SEMANTIFY: Unveiling Memes with Robust Interpretability beyond Input Attribution

Dibyanayan Bandyopadhyay[1] , **Asmit Ganguly**[1] , **Baban Gain**[1] and **Asif Ekbal**[2]
[1]Department of Computer Science and Engineering, Indian Institute of Technology, Patna, India
[2]School of AI and Data Science, Indian Institute of Technology, Jodhpur, India
{dibyanayan, asmit.ganguly, gainbaban, asif.ekbal}@gmail.com

Abstract

Memes, initially created for humor and social commentary, have transformed into platforms for offensive online content. Detecting such content is crucial; however, existing deep learning-based meme offensiveness classifiers lack transparency, functioning as opaque black-box systems. While Integrated Gradient and similar input-attribution interpretability methods exist, they often yield inadequate and irrelevant keywords. To bridge this gap, we introduce SEMANTIFY, a novel system featuring a theoretically grounded multi-step filtering process. SEMANTIFY extracts meaningful "tokens" from a predefined vocabulary, generating a pertinent and comprehensive set of interpretable keywords. These extracted keywords reveal the model's awareness of hidden meanings in memes, enhancing transparency. Evaluation of SEMANTIFY using interpretability metrics, including 'leakage-adjusted simulatability,' demonstrates its superiority over various baselines by up to 2.5 points. Human evaluation of 'relatedness' and 'exhaustiveness' of extracted keywords further validates its effectiveness. Additionally, a qualitative analysis of extracted keywords serves as a case study, unveiling model error cases and their reasons. SEMANTIFY contributes to the advancement of more interpretable multimodal systems for meme offensiveness detection, fostering trust for real-world applications.

1 Introduction

In recent times, memes have emerged as a ubiquitous form of online expression, blending humor, satire, and social commentary to encapsulate complex ideas in a single image or short video. While created to disseminate humor, it is often misused to perpetuate societal harm [Kiela *et al.*, 2021]. A significant portion of the meme ecosystem is tainted with content that is offensive, hateful, or even dangerous. Therefore, it is crucial to develop effective tools for the automated detection of offensive memes, to preserve the integrity of online spaces.

However, a simple classification of memes as offensive is often insufficient. Making the system interpretable is paramount as it can elucidate whether the system learns from spurious correlations in the data or whether it can reliably classify a meme as offensive. This clarity aids in enhancing user trust on these systems through transparency. Further, interpretability methods help users to know if the model acquired some kind of inadvertent biases while training.

Existing input-attribution-based explainability methods like LIME [Ribeiro *et al.*, 2016], SHAP [Guo *et al.*, 2019], and GradCAM [Selvaraju *et al.*, 2019] work well in practice but suffer from two issues, *viz.* i) *Semantic Irrelevance:* The keywords that are attributed to model predictions are often semantically irrelevant to the model input, making it hard for humans to assess their effect on the model's behavior; and ii) *Incohesive:* Existing input-attribution methods operate on the input space, generating a set of keywords that often lack a central theme and consequently miss crucial words essential for a more comprehensive explanation of the model and its predictions.

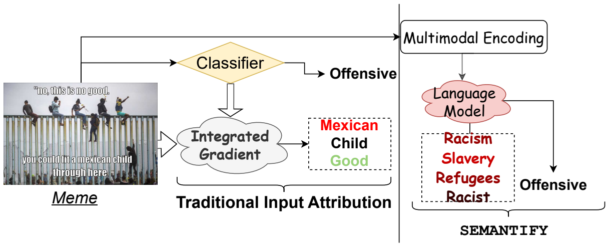


Figure 1: We compare and contrast SEMANTIFY vs *Integrated Gradient*. Observe the relevance of extracted keywords obtained by SEMANTIFY. Red denotes offensive and green denotes neutral or non-offensive keywords.

An example of the above drawbacks can be shown in Figure 1, where we compare our method, SEMANTIFY with Integrated Gradient [Sundararajan *et al.*, 2017], which is an input-attribution based explainability method. The tokens retrieved by input attribution are 'Mexican', 'Child', and 'Good', which are *Semantically Irrelevant* to the hidden meaning of the meme which is associated with 'Racism'. In contrast, our method, SEMANTIFY consults a large set of vocabulary space and retrieves a much more relevant set of keywords (e.g. 'Refugees', 'Slavery' etc.). Also, the set of retrieved keywords from Integrated Gradient lacks a central theme, which makes them *Incohesive*.

We enumerate the major contributions/attributes of our current work as follows[1]:

1. We propose a theoretically grounded technique that (refer *alignment-optimization correlation* in Section §3.3) could explain a model's behavior by retrieving 'tokens' from a global vocabulary space. The retrieved tokens are compared with input attribution based baselines and found to carry both a 'faithful' representation of the input meme as well as semantically relevant information.
2. Our method is extensively evaluated for both automatic and human evaluation. A detailed analysis is performed to assess its effectiveness.
3. While we demonstrate the application of our method in the realm of internet memes, it is fundamentally model and task-agnostic. Expanding its application to other tasks is beyond the scope of this paper and is left for future research.

