

Multimodal Sarcasm Target Identification in Tweets

Jiquan Wang^{1,2*}, Lin Sun^{1*†}, Yi Liu^{1,2}, Meizhi Shao¹, Zengwei Zheng¹

¹Department of Computer Science, Zhejiang University City College

²College of Computer Science and Technology, Zhejiang University

{wangjiquan,liuyi79}@zju.edu.cn, {sunl,zhengzw}@zucc.edu.cn

Abstract

Sarcasm is important to sentiment analysis on social media. Sarcasm Target Identification (STI) deserves further study to understand sarcasm in depth. However, text lacking context or missing sarcasm target makes target identification very difficult. In this paper, we introduce multimodality to STI and present Multimodal Sarcasm Target Identification (MSTI) task. We propose a novel multi-scale cross-modality model that can simultaneously perform textual target labeling and visual target detection. In the model, we extract multi-scale visual features to enrich spatial information for different sized visual sarcasm targets. We design a set of convolution networks to unify multi-scale visual features with textual features for cross-modal attention learning, and correspondingly a set of transposed convolution networks to restore multi-scale visual information. The results show that visual clues can improve the performance of TSTI by a large margin, and VSTI achieves good accuracy.

1 Introduction

Sarcasm is a type of sentiment in which people express their negative feelings using positive or intensified positive words. It has the power to disguise the hostility of the speaker (Dews and Winner, 1995), thereby enhancing the effect of mockery or humor on the listener. Sarcasm is prevalent on today’s social media platforms such as Twitter, and automatic Sarcasm Target Identification (STI) bears great significance in customer service, opinion mining, and online harassment detection. Previous works about STI have focused on text modality and proposed some methods such as rule-based, statistical classifier-based (Joshi et al., 2018), and deep learning models with socio-linguistic features (Patro et al., 2019).

*Equal contribution

†Lin Sun is the corresponding author.



Figure 1: Two examples of MSTI. (a) “This guy” and the blue bounding box denote textual and visual STs, respectively; (b) No textual ST and the blue bounding box denotes visual ST.

However, detecting a sarcasm target with only text modality is not sufficient and complete. For example, in Figure 1(a), we are not sure whether the context conveys positive or negative emotions if we only see the text “This guy definitely deserves \$15 an hour!”. However, the negative information comes from the image. When observing a lazy guy in the picture lying on the chair, we can easily determine that the lazy guy is a sarcasm target and label the text “This guy” as a sarcasm target (ST). Moreover, the sarcasm target sometimes does not appear explicitly in the text, which is marked as ‘OUTSIDE’. In ALTA Shared Task (Molla and Joshi, 2019), ‘OUTSIDE’ cases account for over 30% of the data. For example, in the tweet of Figure 1(b), the author teased that the skirt was too long, similar to a bed sheet; therefore, the sarcasm target should be the long skirt. However, no sarcasm target appears in the text but we can label the long skirt as an ST with a blue bounding box in the picture. The above examples illustrate the necessity of combining images for STI.

In this paper, we introduce a novel task called Multimodal Sarcasm Target Identification (MSTI) on social media data. The MSTI task is to extract sarcasm targets (STs) from both texts and images

in tweets. The textual ST is a word or a phrase, and the visual ST is an object labeled by a bounding box, as shown in Figure 1. The challenge of the MSTI task is not only to extract both textual and visual features but also to leverage cross-modality interaction and semantic learning to improve the performance of STI. The contributions of this paper can be summarized as follows:

- To the best of our knowledge, this is the first attempt to perform the MSTI task. We build an MSTI dataset and propose a novel cross-modality MSTI framework. Textual and multi-scale visual features are fused in a cross-modality encoder via convolution and transposed convolution. Our model performs textual and visual tasks simultaneously in an end-to-end manner, and it can leverage textual and visual contexts to enhance textual and visual representations for MSTI.
- We design a cross-modality attention visualization method in terms of text-to-image and image-to-text to illustrate the mutual effects between textual and visual modalities. These results show the image regions and words extracted through cross-modality attention are the keys to sarcasm and explain the improved performance of TSTI and VSTI by cross-modality learning.
- The comprehensive experimental results are presented. The results indicate that the images in tweets improve the performance of TSTI by a large margin. Comparisons with textual, object detection, and pretrained multimodal baselines show the advanced performance of our model.

2 Related Work

The existing research on sarcasm analysis mainly focuses on sarcasm detection (SD) and sarcasm target identification (STI). We begin the literature review with textual sarcasm.

Textual Sarcasm Detection. Traditional sarcasm detection is defined as a binary classification of sarcastic or non-sarcastic sentiments in text (Guo et al., 2021). Earlier approaches (Joshi et al., 2017) were based on sarcastic pattern rules (Riloff et al., 2013) or statistical models such as SVM (Joshi et al., 2015) or logistic regression (Bamman and Smith, 2015). Recently, deep learning techniques

have gained popularity. Word embeddings and LSTM/CNN model were employed in Joshi et al. (2016); Zhang et al. (2016). Furthermore, Peled and Reichart (2017) presented a neural machine translation framework and Tay et al. (2018) proposed an attention-based neural model to interpret and reason with sarcasm. Xiong et al. (2019) proposed a self-matching network to capture incongruity information by exploring word-to-word interactions. Agrawal et al. (2020) formulated sarcasm detection as a sequence classification problem by leveraging the natural shifts in various emotions over the course of a piece of text. Babanejad et al. (2020) extended the architecture of BERT by incorporating both affective and contextual feature embeddings. Guo et al. (2021) proposed a latent-optimized adversarial neural transfer model for cross-domain sarcasm detection.

