MemHateCaptioning: Enhancing Hate Speech Detection in Memes with Context-Aware Captioning and Chain-of-Thought

Rishik Sood
University of Technology Sydney
School of Computer Science
Ultimo, NSW, Australia

Ali Anaissi
University of Technology Sydney
TD School
Ultimo, NSW, Australia

Weidong Huang
University of Technology Sydney
TD School
Ultimo, NSW, Australia

Ali Braytee
University of Technology Sydney
School of Computer Science
Ultimo, NSW, Australia

Abstract

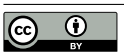
Hate speech has become increasingly prevalent on social media, with memes presenting a unique challenge due to their multimodal nature, combining text, images, and often subtle cultural cues that spread hate online. Several hate speech detection models have recently been proposed to identify hate in multimodal memes. However, these methods may suffer from issues like context ambiguity, subtle visual cues, and multimodal complexity. Generating accurate captions from memes while considering the context, images, symbols, and text can help capture the intended meaning and, consequently, improve the accuracy of hate speech detection. This study introduces MemHateCaptioning, a framework designed to generate clear, human-like explanations to contextualize why a meme is flagged as hateful. MemHateCaptioning leverages recent advancements in vision-language models (VLMs) and large language models (LLMs), integrating ClipCap for image captioning, BLIP for language-image pretraining, and T5 for explanation generation. The framework incorporates Chain-of-Thought (CoT) prompting to enhance interpretability, enabling the model to break down complex reasoning step by step, which helps in comprehending the subtle interplay between text and images in hateful content. MemHateCaptioning is evaluated on the HatReD dataset and demonstrates strong performance, achieving higher BLEU and ROUGE-L scores compared to existing models. It also effectively reduces issues such as hallucinations and context misinterpretation by providing detailed, context-aware explanations.

CCS Concepts

• **Computing methodologies** → *Computer vision*; **Machine learning**.

Keywords

Hate speech detection, memes, AI captioning methods, multimodal models



WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3718385

ACM Reference Format:
Rishik Sood, Ali Anaissi, Weidong Huang, and Ali Braytee. 2025. MemHateCaptioning: Enhancing Hate Speech Detection in Memes with Context-Aware Captioning and Chain-of-Thought. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3701716.3718385

Disclaimer: This paper contains discriminatory content that may be disturbing to some readers.

1 Introduction

Hate speech is characterized by language that incites violence, discrimination, and/or hostility against individuals or groups based on their race, ethnicity, religion, etc. It has become a significant challenge in the digital age, where access to social media and resources is easier than ever. Social media is often abused by people to spread hate speech through text, images, and videos. The rapid spread of these hateful memes creates several social problems, such as exacerbating racism and circulating disinformation. It is of critical importance to identify, understand, and mitigate hate speech across all these mediums. However, this is a complex task due to the varying interpretations of content and contextual subtleties. To truly understand a meme, it is essential to grasp the corresponding background knowledge, which includes but is not limited to, political issues, newsworthy events, cultural references, and historical events [5]. Traditional approaches to hate speech detection have predominantly focused on unimodal (usually text-based) analysis. Early work in this field employed machine learning classifiers trained on textual features to identify explicit hate speech. For example, [6] used text-based features like unigrams and bigrams to detect hate speech on Twitter, while [19] differentiated hate speech from offensive language using linguistic markers. These early studies focused on textual data and could not interpret complex multimodal content.

With the advent of pre-trained large language models (PLMs) such as BERT, GPT, and T5, the field has witnessed significant advances in natural language processing (NLP) capabilities. These models, when fine-tuned on hate speech detection tasks, have shown considerable success in interpreting nuanced text-based hate speech by leveraging contextual embeddings [8] [26]. These advancements in PLMs have highlighted their potential for hate