2 Related Work

Multimodal Offensiveness Detection. In the realm of Natural Language Processing (NLP), previous research has primarily concentrated on identifying offensive content [Waseem and Hovy, 2016; Sarkar *et al.*, 2021], addressing cyberbullying [Van Hee *et al.*, 2018], hate speech [Caselli *et al.*, 2021], and similar issues within social media posts [Roberts *et al.*, 2012]. Nevertheless, these computational methods have predominantly been evaluated using textual data. Turning to visual offensiveness detection, earlier investigations have centered on identifying sexually explicit images [Ganguly *et al.*, 2017]. The pivotal moment came when Kiela *et al.* [2021] introduced a set of benchmarks and released the Facebook Hateful meme dataset, which ignited research in this field. This led to a series of research on detecting offensiveness in multimodal media [Yang *et al.*, 2022], particularly in memes [Sharma *et al.*, 2020]. Suryawanshi *et al.* [2020] used an early fusion method for combining visual and textual features, leading to more accurate detection. Chen and Pan [2022] stacked visual features, object tags, and text features to Vision-Language Pre-Training Model with anchor points to detect offensive memes. While these models are proven to be useful for predictions, their outputs are not interpretable and cannot be reliably used in real-world use cases.

Multimodal Interpretability. Recently, there have been a notable number of multimodal models [Du *et al.*, 2022; Li *et al.*, 2023; Liu *et al.*, 2023b; Zhu *et al.*, 2023] for various tasks. However, there is a dearth of research on generating explanations or justifications around their predictions. Researchers predominantly relied on interpretability techniques. LIME [Ribeiro *et al.*, 2016] explains predictions of any classifier by fitting a sparse linear model locally around the instance being explained. It converts the instance into a binary vector, indicating the presence or absence of interpretable components (like words). SHAP [Lundberg and Lee, 2017a] explains machine learning model's predictions by computing Shapley values, inspired from game theory. These values represent each feature's contribution to a prediction. Gradient heatmap [Guo *et al.*, 2019] explains predictions by computing gradients of the model output concerning the input features.

However, recent years have witnessed a shift in the focus of interpretability research, recognizing the potential for generating natural language explanations for both unimodal and multimodal systems [Kayser *et al.*, 2021]. Instead of traditional end-to-end training, Koh *et al.* [2020] first predicted concepts and used those to predict the labels such that the model could be interpreted by changing the concepts. There exist some natural-language-based techniques like wt5 [Narang *et al.*, 2020], which is available for text-only systems. Some recent methods like NLX-GPT [Sammani *et al.*, 2022] bridges the gap between text-based and multimodal natural language generation. Cross-modal attention, which attends to the distinguishing features between text and image modalities, is used in the transformer encoder for sarcasm explanation [Desai *et al.*, 2021]. Sharma *et al.* [2022] generates explanations for visual semantic role labeling in memes. These methods together can generate textual explanations for the behavior of multimodal models. As shown in Figure 1 and mentioned in the Introduction, all of these current methods fall short in adequately explaining model behavior for inputs with implicit meaning. To tackle this, we aim to address the issue by retrieving keywords aligned with the model's inner workings.

3 Methodology

The proposed systems combine a multimodal encoder for an input meme and a language model (LM). The LM acts both as the *final* classifier and interpretable keyword retriever. It categorizes the input meme as either non-offensive or offensive, and, through sampling, can extract a set of keywords to elucidate its prediction. Thus, our system follows a two-step strategy, i.e. i) Multimodal encoding followed by ii) Classifying via a language model (LM). We elaborate the steps in details as below:

3.1 System Design

Multimodal Encoding. Let M denote the input meme (single instance), consisting of an image V and accompanying text T. We utilize a pre-trained and frozen CLIP model [Radford *et al.*, 2021] to obtain textual (f_t) and visual (i_t) representations. These features, with dimensions $\mathbb{R}^{m\times 1}$ and $\mathbb{R}^{n\times 1}$ respectively, are used to generate a multimodal representation $M_t \in \mathbb{R}^{o\times 1}$ (here, $m = n = 512$).

The fusion process employs trainable weight matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ with dimensions $\mathbb{R}^{m\times ko}$. The multimodal representation is calculated as follows: $\boldsymbol{M}_t = AveragePool(\boldsymbol{U}^T\boldsymbol{f}_t \circ \boldsymbol{V}^T\boldsymbol{i}_t, k)$, where $\circ$ denotes element-wise multiplication, k

[1] Code and Supplementary Material available at: https://github.com/newcodevelop/semantify

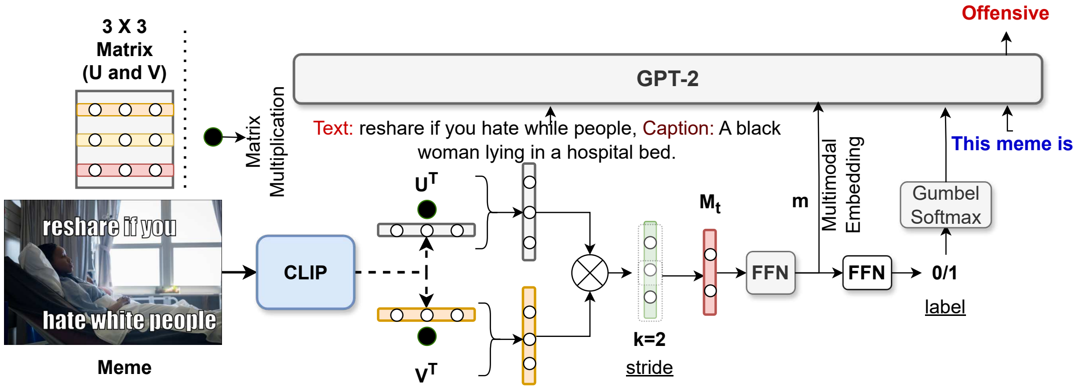


Figure 2: Detailed Diagram outlining the system underlying the proposed framework. Multimodal encoding as well as the gumbel logits from the first stage is forwarded to the second stage where GPT2 classifies the input meme as offensive or normal. *The exact input prompt used is more verbose, simplified here for illustration.*

represents the stride for the overlapped window used in the pooling operation, and o denotes the output dimension. This encoding scheme, inspired by a similar approach [Bandyopadhyay *et al.*, 2023], maintains high performance with a low parameter count. This multimodal encoding is passed through two feed-forward neural networks (FFNs) which produce predicted labels l (0 or 1) corresponding to the input meme.