Textual Sarcasm Target Identification. To deepen the field of sarcasm analysis, STI has been well studied recently (Patro et al., 2019; Parameswaran et al., 2021). The goal of STI is to label the subject of mockery or ridicule within sarcastic texts. Patro et al. (2019) showed that the Exact Match (EM) accuracy on tweets is approximately 30%.

Joshi et al. (2018) introduced the STI problem and summarized the 2019 ALTA shared task regarding STI (Molla and Joshi, 2019). The evaluation metrics such as EM accuracy and F1 score were presented. Patro et al. (2019) presented a deep learning framework augmented with socio-linguistic features to detect sarcasm targets. Parameswaran et al. (2019) employed an ensemble of classifiers such as SVM, logistic and linear classifiers to classify ‘OUTSIDE’ and ‘NOT OUTSIDE’, then used a rule-based approach to extract the target sarcasm words from the ‘NOT OUTSIDE’ samples.

Multimodal Sarcasm Detection. Benefiting from images, multimodal sarcasm detection (MSD) has gained increasing research attention. Schifanella et al. (2016) first tackled this task as a multimodal classification problem. They concatenated the visual and textual features and employed SVM or a neural network consisting of fully connected and softmax layers, to detect sarcasm. Cai et al. (2019) extended the input modalities to a triplet of text, image, and image attributes, and they proposed a hierarchical fusion model for sarcasm detection. Castro et al. (2019) proposed a video-level multimodal sarcasm detection task. Features were ob-

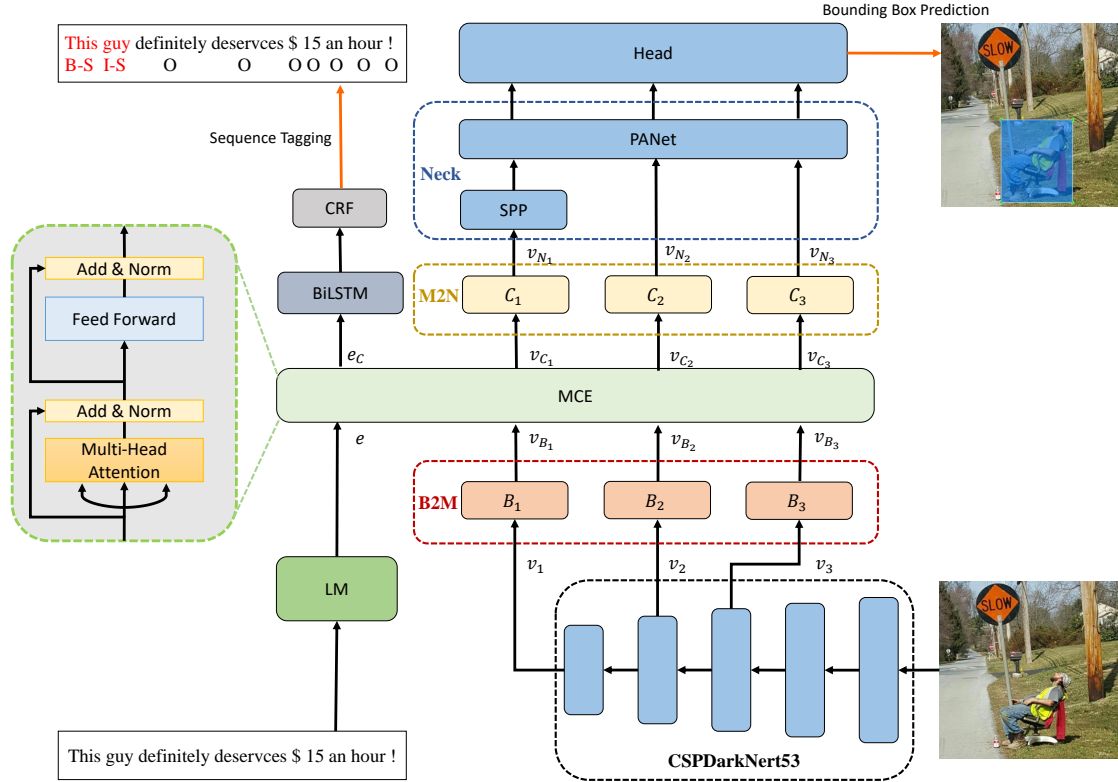


Figure 2: Our MSTI architecture overview.

tained from three modalities, i.e., text, speech, and video, and an SVM classifier with RBF kernel was employed. Sangwan et al. (2020) presented an RNN-based model and gating mechanism, which attempted to decide the weight of image modality regarding textual modality. Pan et al. (2020) proposed a BERT-based model that concentrated on both intra- and inter-modality incongruity for multimodal sarcasm detection. Xu et al. (2020) constructed the decomposition network to model the discrepancy between image and text and the relation network to model the semantic association in cross-modality context.

3 The Proposed Approach

In this section, we introduce a novel neural architecture for MSTI, as shown in Figure 2.

3.1 Neural Architecture

Figure 2 illustrates the overall architecture of our MSTI model. The model mainly consists of five components: (1) Backbone to MCE converter (B2M), (2) Multi-scale Cross-modality Encoder (MCE), (3) MCE to Neck network converter (M2N), (4) Textual Sarcasm Target Identification (TSTI), and (5) Visual Sarcasm Target Identifica-

tion (VSTI). We first extract textual and visual features separately. The multi-scale visual features of the last three blocks of the pretrained backbone network are unified to the same dimension by B2M and input to the MCE together with textual features. The MCE outputs cross-modality representations, where the parts corresponding to textual features are fed into BiLSTM-CRF to label the sequence for TSTI and the parts corresponding to multi-scale visual features restored by M2N are connected to the neck and head networks to predict bounding boxes for VSTI.