speech detection, yet they still fall short when tasked with interpreting complex multimodal content, such as memes. Recent developments in vision-language models (VLMs), such as CLIP, BLIP, and LLaVa, have opened new possibilities for multimodal hate speech detection. VLMs are designed to process and align visual and textual data, making them well-suited for tasks that require an understanding of both image and text [18]. CLIP, for instance, demonstrated impressive zero-shot capabilities, achieving notable accuracy in detecting hate speech in multimodal content without task-specific training [2]. BLIP and BLIP-2 have further enhanced this capability by integrating a multimodal mixture of encoder-decoder techniques and employing innovative training methods, such as CapFilt, which generates and filters synthetic captions to improve image-text alignment. These VLMs show promise in the domain of hate speech detection, especially when combined with models for captioning and explanation generation, which add interpretive depth to detection results. Chain-of-thought (CoT) prompting is another critical advancement that has implications for hate speech detection. CoT prompting allows models to "think" step-by-step, breaking down complex reasoning processes into sequential steps. In the context of hate speech detection, CoT prompting enables models to not only identify hateful content but also explain the rationale behind their classifications. For instance, Huang et al. (2023) applied CoT prompting to generate explanations for implicit hate speech, achieving improvements in both informativeness and clarity [13]. However, there remains a gap in using CoT prompting for multimodal hate speech classification, where both visual and textual reasoning are required to produce meaningful explanations.

Memes are a multimodal form of data that often combines image-based and text-based elements. Several works to detect hate in memes, such as using the CLIP model, have shown excellent results even with zero-shot training [2]. While current approaches can flag memes as hateful on social media platforms, methods that can generate semantically aware captions explaining why a meme was classified as hateful are very limited. Our study aims to bridge this gap by developing MemHateCaptioning, a framework that leverages VLMs, PLMs, and CoT prompting to generate explanations for hateful memes. This approach builds on prior work by integrating advanced models for visual and textual analysis and applying CoT prompting to facilitate step-by-step reasoning. MemHateCaptioning utilizes several models, including ClipCap for image captioning, BLIP as a vision-language model, and T5 for explanation generation, to construct comprehensive, contextually accurate interpretations of memes. By focusing on the interpretive aspect, MemHateCaptioning addresses the need for transparency and accountability in automated hate speech detection, particularly in complex multimodal contexts. Our method was evaluated on the HatReD dataset [12], a collection of annotated hateful memes, making it a suitable benchmark for this study.

2 Related Work

2.1 Pre-trained Large Language Models

Pre-trained Large Language Models (PLMs) like GPT [22], BERT [14], and T5 [23] have revolutionized NLP by leveraging self-supervised learning and encoder-decoder architectures to learn contextual representations. These models are fine-tuned for specific tasks, achieving state-of-the-art results. For example, SciBERT [3], fine-tuned on scientific publications, outperformed BERT in tasks like Named Entity Recognition and Text Classification due to its specialized vocabulary. T5 is fine-tuned on the CodeSearchNet dataset [27], improved tasks like bug fixing and code reviews by combining generation and edit-based models. GENBERT [11] addressed numerical reasoning challenges and showed improved performance on mathematical tasks such as the DROP dataset. These studies highlight the versatility of PLMs across various NLP tasks. ClipCap [20], a lightweight image captioning model, utilizes Contrastive Language-Image Pretraining (CLIP) for visual encoding and GPT2 for generating captions.

2.2 Unimodal Classification of Hate Speech

Several works, such as [6], used machine learning models on text-based features like unigrams and sentiment analysis to study hate speech on Twitter. They found that hate speech clusters around specific events. [1] improved classification accuracy on an Indonesian Twitter dataset using GloVe embeddings and Random Forest. A study [19] focused on distinguishing hate speech from offensive content, achieving 78% accuracy with character 4-grams and SVM. With the rise of PLMs, recent work has shifted to fine-tuning models. HateBERT is developed by fine-tuning BERT on Reddit data, outperforming generic BERT [8]. Wullach et al. (2021) used GPT2 for data augmentation, enhancing PLM performance by 5-10% in F1 scores [26]. A method combined ELMo, BERT, and CNN, improving accuracy and F1 scores on a hate speech detection dataset [29].