Using LM as the Classifier cum Retriever. We utilize a GPT2 model [Radford *et al.*, 2019] as the classifier as well as for generating the interpretive keywords. The input prompt of GPT2 constitutes of the plain text input of the 'meme text' and the 'meme caption', which is automatically generated via OFA [Wang *et al.*, 2022] module. Along with these plain text input prompts, it also takes an embedding representation $\boldsymbol{m} \in \mathbb{R}^{1\times 1024}$, which is obtained by projecting $\boldsymbol{M}_t$ via an FFN. Further, to facilitate backpropagation, the predicted label from the first stage, l, is passed through Gumbel softmax [Jang *et al.*, 2017] which produces a soft logit score, which is then passed as an input token. The input prompt is then appended by a fixed string: 'This meme is'. The model is then trained to output 'offensive' or 'normal' (the label associated with the meme in the training data) as the next token of the input prompt via causal language modeling objective.

The system architecture is visualized in Figure 2.

3.2 Retrieving Keywords Beyond Input Space

Obtaining a comprehensive set of *human-comprehensible* and *interpretable* keywords that effectively encapsulate the operational characteristics of a classifier when presented with a given input typically poses a challenge. One plausible approach to address this issue involves the utilization of input-attribution-based methodologies, such as Integrated Gradient or Attention Heatmaps. Such techniques serve to highlight specific parts of the input data that are correlated to the classifier output. However, it is noteworthy that these methods only yield set of tokens present in the input space, thereby limiting their applicability due to lack of diversity, as it is illustrated by an example in Figure 1. The proposed method for extracting the set of relevant keywords (interchangeably called tokens) involves four filtering steps involving the vocabulary set of the language model used (i.e. GPT2).

1. **Maximally Relevant:** First, we filter out the keywords that are not relevant continuation of the input prompt. The GPT2 model is trained via causal language modeling (CLM) objective to predict either 'offensive' or 'normal' as the next token of the input prompt. The CLM objective ensures that Top-K sampling of the next token gives the k most plausible set of tokens that are maximally relevant continuation of the input prompt.

2. **Alignment - Optimization Correlation:** The set of extracted keywords from the first step (*their embedding is denoted as* $\boldsymbol{e}$) should be such that they belong in a special ϵ neighborhood. This is an additional filter that theoretically ensures that the set of keywords does not possess redundant information while also not completely alienated from the optimization objective. The definition and interpretation of this is presented in Section 3.3.

3. **Semantically Relevant:** In this step, we filter additional keywords that are semantically irrelevant to the input meme. Mathematically, we take the cosine similarity between $\boldsymbol{m}$ and $\boldsymbol{t}_i$, where $\boldsymbol{t}_i$ is the text embedding of i-th keyword obtained through the CLIP Text encoder. Finally, we sort them by values of cosine distance and only keep top 20 keywords out of them. CLIP being trained with contrastive learning only preserves tokens that are semantically relevant to the input meme.

4. **Prediction Preserving :** The fourth step is the most stringent one. First, we use the trained LM in inference mode to generate *knowledge text* by passing extracted tokens as input prompt. Again, together with the extracted tokens, we pass their generated knowledge text to the LM. If the model predicts the same class as it has predicted before, we call the passed token *prediction preserving*. If the passed token flips the actual prediction then we can confidently say that the token is not causally related to model prediction and thus it is not faithful. We

filter out only top four keywords after this step by sorting with respect to the log-likelihood score of the predicted tokens in decreasing order.

3.3 Alignment - Optimization Correlation

In the pursuit of optimizing machine learning models, we often encounter the challenge of striking the right balance between the alignment of information vectors and optimization efficiency. To explore this delicate trade-off, we introduce the following theorem.

We first assume that our objective function $\hat{y} = f(\boldsymbol{m})$ is strongly convex. Here, $\hat{y}$ shows the model predicted class. Also, we consider two non-zero real column vectors $\boldsymbol{e}$ (token embedding) and $\boldsymbol{m}$ (multimodal embedding of the input meme). $\nabla_{\boldsymbol{m}}\hat{y} = \nabla_{\boldsymbol{m}} f(\boldsymbol{m})$ refers to the gradient of the prediction with respect to $\boldsymbol{m}$ such that $\boldsymbol{m}^+ = \boldsymbol{m} + \gamma\nabla_{\boldsymbol{m}} f(\boldsymbol{m})$.

Theorem 1. *With very small step size γ, the condition $\boldsymbol{e}^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m}^+) > \boldsymbol{e}^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m})\rho$, where $\rho > 1$, holds true.*

This theorem carries substantial empirical implications:

i) Sampling tokens with their token embedding $\boldsymbol{e}$ such that $\boldsymbol{e}^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m}) > 0$ would imply alignment between $\boldsymbol{e}$ and $\nabla_{\boldsymbol{m}} f(\boldsymbol{m})$, i.e. moving $\boldsymbol{m}$ in the direction of $\boldsymbol{e}$ aids optimization. As demonstrated by the left-hand side (LHS) of the inequality, successive gradient ascents on $\boldsymbol{m}$ progressively reduce the angle between $\boldsymbol{e}$ and $\nabla_{\boldsymbol{m}} f(\boldsymbol{m})$. Intuitively, this entails $\boldsymbol{e}$ aids in the model optimization process, thus its corresponding tokens are also interpretable.

