MULTIMODAL LEARNING FOR HATEFUL MEMES DETECTION

Yi Zhou[1†], *Zhenhao Chen*[2†], *Huiyuan Yang*[3]

[1]IBM, Singapore [2]The University of Maryland, USA [3]Binghamton University, USA
[1]joannezhouyi@gmail.com, [2]zhenhao.chen@marylandsmith.umd.edu, [3]hyang51@binghamton.edu

ABSTRACT

Memes are used for spreading ideas through social networks. Although most memes are created for humor, some memes become hateful under the combination of pictures and text. Automatically detecting hateful memes can help reduce their harmful social impact. Compared to the conventional multimodal tasks, where the visual and textual information is semantically aligned, hateful memes detection is a more challenging task since the image and text in memes are weakly aligned or even irrelevant. Thus it requires the model to have a deep understanding of the content and perform reasoning over multiple modalities. This paper focuses on multimodal hateful memes detection and proposes a novel method incorporating the image captioning process into the memes detection process. We conduct extensive experiments on multimodal meme datasets and illustrate the effectiveness of our approach. Our model achieves promising results on the Hateful Memes Detection Challenge. Our code is made publicly available at GitHub.

***Index Terms*—** Hateful Memes Detection, Multimodal

1. INTRODUCTION

Automatic hateful memes detection is crucial for a good social network environment. Given an image, the multimodal meme detection task is expected to find clues from the sentences in the meme image and associate them with the relevant image regions to reach the final detection. Due to the rich and complicated mixture of visual and textual knowledge in memes, it is hard to identify the implicit knowledge behind the multimodal memes efficiently. Driven by the recent advances in neural networks [1], some works try to detect offensive or misleading content for visual and textual content [2]. However, current methods are still far from mature because of the huge gap between meme images and text content.

Hateful memes detection can be reviewed as a vision-language (VL) task, which has gained much attention in recent years[3, 4]. Specifically, the multimodal memes detection task shares the same spirit as Visual Question Answering (VQA) [5], which predicts the answer base on the image and question input. VQA has been boosted by the advances of image understanding and natural language processing (NLP) [6, 7]. Recently, VQA methods follow the multimodal fusion framework that encodes the image and sentence and then fuses them for answer prediction, during which the given image is encoded with a Convolutional Neural Network (CNN) based encoder, and the sentence is encoded with a Recurrent Neural Network (RNN) based encoder. With the advancement of Transformer [8] network, many recent works incorporate multi-head self-attention mechanisms into their methods [9], and achieve a considerable jump in performances. The transformer's core part lies in the self-attention mechanism, which transforms input features into contextualized representations with multi-head attention, making it an excellent framework to share information between different modalities.

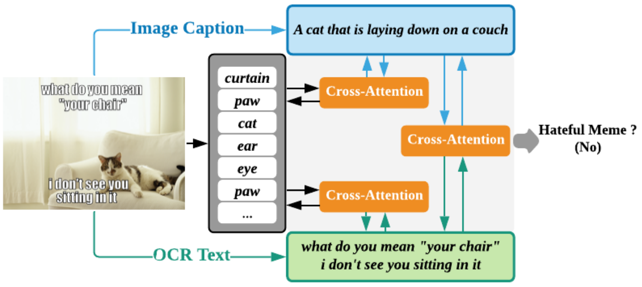


Image Caption
A cat that is laying down on a couch
what do you mean "your chair"
i don't see you sitting in it
curtain
paw
cat
ear
eye
paw
...
Cross-Attention
Cross-Attention
Cross-Attention
Hateful Meme ?
(No)
OCR Text
what do you mean "your chair"
i don't see you sitting in it

Fig. 1. Illustration of our proposed method. It consists of an image captioner, an object detector, a triplet-relation network, and a classifier. The proposed triplet-relation network models the triplet relationships among caption, objects, and Optical Character Recognition (OCR) sentences, adopting the cross-attention model to learn the more discriminative features from cross-modal embeddings.

Although a lot of multimodal learning works focus on the fusion of visual and language features [10], it is difficult to directly apply the multimodal fusion method to memes detection, as it focuses more on the reasoning between visual and textual modalities. Modeling the relationships between multiple modalities and exploring the implicit meaning behind them is still a challenging task. Take Fig. 1 as an example, a cat is lying down on a couch, but the sentences in the image are not correlated to the picture. The misaligned semantic information between visual and textual features adds significant

†Equal contribution.