2.3 Multimodal Classification of Hateful Memes

Multimodal analysis of hate speech, particularly in memes, is crucial due to the need to capture both visual and textual clues. Relying solely on text often results in poor classification performance. A study proposed an architecture that combines object tags and image captions, converting them into text embeddings with BERT for improved classification using models like Support Vector Classifier (SVC) and Gradient Boosting Decision Trees (GBDT) [4]. Another study introduced the LUMEN model, a multimodal, multitask transformer-based system that labels entities (hero, villain, victim) and generates natural language explanations [25]. LUMEN integrates pre-trained models like ViT, DeBERTa, and T5, optimizing tasks with a joint loss function. Chhabra & Vishwakarma (2021) showed success using the MSKAV module for visual features and KDAC for textual features, achieving 87.5% accuracy [9]. Arya et al. (2024) demonstrated the effectiveness of the CLIP model with zero-shot learning, achieving 87.42% accuracy, making it ideal for online content moderation [2]. Pro-Cap [7] aimed to solve the classification problem by using probe captioning method and employed 6 questions to generate additional context about the image. This research showed good results however, they also discussed that using all questions for all images could also be a limitation.

2.4 Hateful Meme Explanation Generation

Multimodal approaches have shown strong results in classifying hateful memes, even with zero-shot training. However, this research

addresses the gap in providing semantically aware explanations for why a meme is considered hateful, crucial for social media moderation. A study used a probe-captioning method with BLIP-2 and PromptHate to generate annotations based on questions about image content, achieving 80.87% accuracy on the FHM dataset, though captions sometimes included irrelevant answers [7]. HatReD dataset [12] is created with human-annotated captions, aiming to improve explanation generation, through baseline models like T5, VisualBERT, and RoBERTa.

3 Our Proposed Framework

The proposed method, MemHateCaptioning, leverages a combination of VLMs and CoT prompting techniques to generate context-aware captions for memes. MemHateCaptioning utilizes pre-trained models such as ClipCap for image captioning, BLIP for image-language modeling, EasyOCR for OCR extraction, and the Google Web Detection API for entity extraction, ensuring complete image-text understanding. T5 is used for generating explanatory text, making the approach suitable for multimodal analysis. As shown in Fig. 1, MemHateCaptioning consists of two modules: Annotation Generator and Explanation Generator.

3.1 Annotation Generator Module

This module creates the input sentence for the explanation generation module. It is crucial to capture all relevant features, entities, and contexts from the meme to understand its hateful nature. Failure to capture important features such as race, religion, or others, or to extract OCR data, could lead to misjudgments by the explanation module, resulting in invalid captions. Our annotation generator uses four techniques: ClipCap (image captioning), Google Vision Web Detection API (entity extraction), EasyOCR (OCR extraction), and CoT Annotation Generator. These components are shown in Fig. 1. ClipCap [20] generates a general description of the image to capture basic visual elements. While the caption may not directly explain the nature of hate in the meme, it helps identify benign confounders, as defined by Das et al. (2020), in cases where the extracted OCR and image description conflict. Smaller VLMs often cannot store all contextual information about the people in the image, entities, and demographics, so the Google Vision Web Detection API is used here. Capturing textual information is essential for this multimodal task, as it provides the second half of the meme's context. EasyOCR, a state-of-the-art tool for OCR extraction, was employed for this purpose. The novelty of this research lies in the creation of the CoT annotation generator, which generates additional context to enhance the explanation model. As mentioned earlier, BLIP [15] has shown strong performance in Visual Question Answering (VQA). However, a limitation of BLIP is that the generated text is often short and generic, making it insufficient for generating a complete step-by-step explanation of the meme. To address this, we employed a set of six questions provided by ProCap [7] designed to accurately describe the people depicted in the meme as shown in Fig. 2. These questions provide high-quality text that includes information about race, religion, country, disability, and any mention of animals in the meme. These additional captions help the explanation module focus on the relevant context. CoT prompting was incorporated by adding "Think step by step" to the questions, encouraging the model to reason sequentially before generating an output. An example of final input sequences is provided in Fig. 3.