ii) With $\nabla_{\boldsymbol{m}} f(\boldsymbol{m})$ being smooth and differentiable, when $\boldsymbol{e}^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m}) \to 0$, we find that $\boldsymbol{e}^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m}^+) > 0$. Even as $\boldsymbol{e}$ and $\nabla_{\boldsymbol{m}} f(\boldsymbol{m})$ approach near-orthogonality, indicative of $\boldsymbol{e}$ carrying diverse information rather than perfect alignment with the gradient, the positive value of $\boldsymbol{e}^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m}^+)$ signifies $\boldsymbol{e}$ as aligned for subsequent gradient-based optimization steps w.r.t $\boldsymbol{m}$. We term this phenomenon the *'Alignment - Optimization Correlation Criteria'*. In practical applications, this serves as a filtering mechanism to retain tokens relevant to regions where $\boldsymbol{e}^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m}) \to 0$. This condition is henceforth referred to as ϵ ball or ϵ neighborhood constraint interchangeably. This theoretically grounded motivation significantly enhances our ability to extract diverse yet interpretable tokens, as shown through results and analysis. The proof of this Theorem 1 and the physical implication are described in detail in Technical Supplementary Material Section A and B, respectively. The Algorithm for our filtering process is shown in Algorithm 1.

4 Experiments and Analysis

4.1 Experimental Setup

Our proposed model was constructed using PyTorch, a Python-based deep-learning library. In our experimentation, we imported GPT-2 from the Huggingface transformers package. All experiments were conducted on a single Nvidia A100 80GB GPU. We employed the Adam optimizer [Kingma and Ba, 2017] with a learning rate of 0.005 for optimization. We use the Facebook Hateful Meme dataset [Kiela *et al.*, 2021] for performing the experiments.

To ensure robust evaluation on simulatability, we conduct a 5-fold cross-validation for testing the surrogate models (§Section 4.2) after running experiments for $3,500$ steps on the respective train set. We report averaged scores obtained from 5 experiment runs. Additionally, we maintain a consistent random seed of 42 across all our experiments.

Algorithm 1 Retrieve explainable keywords with four step filtering

```
$explain\_out$ = [] ;                          /* Final token placeholder */
$first\_stage$ = [] ;      /* Placeholder or TopK & $\epsilon$ neighborhood
  constraint */
$\boldsymbol{r}_{clip}$ ← Meme Image Embedding from CLIP
  { $\boldsymbol{t}_i$ } ← Top-k tokens from Vocabulary set $V$ ;        /* TopK Filtering */
for $t_i \in \{t_i\}$ do
    $\boldsymbol{e}_i$ ← GPT2Embedding( $\boldsymbol{t}_i$ )
    if $\|\boldsymbol{e}_i \cdot \nabla_{\boldsymbol{m}}\hat{y}\| \leq \epsilon$ then
        $\boldsymbol{t}_{i_{clip}}$ ← Text Embedding from CLIP($\boldsymbol{e}_i$)
        $sim_{cosine} \leftarrow \boldsymbol{r}_{clip} \cdot \boldsymbol{t}_{i_{clip}}$
        $first\_stage$ . append($\{t_i : sim_{cosine}\}$) ; /* filtered tokens
          from $\epsilon$ neighborhood */
    end
end
$\{\boldsymbol{t}'_i\}$ ← Top 20 tokens decreasingly sorted by $sim_{cosine}$ from $first\_stage$. ;
  /* CLIP filtering */
for $t'_i \in \{t'_i\}$ do
    $\boldsymbol{e}'_i$ ← GPT2Embedding( $\boldsymbol{t}'_i$ )
    if $f(\boldsymbol{e}'_i) = \hat{y}$ then
        $explain\_out$ . append($\boldsymbol{t}'_i$) ;           /* Output preservation
          filtering */
    end
end
$explain\_out$ ← Top 4 tokens from $explain\_out$ sorted by log likelihood. ;
  /* Final step */
```

4.2 Results and Analysis

Automatic Evaluation

For Automatic evaluation, we resort to using 'model faithfulness' as a guiding principle to evaluate the effect of the obtained keywords on model behavior. Especially, we measure 'simulatability', which can be defined as how well we can use the extracted keywords to predict (i.e. simulate via another surrogate model) the model's output. In Table 1, we depict the effect of various filtering mechanisms forming an ablation and compare our proposed method with the well-known input attribution-based methods, e.g. Integrated Gradient and KernelSHAP. For comparison, we use i) *Leakage adjusted simulatability (LAS)* [*Hase* et al.*, 2020*] score, which measures the effect of predicted keywords/explanation on simulating model prediction opting for explanation leakage. A positive LAS score reflects better 'simulatability' or 'faithfulness' of the extracted keywords corresponding to the model prediction. For evaluating the effect of extracted keywords on simulator confidence, we use ii) *Comprehensiveness (↑)* and iii) *Sufficiency (↓)* metrics [DeYoung *et al.*, 2020]. Intuition and technical details of the measurement of these metrics can be found in Supplementary Section I. We also list three accuracy-based measures for the simulator: i) F1 score using both generated explanation and input meme as input to the model (denoted as **F1**), ii) F1 score using only input (denoted as **F1 w/ inp**) and iii) F1 score using only explanation (denoted as **F1 w/ exp**). We also propose two metrics: i) Inter-sample diversity defined as Div (Inter) and ii) Intra-sample diversity defined as Div (Intra) elaborated in the Supplementary Material Section H. These metrics respectively measure the average diversity/exhaustiveness of all the