978-1-6654-4989-2/21/$31.00 ©2021 IEEE

challenges for memes detection. In some cases, even a human being finds it difficult to identity.

Considering the discrepancy between the visual and textual information in memes, we propose a novel method that uses the image caption as a bridge between image content and OCR sentences. To summarize, our contributions in this paper are twofold. First, we design a Triplet-Relation Network (TRN) that enhances the multimodal relationship modeling between visual regions and sentences. Second, we conduct extensive experiments on the meme detection dataset, which requires highly complex reasoning between image content and sentences. Experimental evaluation of our approach shows significant improvements in memes detection over the baselines. Our best model also ranks high in the hateful memes detection challenge leaderboard.

2. RELATED WORKS

Hate Speech Detection. Hate speech is a broadly studied topic in network science and NLP [11]. Detecting hate information in language has been studied for a long time. One of the main focuses of hate speech with diverse targets has appeared in social networks [12]. The general steps for hate speech detection are to obtain a sentence embedding and then feed the embedding into a binary classifier for prediction of hate speech. To facilitate the study of hate speech detection, several language-based hate speech detection datasets have been released [13]. However, hate speech detection has shown to be challenging and subject to undesired bias [14], notably the definition of hate speech [15], which brings the challenges for the machine to understand and detect them. In addition to the single modality-based hate speech detection, researchers also explore multimodal speech detection [16, 17].

Visual Question Answering. Similar to multimodal meme detection, the target of VQA is to answer a question given an image. Most current approaches focus on learning the joint embedding between the images and questions. More specifically, the image and question are passed independently through the image encoder and sentence encoder. The extracted image and question features are then fused to predict the answer. In general, those questions relate to the given images and the challenge of VQA lies in how to reason over the image based on the question. Attention plays a crucial role in the improved performance of VQA [18]. In [18], they first introduce soft and hard attention mechanisms, which model the interactions between image regions and words according to their semantic meaning. Inspired by the huge success of transformer [19] in neural machine translation (NMT), some works have been proposed to use the transformer to learn cross-modality encoder representations[9, 20].

Even though the considerable success of vision-language pre-training in VQA, hateful memes detection is still hard due to its special characteristics. For example, the textual descriptions shown in Fig. 1 are not semantically aligned with the visual content in the image. Applying the methods in VQA to memes detection will encounter some issues. First, unlike questions in VQA that are mostly based on image content, the sentences in a meme can be misaligned with the image. Second, it is difficult to predict the results from visual modality or textual modality directly. The model needs to understand the implicit relationships between image contents and sentences. Our work adopts image captioning as the bridge which connects the visual information and textual information. Besides, it models their relationships with a novel triplet-relation network.

3. METHOD

Fig. 2 shows our proposed framework. The whole system consists of two training paths: image captioning and multimodal learning. The first path (top part) is identical to the image captioning that maps the image to sentences. The second training path (the bottom part) detects the label from the generated caption, OCR sentences, and detected objects. In the following, we describe our framework in detail.

3.1. Input Embeddings

3.1.1. Sentence Embedding

The motivation for generating image caption for each meme is that image captioning provides a good understanding of image content. The goal of image captioning task is to generate a sentence S^c that describes the content of the image I. In particular, we first extract the image feature $\mathbf{f}_I$ with an image encoder $P(\mathbf{f}_I|I)$, and then decode the visual feature into a sentence with a sentence decoder $P(S|\mathbf{f}_I)$. More formally, the image captioning model $P(S|I)$ can be formulated as $P(\mathbf{f}_I|I)P(S|\mathbf{f}_I)$. During inference, we formulate the decoding process as:

$$\hat{S}^c = \arg\max_{\mathcal{S}} P(S|\mathbf{f}_I)P(\mathbf{f}_I|I) \tag{1}$$

where $\hat{S}^c$ is the predicted image description, $\mathcal{S}$ denotes the vocabulary. The most common loss function to train Eq. 1 is to minimize the negative probability of the target caption words with cross-entropy loss.