3.2 Explanation Generator Module

The explanation module is the second part of MemHateCaptioning. It is responsible for converting the raw multimodal input sequence, generated by the annotation generator module, into contextually accurate and semantically aware captions. It combines the image caption, extracted OCR, entity descriptions, and CoT annotations to form a comprehensive understanding of the meme's content. This module primarily uses a T5 model [23] for conditional generation. The explanation generation task is framed as a supervised learning problem, with the input being the concatenation of outputs from the previous models, which contain all the necessary contextual information extracted from the meme. The ground truth outputs are provided by the HatReD dataset, which contains human-annotated explanations for hateful memes. This module is crucial for bridging the gap between a simple classification model and a human-like understanding of the meme. Fig. 4 shows an example of the output generated from the input-generated annotations in Fig. 3.

4 Experiments

4.1 Dataset

In this study, we evaluate the methods on the Hateful Meme with Reasons Dataset (HatReD) [12], which includes a total of 3,304 annotated explanations corresponding to 3,228 hateful memes. Some memes feature multiple annotations because they target more than one social group. The length of the explanations varies, with the shortest being 5 words, an average length of 13.62 words, and the longest reaching 31 words.

4.2 Experimental Settings

To run experiments, we use Nvidia L4 GPUs with 24GB of RAM. We also utilize pre-trained language models, such as T5, BLIP, and others, from the HuggingFace library. To ensure that the model architecture worked consistently across the dataset, it was trained over 5 different random seeds, and the reported results are the average of the outcomes from the different combinations. Furthermore, we use the AdamW optimiser with the following parameters: the weight decay (λ) is set to 0.1, the learning rate (η) is set to 10^{-4}, and epsilon (ε) is set to 10^{-8}. Here, (η) controls the step size for weight updates, (ε) ensures numerical stability to prevent division by 0 while updating parameters, and (λ) acts as a coefficient to control the magnitude of weight decay to regularize the model and prevent overfitting. Additionally, different numbers of epochs were tested for the model, ranging from 5 to 50 in increments of 5. Extensive experimentation revealed that the results stabilized after 12–15 epochs. The T5 model uses the regular cross-entropy loss function as follows

$$L_{CE} = -\sum_{i=1}^{N}\sum_{j=1}^{l} \log p_\theta \left(r_j^i \middle| x_r^i, x_v^i, r_1^i, \ldots, r_{j-1}^i \right) \quad (1)$$