TopK	**ϵ-ball**	**SR**	**OP**	**Div (Inter)**	**Div (Intra)**	**LAS (↑)**	**Compre. (↑)**	**Suff. (↓)**	**F1 (%)**	**F1 w/ inp (%)**	**F1 w/ exp (%)**
Random.	-	-	-	-	-	0.0	$-2.42x10^{-4}$	27.0	79	79	40
Saliency Map	-	-	-	4.06	7.43	-0.4	0.4	14.0	79	79	68
Inp x Grad [Shrikumar *et al.*, 2017]	-	-	-	3.25	7.40	-0.4	0.5	14.8	79	79	69
Int. Grad. [Sundararajan *et al.*, 2017]	-	-	-	3.62	6.94	0.0024	0.3	16.0	79	79	65
KernelSHAP [Lundberg and Lee, 2017b]	-	-	-	4.01	6.57	0.052	0.2	18.0	79	79	59
×	×	✓	✓	10.81	3.81	-0.9	2.8	16.0	82	79	74
×	0.05	✓	✓	6.28	6.94	0.5	2.7	14.0	82	79	76
×	0.01	✓	✓	5.96	7.47	1.0	3.5	13.3	83	79	76
3500	×	✓	✓	5.53	7.21	2.0	4.3	7.9	84	79	84
2500	×	✓	✓	5.48	7.16	2.3	4.2	7.7	84	79	85
1500	×	✓	✓	5.23	7.17	1.6	3.8	7.7	83	79	85
500	×	✓	✓	4.76	7.19	1.2	**5.6**	**6.9**	**85**	79	87
3500	0.1	✓	✓	5.53	7.19	**2.7**	4.3	8.5	**85**	79	85
2500	0.1	✓	✓	5.48	7.16	1.9	4.1	7.5	84	79	85
1500	0.1	✓	✓	5.24	7.17	1.9	3.8	7.6	83	79	85
500	0.1	✓	✓	4.75	7.15	2.3	**5.6**	7.1	**85**	79	**88**

Table 1: Automatic evaluation for faithfulness. Empirical performance of our proposed method in different setups (ablations) and comparison with baselines. *F1 w/ inp* is redundantly kept in the table to aid easier comparison. All the metrics are linearly scaled in the range of 0 to 100 for better comparison. SR (semantic relevance) and OP (output preserving) refer to the CLIP-based filtering step and output preservation steps, respectively.

retrieved keywords across the dataset and for a specific example. The trade-off between these two illustrates various properties of the retrieved keywords, as illustrated in the discussion below.

Comparison with Baselines. In the first five rows of Table 1, we describe the effect of extracting keywords from various input attribution-based explanation methods which are compared with random keywords. As expected, the random keywords obtained very low scores for all metrics reflecting the input attribution-based methods which work well in practice. For every setup of our proposed model in the remaining rows, we observe the superiority of our proposed approach by observing better obtaining scores for all the metrics. We also observe that although *F1 w/ exp* score is better for the baselines compared to the 'Random' explanations, the model performance remains the same when explanation and input both are used as input, as seen through the same F1 score obtained. This intuitively illustrates the fact that the extracted explanations do not provide extra information compared to inputs, such that the before and after F1 scores remain the same. Comparison with LLaVA-13B [Liu *et al.*, 2023a] can be found at Supplementary Material Section D.

i) **Does epsilon ball constraint work in the absence of Top-K constraint?** Firstly, in rows 6 to 8, we consider dropping off the Top-K sampling restriction (first stage) and observing the effect of disabling and enabling the second stage with different values of ϵ. Without any ϵ constraint, we obtain a negative LAS score along with a low 'comprehensiveness' score. It shows that only selecting the keywords using CLIP-based representation does not retrieve semantically relevant keywords. Next, we enable the second stage with two separate values of $\epsilon \in \{0.05, 0.01\}$. As can be seen through tabulated metrics, enabling the second stage has a positive effect on the quality of retrieved keywords. Also, $\epsilon = 0.01$ works better than $\epsilon = 0.05$ in terms of performance, suggesting that our theoretical justification of retrieving tokens in the neighborhood of $e^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m}) \to 0$ indeed works well in practice. However, lowering the ϵ by too much ($\epsilon \sim 0$) would result in non-retrieval of any keyword.

ii) **Why would a larger Top-K be associated with a lower comprehensiveness score?** From the Inter sample diversity score, we observe that a higher Top-K value relates to higher Inter sample diversity, which entails that diversity between two explanation sets will be larger. Intuitively, it can be seen that evaluating the simulator model on a more diverse explanation set leads to a lower probability of the predicted class due to lower model confidence. This consequently leads to lower comprehensiveness and higher sufficiency scores. It may be observed that there is a steady increase in inter-sample diversity with increasing Top-K value, which further leads to lower comprehensiveness scores.

iii) **For the same Top-K value what would the effect of enabling ϵ constraint be?** Comparing scores from the third and fourth set of rows, we observe that enabling the ϵ constraint seems to be beneficial for the 'simulatability', as can be seen by higher LAS scores for the same Top-K value without the ϵ constraint. This can be attributed to the same inter-sample diversity (indicating variations among samples) but lower intra-sample diversity (indicating lesser variation among retrieved keywords specific to an input meme). Less variation among the retrieved keywords for an input meme would intuitively mean better simulatability. However, this case is not always true, as a very low intra-sample diversity score would entail that the retrieved keywords are very similar and would result in a low LAS score (observe the sixth row). Intuitively, there is an optimal spot where the ratio of inter-sample and intra-sample diversity would indicate optimally selected retrieved keywords.