3.2 Textual and Visual Representations

Textual Representation: We obtain contextual word embeddings from pretrained language models (LM) such as BERT (Devlin et al., 2019) to extract linguistic features. Let $S = ([CLS], t_1, \dots, t_n, [SEP])$ be the token sequence and $e = (e_1, \dots, e_n)$ be the contextual word embeddings generated by a pretrained LM, where $e_i \in \mathbb{R}^d$. As shown in Figure 2, the contextual word embeddings e represent the textual input of the next module MCE.

Visual Representation: We extract visual features from an image with pretrained backbone networks

such as ResNet (He et al., 2016), VGG (Simonyan and Zisserman, 2014), and CSPDarkNet (Wang et al., 2020). To improve the detection performance of sarcasm targets with various sizes, our model performs VSTI using multi-scale visual features. The multi-scale outputs at the last three blocks of the backbone are denoted as v_1 , v_2 , and v_3 , shown in Figure 2, for the later use of the Neck network. The dimensions of v_1 , v_2 , and v_3 are $d_{s_1} \times d_{s_1} \times d_1$, $d_{s_2} \times d_{s_2} \times d_2$, and $d_{s_3} \times d_{s_3} \times d_3$, respectively, where $d_{s_i} \times d_{s_i}$ represents image scale and d_i represents feature map, $i = \{1, 2, 3\}$.

3.3 B2M Converter

The B2M converter aims to unify the dimensions of three visual features with the dimension d of textual feature and lower the scales of three visual features to reduce the computation of the MCE. The B2M has three parts B_1 , B_2 , and B_3 corresponding to the visual features v_1 , v_2 , and v_3 . Each part consists of convolutional layers followed by Rectified Linear Unit (ReLU) and max pooling layer.

Table 1 shows the architecture of B2M. The input dimensions of B_1 , B_2 , and B_3 are $19 \times 19 \times 1024$, $38 \times 38 \times 512$, and $76 \times 76 \times 256$, respectively, when the backbone is set to CSPDarkNet53 (Bochkovskiy et al., 2020). We denote the outputs of B_1 , B_2 , and B_3 as v_{B_1} , v_{B_2} , and v_{B_3} , respectively. According to the computation of the convolutional layer, the output scale is $\lfloor \frac{I+2P-K}{S} \rfloor + 1$, where I is the dimension of input scale, P is padding, S is stride, and K is kernel. The Conv generates feature maps of d , which is the same as the dimension of the word embeddings. Then, we can obtain all v_{B_1} , v_{B_2} , and v_{B_3} with size $5 \times 5 \times d$. Finally, we flatten the shape size 5×5 to 25 visual tokens $\{v_{B_i}^{p,q}\}$ ($1 \leq p, q \leq 5$) to generate the visual inputs of the MCE.

3.4 Multi-scale Cross-modality Encoder

The MCE is based on the Transformer encoder architecture presented in Vaswani et al. (2017) and shown in the left of Figure 2. The Transformer encoder has a multi-head self-attention sub-layer and a fully connected feed-forward sub-layer. A residual connection and layer normalization are employed around two sub-layers. The Transformer encoder adopts scaled dot-product attention, which is defined as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \quad (1)$$

B2M	B_1	Conv MaxPooling	$[K = 3 \times 3, P = 1, S = 2]$ $[2 \times 2]$
	B_2	Conv MaxPooling	$\left[\begin{array}{l} K = 3 \times 3, P = 1, S = 2 \\ K = 3 \times 3, P = 1, S = 2 \end{array} \right]$ $[2 \times 2]$
	B_3	Conv MaxPooling	$\left[\begin{array}{l} K = 5 \times 5, P = 2, S = 4 \\ K = 3 \times 3, P = 1, S = 2 \end{array} \right]$ $[2 \times 2]$
M2N	C_1	UpSampling ConvT	$[2 \times 2]$ $[K = 3 \times 3, P = 1, S = 2]$
	C_2	UpSampling ConvT	$[2 \times 2]$ $\left[\begin{array}{l} K = 3 \times 3, P = 1, S = 2 \\ K = 3 \times 3, P = 1, S = 2 \end{array} \right]$
	C_3	UpSampling ConvT	$[2 \times 2]$ $\left[\begin{array}{l} K = 3 \times 3, P = 1, S = 2 \\ K = 5 \times 5, P = 2, S = 4 \end{array} \right]$

Table 1: Architecture of B2M and M2N converters.

where matrices Q , K , and V consist of queries, keys, and values, respectively, and d_k is the dimension of keys. In our model, we concatenate the textual and visual features into a sequence G ,

$$G = (\underbrace{e_1, \dots, e_n}_n, \underbrace{v_{B_1}^{1,1}, \dots, v_{B_1}^{5,5}}_{25=5 \times 5}, \underbrace{v_{B_2}^{1,1}, \dots, v_{B_2}^{5,5}}_{25=5 \times 5}, \underbrace{v_{B_3}^{1,1}, \dots, v_{B_3}^{5,5}}_{25=5 \times 5}). \quad (2)$$

We feed G into the MCE, and therefore $Q = K = V = G^T$.

The outputs of the MCE are divided into two parts: the corresponding textual part e_C is used for TSTI, and the corresponding multi-scale visual parts v_{C_1} , v_{C_2} , and v_{C_3} are used for VSTI.