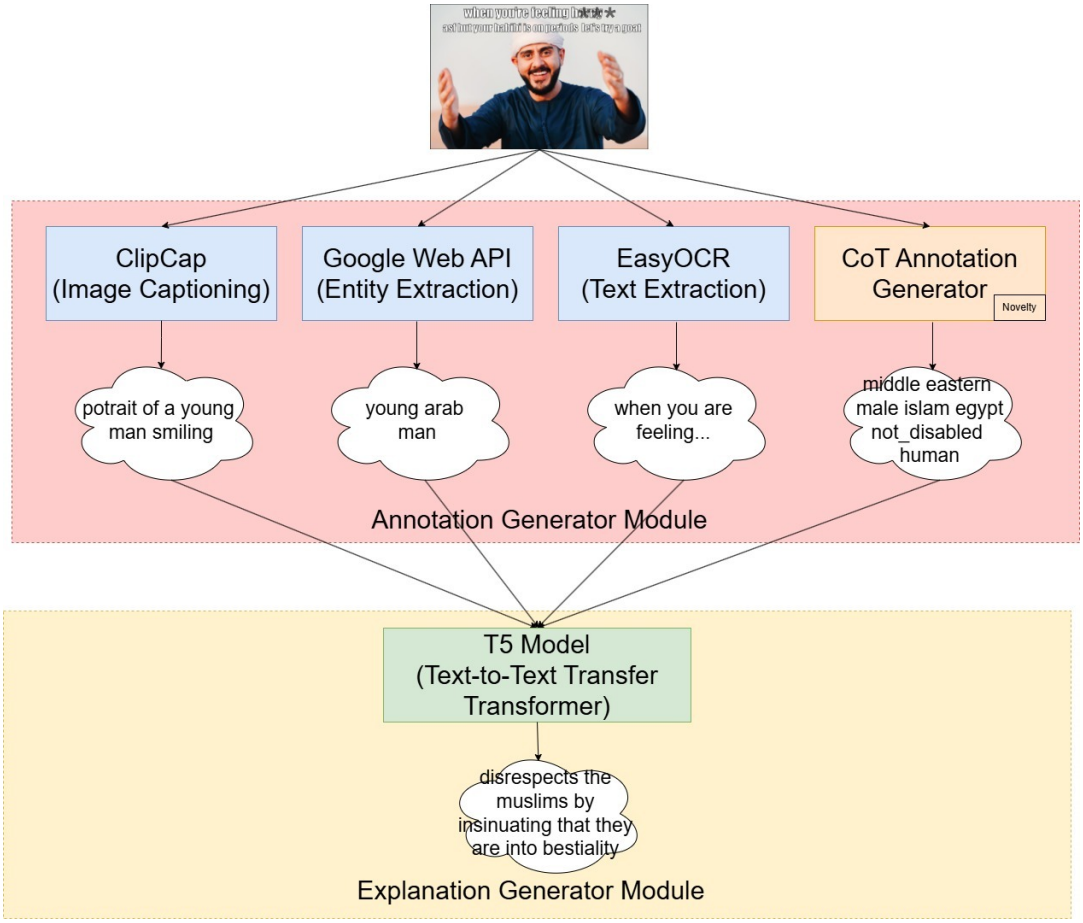


Figure 1: MemHateCaptioning framework architecture

1. What is the race of the person in the meme? Think step by step.
2. What is the gender of the person in the meme? Think step by step.
3. What is the religion of the person in the meme? Think step by step.
4. What country does the person in the meme come from? Think step by step.
5. Are there disabled people in the meme? Think step by step.
6. What animal is in the meme? Think step by step.

Figure 2: The generated questions in CoT Annotation generator component

where L_{CE} represents the cross entropy loss, θ is the model's trainable parameter, x_T^i and x_V^i are textual and visual inputs from the meme, r^i is the annotated reason, and r_j^i is its j-th token.

4.3 Evaluation Metrics

There are two common approaches N-gram and Embedding-based metrics. N-gram-based measures evaluate the similarity between reference and candidate sentences using word overlaps. Common evaluation metrics that involve N-grams include:

- BLEU [21] is a popular metric that calculates precision by comparing matching N-grams (1-grams, 2-grams, 3-grams, and 4-grams) in the candidate sentence to the total number of N-grams. It also includes a Brevity Penalty, which discourages short sentences that only match keywords. The BLEU score ranges from 0 to 1, with higher values indicating greater similarity to the reference text and is defined as follows

'woman reading a book Scholarship Graduate Pharmacy Aptitude Test Human resources Graduate University Student College Career Scholarship i am not racist i just don't like brown sugar indian female female hindu india no_disability human'
'pig in a sty Hampshire pig Pig's ear domestic pig i'm going home i'm russian unk unk unk unk no_disability pig'
'asian man in a crowd Kim Yo-jong Syria Pyongyang South Korea Russia 2018 North Korea\u2013United States Singapore Summit United States Supreme Leader of North Korea 2019 North Korean parliamentary election Kim Jong-un southeast asian male male christian china no_disability human'

Figure 3: Examples of the generated annotations

'insults the blacks by suggesting that the girl does not like brown sugar.'
'insults the russians by calling them pigs.'
'mocks the chinese for eating cats.'