iv) **What is the similarity of the retrieved explanations using a specific Top-K value w.r.t various ϵ balls?** We observe that enabling the Top-K constraint unequivocally retrieves better tokens as illustrated by Table 1. To theoretically justify it, we measure Jaccard similarity between the a) set of tokens retrieved using a specific Top-K value and b) tokens retrieved from the open ϵ neighborhood $[-\epsilon, +\epsilon]$. This test is done on a randomly selected set of 100 test memes, keeping equal representation from all over the full test set. From Figure 3, we observe Jaccard similarity value spikes at $[-0.01, +0.01]$ when $TopK \in \{3500, 2500, 1500\}$ and at $[+0.02, +0.03]$ when $TopK \in \{500\}$. This entails Top-K retrieved tokens mostly lie in the neighborhood where $e^T \cdot \nabla_{\boldsymbol{m}} f(\boldsymbol{m}) \to 0$, which is theoretically justifiable.

v) **Is the Output preservation stage necessary?** We state that this stage is of utmost importance. From Figure 4, we ob-

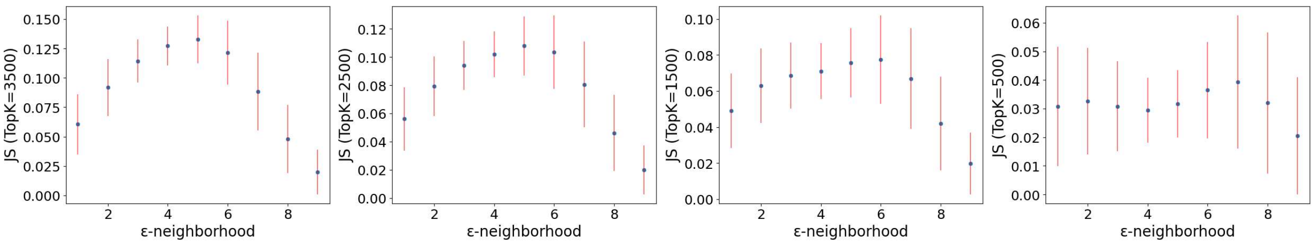


Figure 3: Plot of Jaccard similarity (Y-axis) between set of tokens in specific ϵ neighborhoods and for specific Top-K sizes. Each plot refers to a specific Top-K size. Each point in X-axis refers to an ϵ neighborhood bin starting from '-0.09 to -0.07' and ending at '$+0.07$ to $+0.09$'

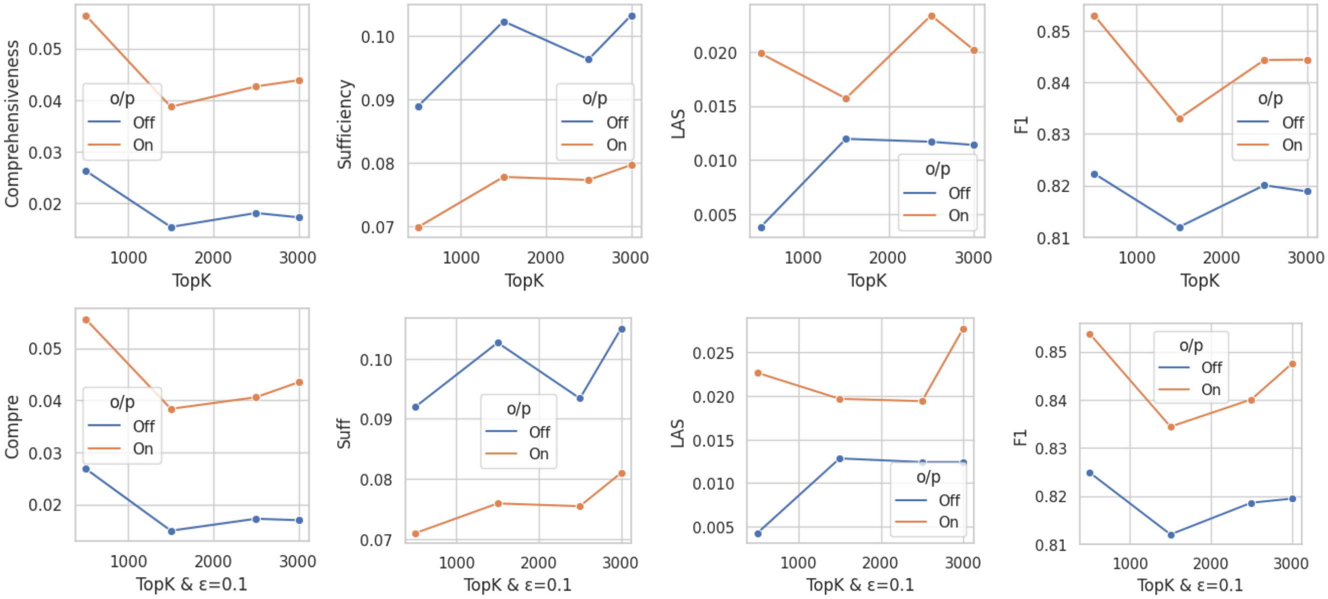


Figure 4: We plot values for various metrics for various fixed Top-K values and ϵ values in X axis. On: O/P Preservation stage is on and Off: O/p preservation stage is off.

serve that for different Top-K values (both with and without ϵ constraint enabled), enabling the output preservation stage always retrieves more interpretable keywords rather than disabling it. The same test subset is used to perform these experiments, as illustrated in the previous point. Also, the performance gap is quite significant, as seen in Figure 4.