As shown in Fig. 1, our model has two kinds of textual inputs: image caption S^c and OCR sentence S^o. The predicted caption $\hat{S}^c$ is first split into words $\{\hat{w}_1^c, \ldots, \hat{w}_{N_C}^c\}$ by WordPiece tokenizer [21], where N_C is the number of words. Following [20, 9], the textual feature is composed of word embedding, segment embedding, and position embedding:

$$\hat{\mathbf{w}}_i^c = \mathrm{LN}\big(f_{\text{WordEmb}}(\hat{w}_i^c) + f_{\text{SegEmb}}(\hat{w}_i^c) + f_{\text{PosEmb}}(i)\big) \tag{2}$$

where $\hat{\mathbf{w}}_i^c \in \mathbb{R}^{d_w}$ is the word-level feature, LN represents the layer normalization, $f_{\text{WordEmb}}(\cdot)$, $f_{\text{SegEmb}}(\cdot)$, and $f_{\text{PosEmb}}(\cdot)$ are the embedding functions.

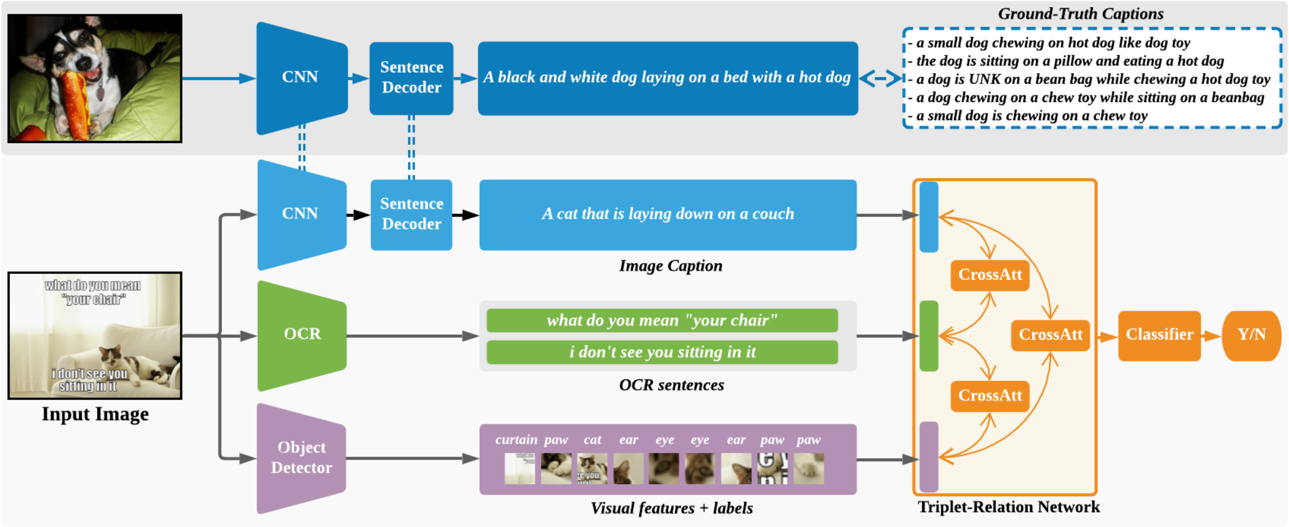

Ground-Truth Captions
- a small dog chewing on hot dog like dog toy
- the dog is sitting on a pillow and eating a hot dog
- a dog is UNK on a bean bag while chewing a hot dog toy
- a dog chewing on a chew toy while sitting on a beanbag
- a small dog is chewing on a chew toy
CNN
Sentence Decoder
A black and white dog laying on a bed with a hot dog
CNN
Sentence Decoder
A cat that is laying down on a couch
Image Caption
what do you mean "your chair"
i don't see you sitting in it
Input Image
OCR
what do you mean "your chair"
i don't see you sitting in it
OCR sentences
Object Detector
curtain paw cat ear eye eye ear paw paw
Visual features + labels
CrossAtt
CrossAtt
CrossAtt
Classifier
Y/N
Triplet-Relation Network

Fig. 2. Overview of our proposed hateful memes detection framework. It consists of three components: image captioner, object detector, and triplet-relation network. The top branch shows the training process of the image captioning model on image-caption pairs. The bottom part is meme detection. It takes image caption, OCR sentences, and object detection results inputs and uses the joint representation for prediction.