Figure 4: Examples of the generated outputs

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where p_n is the precision for n-grams, w_n is the weight for each n-gram precision, and BP is the Brevity Penalty described as

$$BP = \begin{cases} 1 & \text{if candidate}_{\text{length}} > \text{reference}_{\text{length}} \\ e^{\left(1-\frac{\text{reference}_{\text{length}}}{\text{candidate}_{\text{length}}}\right)} & \text{if candidate}_{\text{length}} \leq \text{reference}_{\text{length}} \end{cases}$$

- ROUGE: while BLEU measures word-based similarity, sentence-level similarity also needs to be assessed. ROUGE-L [24] evaluates the longest matching sequence of words, or Longest Common Subsequence (LCS), between the candidate and reference text. It calculates recall, precision, and F1 score as

$$\text{Precision} = \frac{\text{LCS}}{\text{candidate}_{\text{length}}}$$

$$\text{Recall} = \frac{\text{LCS}}{\text{reference}_{\text{length}}}$$

$$\text{ROUGE (F1 Score)} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

This metric has a small advantage over BLEU by not requiring exact matches, but it still lacks semantic understanding, and combining BLEU and ROUGE-L scores provides a broader view of n-gram and sequence similarity as

$$H.\text{Mean} = \frac{2 \times \text{BLEU} \times \text{ROUGE}}{\text{BLEU} + \text{ROUGE}}$$

The main limitation of previous metrics is their inability to capture semantic understanding. Embedding-based metrics such as BERTScore [28] address this by using BERT embeddings to evaluate the quality of the generated text, capturing semantic meaning and comparing it to human annotations through cosine similarity of contextual embeddings. We used BERT-P (Precision), BERT-R (Recall), and BERT-F (F1 Score) for evaluation.

4.4 Compared methods

For text-based models, we use T5 [23], RoBERTa [17], and GPT-2 [22]. Since GPT-2 is used as a decoder-only model, we pair it with RoBERTa, which acts as the encoder. For vision-language models, we evaluate models such as [10] and VisualBERT [16]. Since VisualBERT is an encoder-only model, we incorporate RoBERTa and GPT-2 as decoders in different configurations. These evaluations serve as baseline comparisons for the new task of explaining hateful memes.

5 Results

The results described in Table 1 provide a performance comparison across different PLMs for the hateful meme explanation task, using both text-only and multimodal models. In terms of N-gram-based metrics, our method outperforms all other models. Specifically, the BLEU score for our method is 0.212, representing a 7% improvement over the best text-only model, T5 (0.190). For ROUGE-L, our method achieves a score of 0.401, surpassing T5 (0.392) by 2%, which indicates a better ability to capture the longest matching subsequences. The harmonic mean (H.Mean) also shows the superiority of our model, with a score of 0.277, an 8% improvement over T5 (0.256). In embedding-based metrics, our method consistently leads, with a BERT Precision (BERT-P) score of 0.497, which is 2% higher than T5 (0.485), showing better semantic alignment with human annotations. For BERT Recall (BERT-R), our model scored 0.472, slightly lower than T5 (0.473), but still higher than VL-T5 and VisualBERT + GPT2 (0.409, 0.342). In terms of the BERT F1-score (BERT-F), our method is slightly lower than T5 (0.474 vs. 0.479) but higher than

	RoBERTa + GPT2 (text-only)	T5 (text-only)	VisualBERT + GPT2 (multimodal)	VL-T5 (multimodal)	Our method (multimodal)
N-Gram based metrics					
BLEU	0.068	0.190	0.065	0.180	0.212 (+7%)
ROUGE-L	0.222	0.392	0.219	0.378	0.401 (+2%)
H.Mean	0.104	0.256	0.100	0.244	0.277 (+8%)
Embedding based metrics					
BERT-P	0.112	0.485	0.100	0.472	0.497 (+2%)
BERT-R	0.327	0.473	0.342	0.409	0.472 (-0.2%)
BERT-F	0.218	0.479	0.219	0.446	0.474 (-1.0%)

Table 1: Performance comparison across different PLM models

 **Example 1**	**Ground truth**: ridicules the blacks as inferior by joking about not liking brown sugar. **T5**: insults the blacks by using the word n****r to mock brown sugar. **Our method**: insults the blacks by suggesting that the girl does not like brown sugar.
 **Example 2**	**Ground truth**: dehumanizes the russians as pigs. **T5**: insults the russians by suggesting that they are slaves. **Our method**: insults the russians by calling them pigs.
 **Example 3**	**Ground truth**: mocks the asians for being uncivilised as they eat cats. **T5**: ridicules the chinese by suggesting that they eat cats. **Our method**: mocks the chinese for eating cats.

