, Ziyang Luo, Bo Wang, Ruichao Yang, and Jing Ma. 2024. Goat-bench: Safety insights to large multimodal models through meme-based social abuse.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. 2021. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103.

Krishanu Maity, Prince Jha, Sriparna Saha, and Pushpak Bhattacharyya. 2022. A multitask framework for sentiment, emotion and sarcasm aware cyberbullying detection from multi-modal code-mixed memes. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1739–1749.

Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021a. Detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2783–2796.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021b. Momenta: A multimodal framework for detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455.

Sarthak Roy, Ashish Harshvardhan, Animesh Mukherjee, and Punyajoy Saha. 2023. Probing llms for hate speech detection: strengths and vulnerabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6116–6128.

Lanyu Shang, Yang Zhang, Yuheng Zha, Yingxi Chen, Christina Youn, and Dong Wang. 2021. Aomd: An analogy-aware approach to offensive meme detection on social media. *Information Processing & Management*, 58(5):102664.

S. Süsstrunk, P. Fua, A. Shaji, A. Lucchi, K. Smith, and R. Achanta. 2012. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 34(11):2274–2282.

Minh-Hao Van and Xintao Wu. 2023. Detecting and correcting hate speech in multimodal memes with large visual language model. *arXiv preprint arXiv:2311.06737*.

A Definitions

The definitions provided below are picked from the corresponding dataset papers.

A.1 FHM dataset

- **hateful**: A direct or indirect attack on people based on characteristics, including ethnicity, race, nationality, immigration status, religion, caste, sex, gender identity, sexual orientation, and disability or disease. Attack is defined as violent or dehumanizing (comparing people to non-human things, e.g., animals) speech, statements of inferiority, and calls for exclusion or segregation. Mocking hate crime is also considered hateful.
- **not-hateful**: A meme which is not hateful and follows social norms.

A.2 MAMI dataset

- **misogynistic**: A meme is misogynous if it conceptually describes an offensive, sexist or hateful scene (weak or strong, implicitly or explicitly) having as target a woman or a group of women. Misogyny can be expressed in the form of shaming, stereotype, objectification and/or violence.
- **not-misogynistic**: A meme that does not express any form of hate against women.

A.3 HARM-C and HARM-P datasets

- **harmful**: Multi-modal units consisting of an image and a piece of text embedded that has the potential to cause harm to an individual, an organization, a community, or the society more generally. Here, harm includes mental abuse, defamation, psycho-physiological injury, proprietary damage, emotional disturbance, and compensated public image.
- **not-harmful**: Multi-modal units consisting of an image and a piece of text embedded which does not cause any harm to an individual, an organization, a community, or the society more generally.

B Prompt strategies

We provide a detailed list of templates for the corresponding prompt variants in Table 5.

C Reproducibility steps

We briefly summarize our methodology so that our research can be easily reproduced by the research community:

Datasets: All four datasets which we have used are commonly used for *hateful/misogynistic/harmful* meme detection tasks. The links to these datasets can be found here – *(FHM)*[3], *(MAMI)*[4] and *(HARM-C & HARM-P)*[5].

Processors: We used the respective model processors to process our images and text. From ***HuggingFace***, we used the `AutoProcessor.from_pretrained (model_checkpoint)` API and passed the image and text to the processor before feeding it to the model. Here we passed `model_checkpoint`

[3] https://www.kaggle.com/datasets/parthplc/facebook-hateful-meme-dataset

[4] https://github.com/TIBHannover/multimodal-misogyny-detection-mami-2022?tab=readme-ov-file