Each meme also contains textual information. We can extract the sentences with the help of the off-the-shelf OCR system. Formally, we can extract the $S^o = \{w_1^o, \ldots, w_{N_O}^o\}$ from the given meme image, where N_o is the number of words. We follow the same operations as image caption and calculate the feature for each token as:

$$\mathbf{w}_i^o = \text{LN}\big(f_{\text{WordEmb}}(w_i^o) + f_{\text{SegEmb}}(w_i^o) + f_{\text{PosEmb}}(i)\big) \quad (3)$$

where $\hat{\mathbf{w}}_i^o \in \mathbb{R}^{d_w}$ is the word-level feature for OCR token. Those three embedding functions are shared with Eq. 2.

We concatenate the image caption embeddings with OCR sentence embeddings as $\{\mathbf{w}^o_{1:N_O}, \mathbf{w}_{[\text{SEP}]}, \hat{\mathbf{w}}^c_{1:N_C}, \mathbf{w}_{[\text{SEP}]}\}$, where $\mathbf{w}_{[\text{SEP}]}$ is the word embedding for special token [SEP].

3.1.2. Image Embedding

Instead of getting the global representation for each image, we take the visual features of detected objects as the representation for the image. Specifically, we extract and keep a fixed number of semantic region proposals from the pre-trained Faster R-CNN[22][1]. Formally, an image I consists of N_v objects, where each object o_i is represented by its region-of-interest (RoI) feature $\mathbf{v}_i \in \mathbb{R}^{d_o}$, and its positional feature $\mathbf{p}_i^o \in \mathbb{R}^4$ (normalized top-left and bottom-right coordinates). Each region embedding is calculated as follows:

$$\mathbf{v}_i^o = \text{LN}\left(f_{\text{VisualEmb}}(\mathbf{v}_i) + f_{\text{VisualPos}}(\mathbf{p}_i^o)\right) \quad (4)$$

where $\mathbf{v}_i^o \in \mathbb{R}^{d_v}$ is the position-aware feature for each proposal, $f_{\text{VisualEmb}}(\cdot)$ and $f_{\text{VisualPos}}(\cdot)$ are two embedding layers.

3.2. Triplet-Relation Network

The target of triplet-relation network is to model the cross-modality relationships between image features ($\mathbf{v}^o_{1:N_v}$) and two textual features ($\hat{\mathbf{w}}^c_{1:N_c}$ and $\mathbf{w}^o_{1:N_o}$). Motivated by the success of the self-attention mechanism [8], we adopt the transformer network as the core module for our TRN.

Each transformer block consists of three main components: Query ($\boldsymbol{Q}$), Keys ($\boldsymbol{K}$), and Values ($\boldsymbol{V}$). Specifically, let $\boldsymbol{H}^l = \{h_1, \ldots, h_N\}$ be the encoded features at l-th layer. $\boldsymbol{H}^l$ is first linearly transformed into $\boldsymbol{Q}^l$, $\boldsymbol{K}^l$, and $\boldsymbol{V}^l$ with learnable parameters. The output $\boldsymbol{H}^{l+1}$ is calculated with a softmax function to generate the weighted-average score over its input values. For each transformer layer, we calculate the outputs for each head as follows:

$$\boldsymbol{H}^{l+1}_{\text{Self-Att}} = \text{Softmax}(\boldsymbol{Q}^l(\boldsymbol{K}^l)^T/\sqrt{d_k}) \cdot \boldsymbol{V}^l \quad (5)$$

where d_k is the dimension of the Keys and $\boldsymbol{H}^{l+1}_{\text{Self-Att}}$ is the output representation for each head.

Note that $\boldsymbol{H}^0$ is the combination of the two textual features and visual features. In this paper, we explore two variants of TRN: one-stream [20] and two-stream [9]. One-stream denotes that we model the visual and textual features together in a single stream. Two-stream means we use two separate streams for vision and language processing that interact through co-attentional transformer layers. For each variant, we stack L_{TRN} these attention layers which serve the function of discovering relationships from one modality to another. For meme detection, we take the final representation $h_{[\text{CLS}]}$ for the [CLS] token as the joint representation.