Table 2: Generated captions for hateful memes in the HatReD dataset

the other models. Overall, these results demonstrate that our multimodal method outperforms most of the text-only and multimodal methods across various metrics, making it a robust solution for the hateful meme explanation task.

 **Example 1**	**Ground truth**: disrespects the blacks by insinuating that they are into beastiality. **Our method**: mocks the people with physical disabilities for being unable to see the difference between an elf and a person. **Explanation**: This is a complicated meme which requires the model to relate to the racial slur which the model. Instead the model picks the context that the meme is related to people not being able to see i.e. people with disabilities and forms a caption for the meme.
 **Example 2**	**Ground truth**: disrespects the whites by suggesting that they should be killed. **Our method**: dehumanizes the whites as murderers. **Explanation**: The model misinterprets the message of the meme due to the mention of white people and murder and labels them as murderers whereas they are the victims here.

Table 3: Erroneous generated captions

5.1 Case studies

In this section, we present multiple examples for human evaluation. The captions generated by our method are compared with the ground truth (GT) captions and the best-performing PLM in the experiments, T5. In the first example, as shown in Table 2, the T5 model hallucinates based on the given information. It uses its pretraining and prior context to generate unnecessary text that does not even relate to the meme. On the other hand, our method accurately generates the meme's caption, considering the context and producing an output that is closer to the GT. Additionally, our caption mentions the person's gender in the image, which emphasizes the entity in the meme. In the second example, our method's caption is closer to the GT in terms of context and relevance, as it accurately identifies the intended insult by calling Russians "pigs". In contrast, the T5 model misses the nuance by suggesting they are "slaves", which diverges from the context of dehumanization present in the GT. In the third example, our method's caption is more aligned with the GT, correctly focusing on mocking Asians for eating cats. The T5 model slightly diverges from the GT by emphasizing ridicule rather than mockery and suggesting a broader insult to Chinese culture. However, both methods identify the individuals as Chinese, even though the picture includes North Korea's president.

Table 3 shows two cases where the model generates inaccurate captions. Example 1 is a complicated meme. Our method identifies the disrespect, but it misinterprets the meme. In example 2, a meme inciting violence against white people is misread—the model captures "kill" but wrongly labels them as murderers. These examples highlight the struggle of the model with context and cultural nuances, despite its reasoning being somewhat aligned.

6 Conclusion

The MemHateCaptioning framework advances hate speech detection in multimodal contexts, utilizing VLMs like ClipCap and BLIP, along with CoT prompting, to offer natural language explanations for hateful memes. Our method helps the explanation module better understand the meme by providing a relevant context. Quantitative evaluation metrics like BLEU and ROUGE-L indicate that MemHateCaptioning outperforms baseline models, demonstrating improved alignment with human-like explanations. Through case studies, we observe that MemHateCaptioning mitigates common issues in baseline models, such as hallucinations and incorrect entity associations, by prompting the model to break down its reasoning into smaller steps. These findings suggest that the use of stronger models and CoT prompting can yield better results with more contextual captions. For future work, this research can be extended by incorporating multilingual text, addressing cultural-specific nuances in meme interpretation, and improving model robustness against lower-quality captions in the dataset.

References

[1] Febiana Anistya, Erwin Budi Setiawan, et al. 2021. Hate Speech Detection on Twitter in Indonesia with Feature Expansion Using GloVe. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* 5, 6 (2021), 1044–1051.
[2] Greeshma Arya, Mohammad Kamrul Hasan, Ashish Bagwari, Nurhizam Safie, Shayla Islam, Fatima Rayan Awad Ahmed, Aaishani De, Muhammad Attique Khan, and Taher M Ghazal. 2024. Multimodal Hate Speech Detection in Memes Using Contrastive Language-Image Pre-Training. *IEEE Access* (2024).
[3] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676* (2019).
[4] Aruna Bhat, Vaibhav Vashisht, Vaibhav Raj Sahni, and Sumit Meena. 2023. Hate speech detection using multimodal meme analysis. In *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. IEEE, 1137–1142.