vi) **What is the interpretive impact of keywords on model decisions?** We input the extracted keywords as knowledge tokens during testing to empirically verify their influence on the model. The analysis reveals that in the majority of cases (86.23%), the predicted probability for the model-assigned class increased, with a mean rise of 0.06 and a standard deviation of 0.07. Moreover, the predicted class remains consistent in 97.41% of cases. Additionally, incorporating keywords does not adversely affect the model; instead, it boosts confidence in predicting the class, emphasizing the interpretive value of keywords in shaping the model's decisions.

vii) **Is semantic relevance (CLIP filtering) stage necessary?** For every set-up in Table 1, we manually tested a random set of same 30 memes with and without the CLIP filtering stage enabled. Without CLIP filtering, the quality of the retrieved keywords are worse such that they do not semantically match with the input meme, which renders them unusable for explanation purposes.

viii) **How well does it classify input memes?** The macro-F1 and Accuracy score on the offensiveness classification task using our proposed system are 73.46% and 75.64% respectively on the test set. This is at par or better than the standard visuo-lingual baselines used in [Kiela *et al.*, 2021] where the FB hateful meme dataset was originally proposed. We also observe that the GPT2 prompt without the projected multimodal embedding M_t obtains only around 57% F1 score.

Human Evaluation

We perform a human evaluation of the generated keywords using two metrics, *viz.* Relatedness and Exhaustiveness. Relatedness is defined as how much a set of generated keywords is relevant to the content of the input meme, and Exhaustiveness is defined as how much of the aspects of an input meme are correctly represented in a set of retrieved keywords. The scale of these metrics is illustrated in Supplementary Material Section G. Based on these definitions, five people (three authors of this paper and two undergraduate students) are cho-

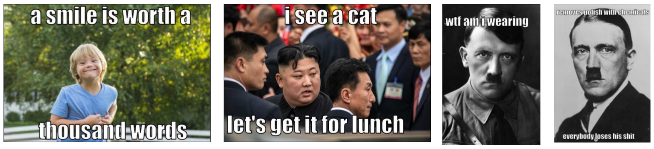
	Meme	Ours	ϵ-ball w/ CLIP	CLIP only	Integrated Gradient	Pred	Act
Correctly Classified	01276	*philanthrop* *words* *encourage* *happiness*	*charitable* *encourage* *charity* *estimates*	*optimistic* *optimism* *worth* *Worth*	*smile* *worth* *thousand* *words*	0	0
	98724	*jews* *holocaust* *nazis* *hitler*	*nazis* *adolf* *sergei* *churchill*	*Adolf* *ologists* *Stalin* *Polish*	*shits* *loses* *polish* *normal*	1	1
Misclassified	91768	*jew* *holocaust* *hitler*	*jew* *abolished*	*jew* *jew* *Jew* *Jew*	*wearing* *:wtf* *normal* *adolf*	1	0
	13875	*cats* *cat* *lunch* *eat*	*cat* *cooperation* *sandwiches* *menu*	*cats* *cats* *cat* *Cat*	*see* *normal* *lunch* *let's*	0	1

Table 2: Qualitative analysis of our proposed method's output w.r.t several baseline outputs. Model outputs are shown for both success and failure cases for our model. **0:** Not Hateful and **1**: Hateful

	Ours	ϵ-ball (0.1)	CLIP	Integrated Gradient
Relevance	**3.32**	2.65	2.77	2.51
Exhaustiveness	**3.30**	2.58	2.69	2.44

Table 3: Human evaluation results based on 200 samples, evaluated by five annotators across Relevance and Exhaustiveness.

sen to rate the generated explanations (a randomly chosen set of 200 examples) on a scale of 1-5 in the 5 point Likert scale. The inter-rater agreement (Cohen kappa score [Cohen, 1960]) for all the settings is more than 0.7, indicating fair agreement. For both exhaustiveness and relatedness, our methods achieve the best performance as observed from Table 3, followed by ϵ-ball, CLIP, and lastly Integrated Gradient.



01276 13875 91768 98724

Figure 5: Memes corresponding to the examples in Table 2

4.3 Interpreting Model from Retrieved Keywords

In our analysis, we employ sampling from GPT-2 for both keyword extraction and prediction to enhance interpretability. Our focus is on understanding the model's decision-making process in meme explanation and assessing offensiveness, leading us to prioritize interpretability over utilizing more advanced language models like GPT-3, which may generate unfaithful yet plausible explanation [Filippova, 2020; Maynez *et al.*, 2020].

For the correctly classified memes (with IDs 01276, and 98724), our proposed approach (Top-K=3500 with $\epsilon = 0.1$ and other filters enabled) provides a relevant and exhaustive set of keywords for the input meme which may adequately represent the correct model prediction obtained. These explanations are also intuitive and help us to clarify that the model is not relying on spurious correlations to predict its decision for that particular meme. For other variations of our proposed methods and the baseline method, we observe the quality of the retrieved keywords seems arbitrary to the input meme. Thus, they do not adequately reflect the reason based on which the model might have predicted the correct label.

Even though the CLIP retrieves semantically relevant tokens, they are not exhaustive and often repetitive.

From meme ID 91768, we observe that the model predicts the meme as offensive even though it is a funny meme about Hitler. Due to the presence of Hitler's face, the model classifies it as offensive, which may be correctly illustrated by the retrieved keywords using our method. The baseline performs poorly and the variations of our method yield retrieved keywords that are either repetitive or not very semantically relevant to the input meme. Another example is shown for meme ID 13875, where the the model predicted an offensive meme as normal. The prediction appears to be influenced by the presence of the word 'cat,' which the model uses as a determining factor. This phenomenon can be attributed to the model's lack of exposure to relevant memes during training, resulting in an inability to recognize the underlying racism.