[1] https://github.com/airsplay/py-bottom-up-attention

3.3. Learning

We first train the image encoder and sentence decoder for image captioner training by minimizing the cross-entropy (CE) loss. After training with CE loss, we further apply a reinforcement learning loss that takes the CIDEr [23] score as a reward and optimize the image captioner with the SCST in [24]. For meme detection training, we feed the joint representation $h_{[\text{CLS}]}$ of language and visual content to a fully-connected (FC) layer, followed by a softmax layer, to get the prediction probability: $\hat{y} = \text{softmax}(f_{\text{FC}}(h_{[\text{CLS}]}))$. A binary cross-entropy (BCE) loss function is used as the final loss function for meme detection:

$$\mathcal{L}_{\text{BCE}}(\theta) = -\mathbb{E}_{I \sim \mathcal{D}}[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \tag{6}$$

where N is the number of training samples, I is sampled from the training set $\mathcal{D}$, y and $\hat{y}$ represent the ground-truth label and detected result for the meme, respectively.

4. EXPERIMENTS

4.1. Dataset and Implementation Details

In our experiments, we use two datasets: MSCOCO [25] and Hateful Memes Detection Challenge dataset are provided by Facebook [17]. We describe the detail of each dataset below.

MSCOCO. MSCOCO is an image captioning dataset which has been widely used in image captioning task. It contains 123,000 images, where each image has five reference captions. During training, we follow the setting of [26]. The best image captioner is selected base on the highest CIDEr score. Note that we only use MSCOCO during preprocessing.

The Hateful Memes Challenge Dataset. This dataset is collected by Facebook AI as the challenge set. The dataset includes 10,000 memes, where each sample contains an image and OCR sentence in the image. We use the official OCR results provided in the dataset. For the purpose of this challenge, the labels of memes have two types, non-hateful and hateful. The dev and test set consist of 5% and 10% of the data, respectively, and are fully balanced. The rest of the data is used as train set, which contains 64% non-hateful memes and 36% hateful memes.

Data Augmentation. We augment the sentence in the hateful memes dataset with the back-translation strategy. Specifically, we enrich the OCR sentences through two pretrained back-translator[2]: English-German-English and English-Russian-English. We also apply different beam sizes (2, 5, and 10) during the sentence decoding to get the diverse sentences.

Implementation Details. We present the hyperparameters related to our baselines and discuss those related to model training. For visual feature preprocessing, we extract the RoI features using the pretrained Faster R-CNN object detector [22]. We keep 36 region proposals for each image and get the corresponding RoI feature, bounding box, and predicted labels. During image captioning training, we use a mini-batch size of 100 and an initial learning rate of 1e-4. We use Adam [27] as the optimizer. During the training of memes detection, we set the dimension of the hidden state of the transformer to 768, and initialize the transformer in our model with the BERT models pertained in MMF[3]. The number of tokens N is set to 100 in our experiments. We use Adam [27] optimizer with an initial learning rate of 5e-5 and train for 10,000 steps. We report our models' performance with the AUROC. It measures how well the memes predictor discriminates between the classes as its decision threshold is varied. During online submission, we submit the predicted probabilities for the test samples. The online (Phase1 and Phase2) rankings are decided based on the best submissions by AUROC.

Table 1. Experimental results on Hateful Memes Detection dev split. 'Obj.+OCR' means the object RoI features and OCR text. 'Back-Trans.' denotes back-translation.