3.5 M2N Converter

The M2N converter is an inverse procedure of the B2M converter. The dimensions of the output v_{N_i} of the M2N converter are the same as those of the input v_i of the B2M converter, where $i = \{1, 2, 3\}$. The M2N converter has three parts C_1 , C_2 , and C_3 , corresponding to B_1 , B_2 , and B_3 of the B2M converter, respectively. The architecture of M2N is shown in Table 1. Each part consists of transposed convolution (ConvT) (Dumoulin and Visin, 2016) followed by ReLU and upsampling layer. The ConvT is considered as the reverse operation of convolution. If the ConvT’s kernel size, padding size, and stride are the same as those carried out on the Conv layer, then the ConvT generates the same spatial dimension as that of the Conv’s input. Upsampling reverses the pooling operation by the nearest-neighbor interpolation algorithm.

3.6 Textual Sarcasm Target Identification

We use the BIO (short for Beginning, Inside, and Outside) schema (Ramshaw and Marcus, 1995)

to label textual sarcasm targets. The ‘B-ST’ tag indicates the beginning of an ST and the ‘I-ST’ tag indicates the inside of an ST. The ‘O’ tag indicates that a token does not belong to any ST.

We employ a classical sequence tagging model, i.e., BiLSTM-CRF (Huang et al., 2015), to label the textual STs. The bidirectional LSTM (BiLSTM) first processes each sentence token-by-token and produces forward and backward hidden vectors for each token. Then the concatenation of the two hidden vectors is input to a Conditional Random Fields (CRFs) layer (Lafferty et al., 2001). For a sequence of tags $y = \{y_1, \dots, y_n\}$, the probability of the label sequence y is defined as follows:

$$p(y|x) = \frac{e^{s(x,y)}}{\sum_{y' \in Y} e^{s(x,y')}} \quad (3)$$

where Y is all possible tag sequences for the sentence x and $s(x, y)$ are feature functions modeling transitions and emissions. The computation details can be found in Lample et al. (2016). The objective of labeling ST is to minimize the negative log-likelihood over the training data $\mathcal{D}_t = \{(x^{(i)}, y^{(i)})\}_{i=1}^M$:

$$\mathcal{L}_{TSTI} = - \sum_{i=1}^M \log(p(y^{(i)}|x^{(i)})). \quad (4)$$

3.7 Visual Sarcasm Target Identification

There are two kinds of object detectors, one-stage and two-stage. One-stage object detector such as YOLO (Redmon et al., 2016) is faster and simpler. In this paper, we adopt YOLOv4’s Neck and Head networks (Bochkovskiy et al., 2020) to perform VSTI. The multi-scale cross-modality features v_{N_1} , v_{N_2} , and v_{N_3} are connected to the Neck network, which consists of Spatial Pyramid Pooling (SPP) (He et al., 2015) and Path Aggregation Network (PANet) (Liu et al., 2018). The Neck network is to increase the receptive field and preserve spatial information. The Head network is used for predicting bounding boxes at 3 different scales. The output tensor of the Head network of YOLOv4 is $d_{s_i} \times d_{s_i} \times [3 * (4 + 1 + C)]$ at each scale, predicting 3 boxes per grid cell where each box has 4 bounding offsets (t_x, t_y, t_w, t_h) , 1 objectness score, and C class scores. The detailed computation of bounding offsets can be found in Redmon and Farhadi (2018). Each grid cell predicts the object probability and C class probabilities. In our model, since there is 1 sarcasm object class, we ablate C class scores.



Figure 3: Examples of the MSTI dataset. (a) STs are both in text and image, (b) ST is only in text, and (c) ST is only in image. The green bounding box indicates a visual ST.

The objective function of VSTI consists of bounding box regression loss \mathcal{L}_b and objectness score loss \mathcal{L}_o :

$$\mathcal{L}_{VSTI} = \mathcal{L}_b + \mathcal{L}_o. \quad (5)$$

As in YOLO, \mathcal{L}_b is based on the bounding box priors that are assigned to ground truth objects and computed by mean squared error (MSE). \mathcal{L}_o is computed by binary cross-entropy (BCE) for classifying the bounding box priors as object or non-object.

Finally, combining the TSTI and VSTI tasks, the objective function for MSTI is as follows:

$$\mathcal{L}_{MSTI} = \mathcal{L}_{TSTI} + \mathcal{L}_{VSTI}. \quad (6)$$

4 Experiments

4.1 Dataset

In this paper, we build an MSTI dataset for public research¹. We label textual and visual STs on the dataset collected by Cai et al. (2019) for multimodal sarcasm detection. Each sample is manually annotated by three persons based on their common sense. The agreement between the annotators is measured using a percentage of overlapping choices between the annotators, i.e., above one word overlapping for text phrases and above 50% intersection-over-union (IoU) overlapping for image regions. We ensure the quality of ground truth by keeping the consistency of all annotator’s

¹<https://github.com/wjq-learning/MSTI>

	#Tweet	#Textual ST	#Visual ST
Train	3,546	2,501	2,172
Dev	727	542	543
Test	742	573	524

Table 2: Statistics of the MSTI dataset.

Both in text and image	Only in text	Only in image
26.4%	41.9%	31.7%

Table 3: Proportion of multimodal sarcasm target types.

Small	Medium	Large
939 (29.0%)	1,024 (31.6%)	1,276 (39.4%)

Table 4: Number (percentage) of different sized visual STs.

opinions. The samples with annotations that the three annotators agree on are put into the dataset otherwise removed, making the annotations valid. Figure 3 shows three examples.