[5] https://github.com/LCS2-IIITD/MOMENTA

Prompt variants	Prompt templates
Vanilla (input) + Vanilla (output)	Classify the input meme as ‘positive_label’ or ‘negative_label’. Provide the answer as either ‘positive_label’ or ‘negative_label’ only. **Example output for ‘positive_label’ meme : ‘positive_label’** **Example output for ‘negative_label’ meme : ‘negative_label’**
Definition (input) + Vanilla (output)	Consider the following **definitions.** 1. ‘positive_label’ - **"Definition of ‘positive_label’ corresponding to dataset"** 2. ‘negative_label’ - **"Definition of ‘negative_label’ corresponding to dataset"** Classify the input meme as ‘positive_label’ or ‘negative_label’ based on the **above definitions** considering the image. Provide the answer as either ‘positive_label’ or ‘negative_label’ only. Example output for ‘positive_label’ meme : ‘positive_label’ Example output for ‘negative_label’ meme : ‘negative_label’
OCR (input) + Vanilla (output)	Classify the input meme as ‘positive_label’ or ‘negative_label’ considering the image as well as the **extracted text from the image which is delimited by three backticks.** **Extracted text from the image: “‘OCR extracted text goes here’”** Provide your answer in the format: ‘positive_label’ or ‘negative_label’. Example output for ‘positive_label’ meme : ‘positive_label’ Example output for ‘negative_label’ meme : ‘negative_label’
OCR & Definition (input) + Vanilla (output)	Consider the following **definitions** . 1. ‘positive_label’ - **"Definition of ‘positive_label’ corresponding to dataset"** 2. ‘negative_label’ - **"Definition of ‘negative_label’ corresponding to dataset"** Classify the input meme as ‘positive_label’ or ‘negative_label’ based on the **above definitions considering the image as well as the extracted text from the image which is delimited by three backticks.** **Extracted text from the image: “‘OCR extracted text goes here’”** Provide the answer as either ‘positive_label’ or ‘negative_label’ only. Example output for ‘positive_label’ meme : ‘positive_label’ Example output for ‘negative_label’ meme : ‘negative_label’
Vanilla (input) + Explanation (output)	Classify the input meme as ‘positive_label’ or ‘negative_label’. Provide the answer as either ‘positive_label’ or ‘negative_label’ only with an **explanation within 30 words explaining your classification.** Example output for ‘positive_label’ meme : ‘positive_label’ - **Explain within 30 words that why you classified this meme as ‘positive_label’.** Example output for ‘negative_label’ meme : ‘negative_label’ - **Explain within 30 words that why you classified this meme as ‘negative_label’.**
Definition (input) + Explanation (output)	Consider the following **definitions** . 1. ‘positive_label’ - **"Definition of ‘positive_label’ corresponding to dataset"** 2. ‘negative_label’ - **"Definition of ‘negative_label’ corresponding to dataset"** Classify the input meme as ‘positive_label’ or ‘negative_label’ based on the **above definitions** considering the image. Provide your answer as either ‘positive_label’ or ‘negative_label’ only with an **explanation within 30 words explaining your classification.** Example output for ‘positive_label’ meme : ‘positive_label’ - **Explain within 30 words that why you classified this meme as ‘positive_label’.** Example output for ‘negative_label’ meme : ‘negative_label’ - **Explain within 30 words that why you classified this meme as ‘negative_label’.**
OCR (input) + Explanation (output)	Classify the input meme as ‘positive_label’ or ‘negative_label’ considering the image as well as the **extracted text from the image which is delimited by three backticks.** **Extracted text from the image: “‘OCR extracted text goes here’”** Provide your answer in the format: ‘positive_label’ or ‘negative_label’, followed by an **explanation within 30 words explaining your classification.** Example output for ‘positive_label’ meme : ‘positive_label’ - **Explain within 30 words that why you classified this meme as ‘positive_label’.** Example output for ‘negative_label’ meme : ‘negative_label’ - **Explain within 30 words that why you classified this meme as ‘negative_label’.**
OCR & Definition (input) + Explanation (output)	Consider the following **definitions.** 1. ‘positive_label’ - **"Definition of ‘positive_label’ corresponding to dataset"** 2. ‘negative_label’ - **"Definition of ‘negative_label’ corresponding to dataset"** Classify the input meme as ‘positive_label’ or ‘negative_label’ based on the **above definitions considering the image as well as the extracted text from the image which is delimited by three backticks.** **Extracted text from the image: “‘OCR extracted text goes here’”** Provide your answer in the format: ‘positive_label’ or ‘negative_label’, followed by **an explanation within 30 words explaining your classification.** Example output for ‘positive_label’ meme : ‘positive_label’ - **Explain within 30 words that why you classified this meme as ‘positive_label’.** Example output for ‘negative_label’ meme : ‘negative_label’ - **Explain within 30 words that why you classified this meme as ‘negative_label’.**