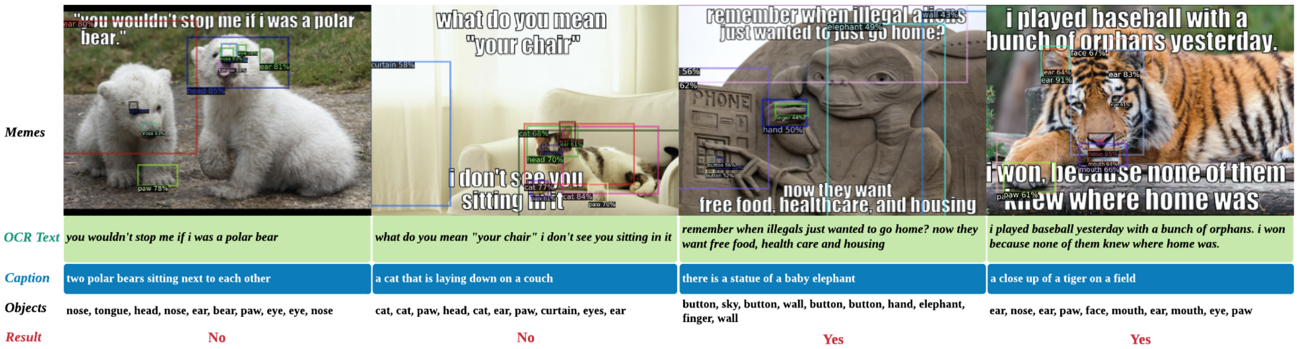
Model	Basic Inputs	Additional Inputs			AUROC
	Obj.+OCR	Obj. Labels	Back-Trans.	Caption	
V+L	✓	✗	✗	✗	70.47
	✓	✗	✗	✓	72.97
	✓	✗	✓	✗	72.43
	✓	✗	✓	✓	**73.93**
	✓	✓	✗	✗	70.96
	✓	✓	✗	✓	72.66
	✓	✓	✓	✗	71.57
	✓	✓	✓	✓	72.15
V&L	✓	✗	✗	✗	66.94
	✓	✗	✗	✓	**71.11**
	✓	✗	✓	✗	63.47
	✓	✗	✓	✓	67.94
	✓	✓	✗	✗	70.22
	✓	✓	✗	✓	70.46
	✓	✓	✓	✗	66.68
	✓	✓	✓	✓	69.85

Table 2. Performance comparison on online test server.

Inputs	Model	AUROC
	Human [17]	82.65
Image	Image-Region [17]	55.92
Text	Text BERT [17]	65.08
Image + Text	ViLBERT [17]	70.45
	Visual BERT [17]	71.33
	ViLBERT CC [17]	70.03
	Visual BERT COCO [17]	71.41
Image + Text + Caption	Ours (V+L)	**73.30**
	Ours (V&L)	**71.88**
	Ours (V+L and V&L)	**78.86**

[2] https://github.com/pytorch/fairseq

[3] Https://github.com/facebookresearch/mmf



Memes				
OCR Text	*you wouldn't stop me if i was a polar bear*	*what do you mean "your chair" i don't see you sitting in it*	*remember when illegals just wanted to go home? now they want free food, health care and housing*	*i played baseball yesterday with a bunch of orphans. i won because none of them knew where home was.*
Caption	two polar bears sitting next to each other	a cat that is laying down on a couch	there is a statue of a baby elephant	a close up of a tiger on a field
Objects	nose, tongue, head, nose, ear, bear, paw, eye, eye, nose	cat, cat, paw, head, cat, ear, paw, curtain, eyes, ear	button, sky, button, wall, button, button, hand, elephant, finger, wall	ear, nose, ear, paw, face, mouth, ear, mouth, eye, paw
Result	No	No	Yes	Yes

Fig. 3. Qualitative examples of of hateful memes detection. Generated caption and object labels are shown for each sample.

4.2. Result and Discussion

We conduct the ablation study in Table 1. The baseline models can be divided into two categories: V+L and V&L. V+L represents the one-stream model. It takes the concatenated visual and textual features as input and produces contextually embedded features with a single BERT. The parameters are shared between visual and textual encoding. We initialize the V+L models with pretrained Visual BERT [20]. V&L represents the two-stream model. It first adopts the two-stream for each modality and then models the cross-relationship with a cross-attention based transformer. The V&L models with pretrained ViLBERT [9]. The parameters for all models are finetuned on the meme detection task.

Effectiveness of Image Captioning. In Table 1, we can see that V&L models with image caption outperform other V&L models by a large margin on the dev set. These results support our motivation that image captioning helps hateful memes detection. The generated image descriptions provide more meaningful clues to detect the 'hateful' memes since the captions can describe the image's content and provide semantic information for cross-modality relationship reasoning. The performance boost brought by image captioning further indicates that, due to the rich and societal content in memes, only considering object detection and OCR recognition is insufficient. A practical solution should also explore some additional information related to the meme.