[5] Nanyi Bi, Yi-Ching Huang, Chao-Chun Han, and Jane Yung-jen Hsu. 2023. You Know What I Meme: Enhancing People's Understanding and Awareness of Hateful Memes Using Crowdsourced Explanations. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–27.

[6] P Burnap and ML Williams. 2015. Cyber hate speech on Twitter: An application of machine classification and statistical modeling for policy and decision making. Policy & Internet, 7 (2), 223-242.

[7] Rui Cao, Ming Shan Hee, Adriel Kuek, Wen-Haw Chong, Roy Ka-Wei Lee, and Jing Jiang. 2023. Pro-cap: Leveraging a frozen vision-language model for hateful meme detection. In *Proceedings of the 31st ACM International Conference on Multimedia*. 5244–5252.

[8] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472* (2020).

[9] Anusha Chhabra and Dinesh Kumar Vishwakarma. 2023. Multimodal hate speech detection via multi-scale visual kernels and knowledge distillation architecture. *Engineering Applications of Artificial Intelligence* 126 (2023), 106991.

[10] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*. PMLR, 1931–1942.

[11] Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. *arXiv preprint arXiv:2004.04487* (2020).

[12] Ming Shan Hee, Wen-Haw Chong, and Ka-Wei Roy Lee. 2023. Decoding the Underlying Meaning of Multimodal Hateful Memes. In *32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)*. International Joint Conferences on Artifical Intelligence (IJCAI).

[13] Fan Huang, Haewoon Kwak, and Jisun An. 2023. Chain of explanation: New prompting method to generate quality natural language explanation for implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*. 90–93.

[14] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, Vol. 1. Minneapolis, Minnesota, 2.

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*. PMLR, 12888–12900.

[16] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557* (2019).

[17] Y Liu, M Ott, N Goyal, J Du, M Joshi, D Chen, O Levy, M Lewis, L Zettlemoyer, and V Stoyanov. 1907. RoBERTa: A robustly optimized BERT pretraining approach. arXiv [Preprint](2019). *arXiv preprint arXiv:1907.11692* (1907).

[18] Zhicheng Liu, Ali Braytee, Ali Anaissi, Guifu Zhang, Lingyun Qin, and Junaid Akram. 2024. Ensemble Pretrained Models for Multimodal Sentiment Analysis using Textual and Video Data Fusion. In *Companion Proceedings of the ACM on Web Conference 2024*. 1841–1848.

[19] Shervin Malmasi and Marcos Zampieri. 2017. Detecting hate speech in social media. *arXiv preprint arXiv:1712.06427* (2017).

[20] Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734* (2021).

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[22] Alec Radford. 2018. Improving language understanding by generative pre-training. (2018).

[23] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[24] Lin CY Rouge. 2004. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*, Vol. 5.

[25] Shivam Sharma, Siddhant Agarwal, Tharun Suresh, Preslav Nakov, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. What do you meme? generating explanations for visual semantic role labelling in memes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 9763–9771.

[26] Tomer Wullach, Amir Adler, and Einat Minkov. 2021. Fight fire with fire: Fine-tuning hate detectors using large samples of generated hate speech. *arXiv preprint arXiv:2109.00591* (2021).

[27] Jiyang Zhang, Sheena Panthaplackel, Pengyu Nie, Junyi Jessy Li, and Milos Gligoric. 2022. Coditt5: Pretraining for source code and natural language editing. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. 1–12.

[28] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).

[29] Yanling Zhou, Yanyan Yang, Han Liu, Xiufeng Liu, and Nick Savage. 2020. Deep learning based fusion approach for hate speech detection. *IEEE Access* 8 (2020), 128923–128929.