The statistics of the MSTI dataset are shown in Table 2. The MSTI dataset is split into 3,546/727/742 as Train/Dev/Test in experiments. The number of textual and visual sarcasm targets are also listed. Table 3 shows the proportion of multimodal sarcasm target types, i.e., STs appear both in the text and image, ST only appears in the text, and ST only appears in the image. Table 4 shows the number and percentage of different sized visual STs. We categorize the size of ST as small (area occupation<1.5%), medium (1.5%<area occupation<10%), and large (area occupation>10%).

4.2 Settings

We use CSPDarkNet53 as backbone and BERT-Base ($d=768$) or BERT-Large ($d=1024$) as LM. All images are shaped to a size of 608×608 . In the VSTI, we use the default settings of YOLOv4, e.g., IoU threshold and object confidence threshold. The weights of neural network are randomly initialized except the pretrained BERT, backbone and Neck networks. We train the model using the Adam (Kingma and Ba, 2014) optimizer with default settings and the learning rate is set to $1e-4$. All pretrained models are finetuned with a learning rate of $1e-5$. The mini-batch size is set to 8 and dropout rate is 0.5. We use two-layer BiLSTM with 768 hidden states and Transformer encoder with 12 heads.

We use Exact Match (EM) accuracy (Joshi et al.,

2018) and F1 score (Molla and Joshi, 2019) as evaluation metrics for TSTI. The EM accuracy is computed as the number of samples that strictly match the boundaries of gold annotations divided by the total number of samples. The F1 score $= 2/(1/P+1/R)$ is calculated from precision $P = TP/(TP+FP)$ and recall $R = TP/(TP+FN)$, where TP is correctly predicted target word, FP is incorrectly predicted target word, and FN is target word but not predicted. Average Precision (AP) is widely used to evaluate object detection (Lin et al., 2014). The COCO-style AP, AP_{50} , and AP_{75} are evaluated for VSTI. The COCO-style AP averages AP at $IoU=[0.5:0.05:0.95]$. AP_{50} corresponds to AP at $IoU=0.5$ and AP_{75} corresponds to AP at $IoU=0.75$. IoU measures the amount of overlap between two bounding boxes and here is used as a criterion that determines if a prediction matches ground truth.

We train the model on one machine with an NVIDIA RTX 3090 (GPU) and Intel Core i9 10900K (CPU). Our model takes approximately 10 hours for 60 epochs in training.

4.3 Baselines

Our model is compared with text baselines such as rule-based & statistical extractors, socio-linguistic features, and BERT, object detection baselines such as Mask R-CNN (He et al., 2020) and YOLOv4, and pretrained multimodal baselines such as VL-BERT (Su et al., 2019) and Unicoder-VL (Li et al., 2020).

Rule-based & Statistical Extractors. Joshi et al. (2018) introduced STI and proposed a method based on rules and statistical classification extractors. The sarcasm target was determined based on the results of two extractors. The configuration of R2 and ‘Hybrid AND’ performs the best on the MSTI dataset. We test the source code² as a baseline of TSTI.

Socio-linguistic Features. Patro et al. (2019) presented a deep learning framework augmented with socio-linguistic features to detect textual ST. Socio-linguistic features include the distribution of location (LOC) and organization (ORG) named entities, the distribution of POS tags, and the distribution of LIWC and Empath (Fast et al., 2016) categories. We test the source code³ as a baseline of TSTI.

²https://github.com/Pranav-Goel/Sarcasm_Target_Identification

³https://github.com/Srijanb97/Sarcasm_Target_Detection-EMNLP-

	Dev					Test				
	EM	F1	AP	AP ₅₀	AP ₇₅	EM	F1	AP	AP ₅₀	AP ₇₅
Rule-based & statistical extractors	12.5	19.1	-	-	-	13.6	19.3	-	-	-
Socio-linguistic features	18.9	23.6	-	-	-	18.1	22.3	-	-	-
BERT-Base	29.0	40.2	-	-	-	28.6	39.4	-	-	-
BERT-Large	33.8	42.5	-	-	-	33.5	42.4	-	-	-
Mask R-CNN (backbone=ResNeXt101+FPN)	-	-	27.2	43.5	28.6	-	-	26.8	43.3	28.1
YOLOv4 (backbone=CSPDarkNet53)	-	-	26.9	42.8	27.6	-	-	27.1	43.9	28.0
VL-BERT	30.8	41.2	24.7	40.2	26.2	30.9	42.0	25.7	40.5	26.7
Unicoder-VL	30.5	41.1	25.0	41.0	26.4	30.5	41.7	25.5	40.8	26.9
UNITER	29.8	40.4	24.9	40.8	26.5	30.0	40.5	25.9	41.1	26.8
Our model										
- backbone=ResNet152, LM=BERT-Base	34.0	45.2	30.4	48.8	31.5	34.4	44.9	29.9	49.6	32.1
- backbone=VGG19, LM=BERT-Base	33.6	44.4	29.6	48.4	30.9	34.5	45.1	28.6	47.7	30.5
- backbone=CSPDarkNet53, LM=BERT-Base	34.2	44.9	32.1	52.3	34.2	35.0	45.8	32.3	51.8	34.0
- backbone=CSPDarkNet53, LM=BERT-Large	36.9	47.3	32.3	52.6	34.6	37.2	47.9	32.6	51.9	34.6

Table 5: Results of our model compared with text, object detection, and pretrained multimodal baselines.