Effectiveness of Language Augmentation. We also verify the effectiveness of data augmentation in Table 1. We can see that back-translation can bring some improvement to V&L models but not for V+L models. We think the reason for the ineffectiveness of back-translation on V+L models is that the one-stream models handle the multimodal embeddings with the shared BERT model. OCR sentences and image content are not semantically aligned in hateful memes detection. Thus the effectiveness of the sentence augmentation is weakened. For V&L, back-translation can improve the intra-modality modeling for language, as it contains independent branches for visual and textual modeling separately.

Effectiveness of Visual Labels. We consider combining the predicted object labels as additional input features and concatenate the object labels with OCR text and image caption. We can see that the object labels can improve the V+L and V&L models. This is reasonable since object labels can be treated as the "*anchor*" between RoI features and textual features (OCR text and caption) [28].

Comparisons with the Existing Methods. Table 2 shows the comparisons of our method on Hateful memes challenge with existing methods. Those comparison results are directly copied from [17]. As we can see, our method achieves better performance, demonstrating the advantage of our triplet-relation network. Our best model ensembled with 12 models (V+L and V&L), named as "*naoki*", achieves the 6th position among 276 teams in the Phase 2 competition[4].

Visualization Results. Fig. 3 shows some generated captions and predicted results. With the help of image caption generation, our model can understand the implicit meaning between the image and sentences. For example, although there is no explicit relationship between the image and OCR text in the last image, our method can still predict the correct result by connecting different modalities with the image caption.

5. CONCLUSION

In this paper, we propose a novel multimodal learning method for hateful memes detection. Our proposed model exploits the combination of image captions and memes to enhance cross-modality relationship modeling for hateful memes detection. It achieves competitive results in the hateful memes detection challenge. We envision such a triplet-relation network extended to other frameworks that require additional features from multimodal signals.

[4] https://www.drivendata.org/competitions/70/hateful-memes-phase-2/leaderboard/

6. REFERENCES

[1] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, 2018.

[2] Paula Fortuna and Sérgio Nunes, "A survey on automatic detection of hate speech in text," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 85:1–85:30, July 2018.

[3] Jiuxiang Gu, Jason Kuen, Shafiq Joty, Jianfei Cai, Vlad Morariu, Handong Zhao, and Tong Sun, "Self-supervised relationship probing," *NeurIPS*, 2020.

[4] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering," in *CVPR*, 2017.

[5] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh, "Vqa: Visual question answering," in *ICCV*, 2015.

[6] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018.

[7] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li, "Dynamic fusion with intra-and inter-modality attention flow for visual question answering," in *CVPR*, 2019.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[9] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee, "VilBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *NeurIPS*, 2019.

[10] Hedi Ben-Younes, Remi Cadene, Nicolas Thome, and Matthieu Cord, "Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection," in *AAAI*, 2019.

[11] Anna Schmidt and Michael Wiegand, "A survey on hate speech detection using natural language processing," in *SocialNLP*, 2017.

[12] Shervin Malmasi and Marcos Zampieri, "Detecting hate speech in social media," *CoRR*, vol. abs/1712.06427, 2017.

[13] Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis, "Large scale crowdsourcing and characterization of twitter abusive behavior," in *ICWSM*, 2018.

[14] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber, "Racial bias in hate speech and abusive language detection datasets," in *ACL*, 2019.

[15] Z. Waseem, T. Davidson, D. Warmsley, and I. Weber, "Understanding abuse: A typology of abusive language detection subtasks," in *ALW*, 2017.

[16] Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas, "Exploring hate speech detection in multimodal publications," in *WACV*, 2020.

[17] 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," *arXiv preprint arXiv:2005.04790*, 2020.

[18] Huijuan Xu and Kate Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *ECCV*, 2016.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

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

[21] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

[23] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh, "Cider: Consensus-based image description evaluation," in *CVPR*, 2015.

[24] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel, "Self-critical sequence training for image captioning," in *CVPR*, 2017.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[26] Andrej Karpathy and Li Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015.

[27] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[28] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al., "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *ECCV*, 2020.