Table 5: Representative examples for different prompt variants. Definition for corresponding labels can be picked from Appendix A. ‘positive_label’ and ‘negative_label’ will be replaced by corresponding labels as per the dataset. *Note:* Important text in each prompt variant has been indicated in **boldface** .

corresponding to the models we used (as discussed in section 4).

Quantization: We used ***HuggingFace*** APIs for quantization[6]. We specifically used *bitsandbytes*[7] library to setup 8-bit quantization.

Embedded text (OCR): We used the OCR text

[6] https://huggingface.co/docs/transformers/en/main_classes/quantization

[7] https://huggingface.co/docs/bitsandbytes/index

already present in the dataset for our experiments on memes.

D Examples of ambiguous outputs

We report four representative example outputs obtained from the VLMs where it was not able to classify amongst `list_of_labels`. Corresponding model, prompt variant and dataset is mentioned for each of these examples. Important text leading to ambiguity is highlighted. Due to these ambiguous outputs, we set a threshold of 90% support while presenting and comparing our results.

(i) IDEFICS: Definition Input + Explanation output (*MAMI* dataset)

`Misogynistic` `- The meme depicts a woman running from a car, which is a stereotype that women are weak and need to be protected. The woman is also shown with a large belly, which is an objectification of women's bodies. The meme is sexist and hateful towards women.`
`Not-misogynistic` `- The meme is not misogynistic because it does not depict any form of hate towards women. It is a simple joke that does not have any negative connotations towards women.`

In the above output, model instead of choosing only one label out of *{Misogynistic or Not-misogynistic}* and explaining the reason behind its classification, provided an output with both labels. It further went on to justify both of the labels.

(ii) INSTRUCTBLIP Vicuna 7B: Vanilla Input + Explanation output (*FHM* dataset)

`Example output for hateful meme: hateful` `- This meme is hateful because it makes fun of a woman's appearance and implies that she is not capable of being a good mother.'`
`Example output for not-hateful meme: not-hateful` `- This meme is not hateful because it is a lighthearted comment about a woman's appearance and does not imply any negative judgments about her abilities as a mother.`

In the above output, model repeated the example output format provided to it for both labels *{Hateful and Not-hateful}*. It then further provided an explanation for both labels corresponding to the input meme.

(iii) INSTRUCTBLIP Vicuna 7B: OCR Input + Vanilla output *(FHM Dataset)*

`The image contains a man who is eating a banana, and the text "bravery at its finest" is written above it. Based on the context and the extracted text, it is not possible to determine whether the meme is` **`hateful or not-hateful`**.

In the above output, the model does not adhere to the desired output format. Instead of generating only the label out of *{Hateful or Not-hateful}*, it gives a brief explanation and finally infers that it is not possible to determine whether the meme is hateful or not.

(iv) INSTRUCTBLIP FlanT5-xl: OCR Input + Explanation output (*HARM-C* dataset)

`The message from President Trump to Covid-19 is a meme that is a warning to people to leave the country.`
`The message is not harmful, but it is not harmless`.

Here, the model's output does not adhere to the desired output format. Moreover, model diplomatically bypassed the classification task by considering meme to be neither harmful nor harmless.