5 Conclusion

Classifying hateful content on social media and generating explanations for moderation is crucial. Existing interpretability methods operate on the input space, making it difficult to explain content with hidden meanings. Our work not only identifies hateful content that requires moderation but also provides explanations for hidden meanings. This improves the efficiency of uncovering hidden meanings in memes and clarifies the model's decision-making process. It also helps identify model bias, as shown in qualitative evaluations. Our method outperforms various baseline models in both automated and manual evaluations and is applicable to social media, with potential real-life impact. Additionally, our task-agnostic method can be applied to various visual-linguistic tasks (e.g., Visual Question Answering, Visual NLI), presenting an important challenge for future studies.

Acknowledgements

Dibyanayan Bandyopadhyay and Baban Gain acknowledge support from the Prime Minister Research Fellowship (PMRF). The authors thank valuable feedback from Samrat Mukherjee and comments from anonymous reviewers.

References

[Bandyopadhyay *et al.*, 2023] Dibyanayan Bandyopadhyay, Gitanjali Kumari, Asif Ekbal, Santanu Pal, Arindam Chatterjee, and Vinutha BN. A knowledge infusion based multitasking system for sarcasm detection in meme. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part I*, page 101–117, Berlin, Heidelberg, 2023. Springer-Verlag.

[Caselli *et al.*, 2021] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. HateBERT: Retraining BERT for abusive language detection in English. In Aida Mostafazadeh Davani, Douwe Kiela, Mathias Lambert, Bertie Vidgen, Vinodkumar Prabhakaran, and Zeerak Waseem, editors, *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online, August 2021. Association for Computational Linguistics.

[Chen and Pan, 2022] Yuyang Chen and Feng Pan. Multimodal detection of hateful memes by applying a vision-language pre-training model. *PLOS ONE*, 17(9):1–12, 09 2022.

[Cohen, 1960] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[Desai *et al.*, 2021] Poorav Desai, Tanmoy Chakraborty, and Md Shad Akhtar. Nice perfume. how long did you marinate in it? multimodal sarcasm explanation, 2021.

[DeYoung *et al.*, 2020] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics.

[Du *et al.*, 2022] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.

[Filippova, 2020] Katja Filippova. Controlled hallucinations: Learning to generate faithfully from noisy data. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 864–870, Online, November 2020. Association for Computational Linguistics.

[Ganguly *et al.*, 2017] Debashis Ganguly, Mohammad H. Mofrad, and Adriana Kovashka. Detecting sexually provocative images. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 660–668, 2017.

[Guo *et al.*, 2019] Hao Guo, Kang Zheng, Xiaochuan Fan, Hongkai Yu, and Song Wang. Visual attention consistency under image transforms for multi-label image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Hase *et al.*, 2020] Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online, November 2020. Association for Computational Linguistics.

[Jang *et al.*, 2017] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2017.

[Kayser *et al.*, 2021] Maxime Kayser, Oana-Maria Camburu, Leonard Salewski, Cornelius Emde, Virginie Do, Zeynep Akata, and Thomas Lukasiewicz. e-vil: A dataset and benchmark for natural language explanations in vision-language tasks, 2021.

[Kiela *et al.*, 2021] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes, 2021.

[Kingma and Ba, 2017] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[Koh *et al.*, 2020] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020.

[Li *et al.*, 2023] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[Liu *et al.*, 2023a] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[Liu *et al.*, 2023b] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[Lundberg and Lee, 2017a] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[Lundberg and Lee, 2017b] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[Maynez *et al.*, 2020] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online, July 2020. Association for Computational Linguistics.

[Narang *et al.*, 2020] Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. Wt5?! training text-to-text models to explain their predictions, 2020.

[Radford *et al.*, 2019] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[Roberts *et al.*, 2012] Kirk Roberts, Michael A. Roach, Joseph Johnson, Josh Guthrie, and Sanda M. Harabagiu. EmpaTweet: Annotating and detecting emotions on Twitter. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3806–3813, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[Sammani *et al.*, 2022] Fawaz Sammani, Tanmoy Mukherjee, and Nikos Deligiannis. Nlx-gpt: A model for natural language explanations in vision and vision-language tasks, 2022.

[Sarkar *et al.*, 2021] Diptanu Sarkar, Marcos Zampieri, Tharindu Ranasinghe, and Alexander Ororbia. fBERT: A neural transformer for identifying offensive content. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1792–1798, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[Selvaraju *et al.*, 2019] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019.

[Sharma *et al.*, 2020] Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. SemEval-2020 task 8: Memotion analysis- the visuolingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online), December 2020. International Committee for Computational Linguistics.

[Sharma *et al.*, 2022] Shivam Sharma, Siddhant Agarwal, Tharun Suresh, Preslav Nakov, Md. Shad Akhtar, and Tanmoy Chakraborty. What do you meme? generating explanations for visual semantic role labelling in memes, 2022.

[Shrikumar *et al.*, 2017] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences, 2017.

[Sundararajan *et al.*, 2017] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[Suryawanshi *et al.*, 2020] Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France, May 2020. European Language Resources Association (ELRA).

[Van Hee *et al.*, 2018] Cynthia Van Hee, Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, and Véronique Hoste. Automatic detection of cyberbullying in social media text. *PLOS ONE*, 13(10):1–22, 10 2018.

[Wang *et al.*, 2022] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework, 2022.

[Waseem and Hovy, 2016] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California, June 2016. Association for Computational Linguistics.

[Yang *et al.*, 2022] Chuanpeng Yang, Fuqing Zhu, Guihua Liu, Jizhong Han, and Songlin Hu. Multimodal hate speech detection via cross-domain knowledge transfer. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4505–4514, New York, NY, USA, 2022. Association for Computing Machinery.

[Zhu *et al.*, 2023] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.