BERT. We follow the sequence tagging task of Devlin et al. (2019) as a baseline to perform TSTI. The BERT-based model followed by linear and softmax layers is tested to tag textual ST.

Object Detection Models. We treat VSTI as a single object detection problem and test two state-of-the-art models, i.e., Mask R-CNN⁴ and YOLOv4⁵. We train the models on the MSTI dataset with the default values of parameters in the repository.

Pretrained Multimodal Models. Recently, pretrained multimodal models such as VL-BERT, Unicoder-VL, and UNITER (Chen et al., 2020), have been proposed. These models use regions-of-interest (RoIs) produced by object detectors such as Faster R-CNN (Ren et al., 2016) as visual tokens. The inputs of pretrained multimodal models are RoI features and token embeddings; In this paper, we design an MSTI baseline approach based on pretrained multimodal models as follows: Labeling of the textual STs is based on the outputs of token embeddings, the same as in our TSTI method; The VSTI is performed by a binary classification on the outputs of RoI features followed by linear+softmax layers, and it is trained by the RoIs, which are considered as visual STs when the IoU with gold ST is larger than an optimal value of 0.7, otherwise they are considered as non-STs. We finetune the IoU threshold for non-maximum suppression (NMS) to ignore overlapping RoIs and find that $\text{IoU}_{NMS} = 0.2$ is optimal.

⁴https://github.com/matterport/Mask_RCNN

⁵<https://github.com/AlexeyAB/darknet>

	Dev			Test		
	AP _S	AP _M	AP _L	AP _S	AP _M	AP _L
Our model	28.7	33.1	34.2	28.3	33.4	34.9

Table 6: Performance on the different sized visual STs.

4.4 Results

Table 5 shows the performance of our model compared with text, object detection, and pretrained multimodal baselines on the Dev and Test sets. The results show that the BERT-based sequence tagging models are better than the previous works of STI (Joshi et al., 2018; Patro et al., 2019). Fusing visual clues, our model outperforms BERT-based textual models on average 5.3% in F1 score and 4.4% in EM accuracy. The object detection baselines such as Mask R-CNN and YOLOv4 which are directly trained by sarcastic objects are better than the pretrained multimodal baselines with RoIs detected by a traditional object detector, obtaining an increase of approximately 2% in AP metrics.

We test state-of-the-art backbones such as ResNet151 and VGG19, in which scale dimensions of the last three blocks are 19×19 , 38×38 , and 76×76 , respectively, the same as in CSPDarkNet53. Therefore, the B2M in Table 1 can be directly used for ResNet151 and VGG19, and the M2N works if the dimensions of the output feature maps of $\{C_1, C_2, C_3\}$ are set to $\{2048, 1024, 512\}$ for ResNet151 and $\{512, 512, 256\}$ for VGG19, respectively. The results show that CSPDarkNet53 achieves the best performance. In addition, Table 6 reports APs (namely, AP_S, AP_M, and AP_L) by our best model based on small, medium, and large

	EM	F1	AP	AP ₅₀	AP ₇₅
Our model	37.2	47.9	32.6	51.9	34.6
- w/o text	-	-	27.4(-5.2)	44.6(-7.3)	28.1(-6.5)
- w/o image	33.1(-4.1)	42.8(-5.1)	-	-	-
- w/ text and w/o TSTI loss	-	-	29.7(-2.9)	49.1(-2.8)	31.6(-3.0)
- w/ image and w/o VSTI loss	34.6(-2.6)	43.9(-4.0)	-	-	-

Table 7: Ablation results of our model on the Test set.

sarcasm targets, respectively.

4.5 Ablation Study

We ablate text (w/o text) or image (w/o image) from our multimodal model. Table 7 shows the results of our model (backbone=CSPDarkNet53, LM=BERT-Large). The performance drops by 5.1% in F1 score and 4.1% in EM accuracy when ablating images, indicating that visual clues are very useful for STI.

In addition, we ablate TSTI training (w/ text and w/o TSTI loss) or VSTI training (w/ image and w/o VSTI loss). We observe that by only adding text but not training the textual task, our model can greatly improve the VSTI performance, i.e., from 44.6% to 49.1% in AP₅₀. However, by only adding image but not training the image task, our model obtains a small increase of 1.5% EM accuracy and 1.1% F1 score for TSTI. These results indicate that texts has more explicit sarcasm information than images and sarcastic message likely comes more from texts, which are consistent with common sense.

4.6 Cross-modality Attention Visualization

We visualize the attentions of the MCE in terms of image-to-text and text-to-image in order to illustrate the sarcasm information added from another modality. The input of the MCE is composed of textual and multi-scale visual embeddings. We abbreviate G , previously defined in Eq. (2), as $G = (e, v_B)$ where $e = (e_1, \dots, e_n)$ and $v_B = (v_{B_1}^{1,1}, \dots, v_{B_1}^{5,5}, v_{B_2}^{1,1}, \dots, v_{B_2}^{5,5}, v_{B_3}^{1,1}, \dots, v_{B_3}^{5,5})$. The attention weight matrix of the h -th head can be divided to four submatrics and denoted as follows:

$$A^h = \begin{pmatrix} A^h(e, e) & A^h(e, v_B) \\ A^h(v_B, e) & A^h(v_B, v_B) \end{pmatrix}. \quad (7)$$

Thus, the scaled dot-product attention of Eq. (1) also can be written as $A^h G^\top$. We define the computation of image-to-text and text-to-image attentions as follows:

Text-to-image Attentions. The goal of text-to-image attention is to quantify the effect of text on each image block. We compute the average

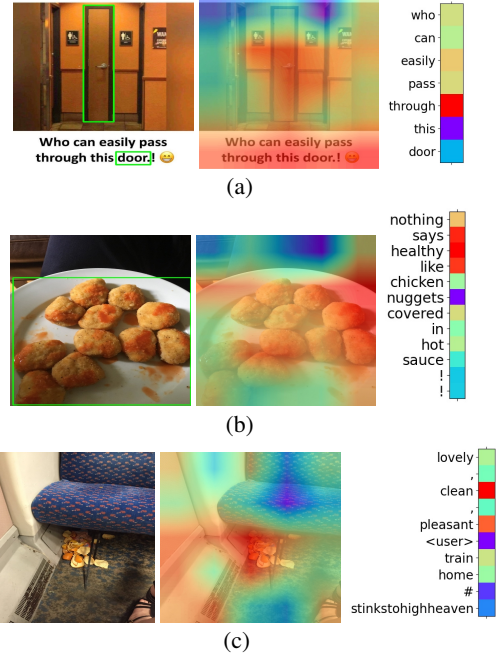


Figure 4: Examples of cross-modality attention visualization. The left, middle, and right columns are images, text-to-image attentions, and image-to-text attentions, respectively. The figures are best viewed in color.

sum of $A^h(v_B, e)$ across all words, H heads, and 3 scales, then obtain the text-to-image attention weights on the image block with the coordinate (p, q) as follows:

$$w_{v^p, q} = \frac{1}{3H} \sum_{i=1}^3 \sum_{h=1}^H \sum_{k=1}^n A^h(v_{B_i}^{p, q}, e_k). \quad (8)$$

Image-to-text Attentions. The text-to-image attention aims to quantify the effect of image on each word. We compute the average sum of $A^h(e, v_B)$ across 25 image blocks with 3 scales and H heads, then obtain the image-to-text attention weights as follows:

$$w_{e_k} = \frac{1}{3H} \sum_{i=1}^3 \sum_{h=1}^H \sum_{p=1}^5 \sum_{q=1}^5 A^h(e_k, v_{B_i}^{p, q}), \quad (9)$$

where $k = 1, \dots, n$.

Figure 4 shows the examples of text-to-image attentions in Eq. (8) and image-to-text attentions in

Eq. (9). The attention maps show some meaningful cues discovered by the MCE regarding sarcasm. The red color denotes the highest weights. We scale up the text-to-image attention map to image size using interpolation. As expected, the text-to-image attentions in the middle columns of Figure 4 focus on the regions that are highly relevant to sarcasm targets, such as door in (a), chicken nuggets in (b), and orange peel in (c). Surprisingly, the image-to-text attentions in the right columns of Figure 4 point out the key words to well understand sarcasm. Using these red colored words, the tweet authors express their opinions: (a) Can the door ‘through’? (b) Are the chicken nuggets ‘healthy’? (c) Is the train ‘clean’ or is train home ‘pleasant’?

5 Conclusions

In this paper, we introduce a new task for identifying both textual and visual sarcasm targets. This work provides a good attempt to detect sarcasm targets on images. Our model integrates multiple components such as sequence labeling, multi-scale cross-modality learning, and object detection. The experimental results not only illustrate that visual clues can improve the performance of TSTI by a large margin, approximately 5% in F1 score, but also prove that it is feasible to detect sarcasm targets in images, obtaining a good accuracy of 51.9% in AP₅₀.

6 Acknowledgements

This work was supported in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LGN22F020002, the National Natural Science Foundation of China (NSFC) under Grant No. 62072402, and Key Research and Development Program of Zhejiang Province under Grant No. 2022C03037.

References

Ameeta Agrawal, Aijun An, and Manos Papagelis. 2020. Leveraging transitions of emotions for sarcasm detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1505–1508.

Nastaran Babanejad, Heidar Davoudi, Aijun An, and Manos Papagelis. 2020. Affective and contextual embedding for sarcasm detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 225–243.

David Bamman and Noah Smith. 2015. Contextualized sarcasm detection on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal speed and accuracy of object detection.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multi-modal sarcasm detection in Twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an *Obviously* perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *ECCV*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shelly Dews and Ellen Winner. 1995. Muting the meaning a social function of irony. *Metaphor Arid Symbolic Activity*, 10(1):3–19.

Vincent Dumoulin and Francesco Visin. 2016. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Ethan Fast, Binbin Chen, and Michael S Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 4647–4657.

Xu Guo, Boyang Li, Han Yu, and Chunyan Miao. 2021. Latent-optimized adversarial neural transfer for sarcasm detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5394–5407.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2020. Mask r-cnn. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):386–397.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916.

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5).
- Aditya Joshi, Pranav Goel, Pushpak Bhattacharyya, and Mark Carman. 2018. Sarcasm Target Identification: Dataset and An Introductory Approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762.
- Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1006–1011.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of NAACL-HLT*, pages 260–270. Association for Computational Linguistics.
- Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. 2020. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, pages 11336–11344.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. 2018. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Diego Molla and Aditya Joshi. 2019. Overview of the 2019 ALTA shared task: Sarcasm target identification. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 192–196.
- Hongliang Pan, Zheng Lin, Peng Fu, Yatao Qi, and Weiping Wang. 2020. Modeling intra and inter-modality incongruity for multi-modal sarcasm detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1383–1392.
- Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, and David Eysers. 2019. Detecting target of sarcasm using ensemble methods. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 197–203.
- Pradeesh Parameswaran, Andrew Trotman, Veronica Liesaputra, and David Eysers. 2021. Detecting the target of sarcasm is hard: Really?? *Information Processing & Management*, 58(4):102599.
- Jasabanta Patro, Srijan Bansal, and Animesh Mukherjee. 2019. A deep-learning framework to detect sarcasm targets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6336–6342.
- Lotem Peled and Roi Reichart. 2017. Sarcasm SIGN: Interpreting sarcasm with sentiment based monolingual machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1690–1700.
- Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.
- Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767.
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149.
- Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 704–714.

- S. Sangwan, M. S. Akhtar, P. Behera, and A. Ekbal. 2020. I didn’t mean what i wrote! exploring multimodality for sarcasm detection. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Rossano Schifanella, Paloma de Juan, Joel Tetreault, and LiangLiang Cao. 2016. Detecting sarcasm in multimodal social platforms. In *Proceedings of the 24th ACM International Conference on Multimedia*, page 1136–1145.
- Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vi-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*.
- Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1010–1020.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- C. Wang, H. Mark Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh. 2020. Cspnet: A new backbone that can enhance learning capability of cnn. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1571–1580.
- Tao Xiong, Peiran Zhang, Hongbo Zhu, and Yihui Yang. 2019. Sarcasm detection with self-matching networks and low-rank bilinear pooling. In *The World Wide Web Conference*, page 2115–2124.
- Nan Xu, Zhixiong Zeng, and Wenji Mao. 2020. Reasoning with multimodal sarcastic tweets via modeling cross-modality contrast and semantic association. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3777–3786.
- Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460.

A Appendix

A.1 Case Study

Table 8 shows the examples of ground truth, BERT-large, VL-BERT, and our model in the MSTI dataset. The textual ST results are incorrect in all examples for the BERT baseline, but they are

correct for our model with visual clues. The detected visual STs are correct for our model; however, there are a few noisy RoIs, such as a child in Example 2, Theresa Mary and tie in Example 3, for the VL-BERT baseline. Examples 3 and 4 show differences between sarcasm targets and traditional object classes, such as strings “Donald Trump” and a scene of a person carrying bags. These cases also show that our model can detect visual STs with various sizes. The small size strings ‘door’ in Example 1 and “Donald Trump” in the middle of Example 3, the medium size string “Donald Trump” at the upper left corner and person “Donald Trump” in Example 3, and the large size objects in Examples 2 and 4, illustrate that the multi-scale features enrich spatial information for visual ST detection.

Table 9 shows the failure cases by our model. Our model fails in detecting the textual ST “chicken nuggets” in Example 1, probably because the linguistic representations do not perform well although the cross-modality attention contributes the sarcastic word ‘healthy’ shown in Figure 4(c). The visual STs such as the innovation object in Example 2 and string “SUNDAY FUNDAY” in Example 4 are not detected. In Examples 3 and 5, our model detects the highly related negative visual clues in images, i.e., crowded train and orange peel trash, although it fails in detecting textual STs.


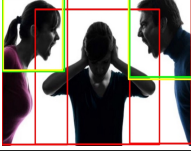



	1	2	3	4	5
Image	 Who can easily pass through this door! 😊	 [ST parents] ruin more young athletic careers than bad grades .	 this is the your <user> well done usa .	 “ me : i have no money . i ’m so poor right now . also me :”	 that ’s true for [ST bachelors] .
Textual ground truth	who can easily pass through this [ST door]	[ST parents] ruin more young athletic careers than bad grades .	this is the your <user> well done usa .	“ me : i have no money . i ’m so poor right now . also me :” that ’s true for [ST bachelors] .
BERT	[ST who] can easily pass through this door	parents ruin more young athletic careers than bad grades .	this is the your <user> well done [ST usa] .	“ me : i have no [ST money] . i ’m so poor right now . also me :” [ST that] ’s true for bachelors .
VL-BERT	who can easily pass through this door	[ST parents] ruin more young athletic careers than bad grades .	this is the your <user> well done usa .	“ me : i have no [ST money] . i ’m so poor right now . also me :” that ’s true for bachelors .
Our model	who can easily pass through this [ST door]	[ST parents] ruin more young athletic careers than bad grades .	this is the your <user> well done usa .	“ me : i have no money . i ’m so poor right now . also me :” that ’s true for [ST bachelors] .

Table 8: Examples of ground truth, BERT baseline, VL-BERT baseline, and our model. Green, red, and yellow rectangles indicate ground truth, VL-BERT baseline, and our model, respectively. It is better to enlarge images when observing visual ST.

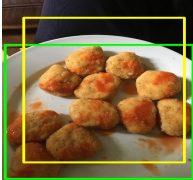




	1	2	3	4	5
Image					
Textual ground truth	nothing says healthy like [ST chicken nuggets] covered in hot sauce ! !	[ST innovation] for future	another wonderful <user> [ST commute home] on <user> here at the wall street station (2/3 station) ! thanks <user> <user> !	when it ’s # [ST bankholidayweek-end] & you have to work an extra shift .	lovely , clean , pleasant <user> [ST train home] # stinkstohigh-heaven
Our model	nothing says healthy like chicken nuggets covered in hot sauce ! !	[ST innovation] for future	another wonderful <user> commute home on <user> here at the wall street station (2/3 station) ! thanks <user> <user> !	when it ’s # [ST bankholidayweek-end] & you have to work an extra shift .	lovely , clean , pleasant [ST <user>] train home # stinkstohigh-heaven

Table 9: Failure cases by our model. Green and yellow rectangles indicate ground truth and our model, respectively.