Explaining (Sarcastic) Utterances to Enhance Affect Understanding in Multimodal Dialogues

Shivani Kumar[1], **Ishani Mondal**[2], **Md Shad Akhtar**[1], **Tanmoy Chakraborty**[3]
[1]Indraprastha Institute of Information Technology Delhi, India
[2]University of Maryland, College Park
[3]Indian Institute of Technology Delhi, India
shivaniku@iiitd.ac.in, ishani340@gmail.com, shad.akhtar@iiitd.ac.in, tanchak@iitd.ac.in

Abstract

Conversations emerge as the primary media for exchanging ideas. Identifying various affective qualities, such as sarcasm, humour, and emotions, is paramount for comprehending the true connotation of the emitted utterance. However, one of the major hurdles faced in learning these affect dimensions is the presence of figurative language *viz.* irony, metaphor, or sarcasm. We hypothesize that any detection system constituting the exhaustive and explicit presentation of the emitted utterance would improve the overall comprehension of the dialogue. To this end, we explore the task of Sarcasm Explanation in Dialogues that aims to unfold the hidden irony behind sarcastic utterances. We propose MOSES, a deep neural network, which takes a multimodal (sarcastic) dialogue instance as an input and generates a natural language sentence as its explanation. Subsequently, we leverage the generated explanation for various natural language understanding tasks in a conversational dialogue setup, such as *sarcasm detection*, *humour identification*, and *emotion recognition*. Our evaluation shows that MOSES outperforms the state-of-the-art system for SED by an average of $\sim 2\%$ on different evaluation metrics, such as ROUGE, BLEU, and METEOR. Further, we observe that leveraging the generated explanation advances three downstream tasks for affect classification – an average improvement of $\sim 14\%$ F1-score in the sarcasm detection task and $\sim 2\%$ in the humour identification and emotion recognition task. We also perform extensive analyses to assess the quality of the results.

Introduction

Expressing oneself eloquently to our conversation partner requires employing multiple affective components such as emotion, humour, and sarcasm. All such attributes interact with each another to present a concrete definition of an uttered statement (Roberts and Kreuz 1994). While affects such as emotion and humour deem easier to comprehend, sarcasm, on the other hand, is a challenging aspect to comprehend (Olkoniemi, Ranta, and Kaakinen 2016). Consequently, it becomes imperative for NLP systems to capture and understand sarcasm in its entirety. Sarcasm Explanation in Dialogues (SED) is a new task proposed recently in this direction (Kumar et al. 2022). In this work, we scour the *task of SED* and considerably improve the performance by proposing MOSES, a deep neural network which leverages the peculiarities of the benchmark dataset, WITS.

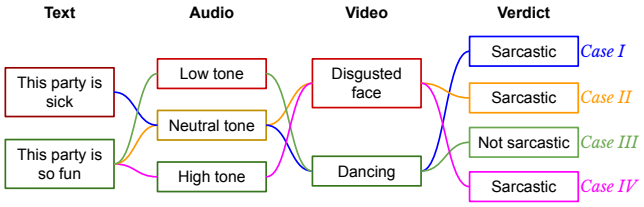

Figure 1: [Best viewed in color] Effect of multimodality on sarcasm.

Congruent to how humans make decisions after compiling data from all their available senses, (predominantly optical and auditory) multimodal analysis helps the machine mimic this behaviour. Particularly in the case of sarcasm interpretation, multimodal information provides us with essential indications to interpret irony. Figure 1 shows cases where multimodal knowledge such as audio and video can assist in comprehending sarcasm. The identical text, "This party is so fun", can result in diverse classes when integrated with different multimodal signals. For instance, when said with a *neutral tone* and a *disgusted face* (Case II), it results in a 'sarcastic' verdict. On the other hand, when the same utterance is expressed with a *low tone* and while the speaker is *dancing* (Case III), we can assume the speaker means what they say without any sarcasm. To capture multimodality efficaciously, it is vital to grant a prime prerogative to each input modality in order to capture its' peculiarities. To this end, we propose a *spotlight-aware fusion mechanism*, where the final multimodal amalgamated vector is tailored by paying special attention to individual modalities.

All affective components, such as sarcasm, humour, and emotion, work in tandem to convey a statement's intended meaning (Hasan et al. 2021; Chauhan et al. 2020). Accordingly, we hypothesize that understanding one of the affective markers, like sarcasm, in its entirety will influence comprehending others. Consequently, in this work, we deal with *leveraging sarcasm explanations* for three affect understanding tasks in dialogues, namely sarcasm detection, humour identification, and emotion recognition. The performance obtained from these tasks can be employed as a method to estimate the relevance of the SED task extrinsically.

We summarize our contributions below:

1. We explore the new task of SED and propose a novel model, MOSES, for it.
2. We compare MOSES with existing baselines and obtain state-of-the-art results for the SED task.
3. We show the application of the generated explanations in understanding different affective components – sarcasm, emotions and humour.
4. We show extensive quantitative and qualitative studies for all our experiments.

Reproducibility: The source code for MOSES and the execution instructions are present here: https://github.com/LCS2-IIITD/MOSES.git.

Related Work

Sarcasm. Figurative language such as sarcasm plays an integral role in resolving the veiled meaning of an uttered statement. Earlier studies dealt with sarcasm identification in standalone texts like tweets and reviews. (Kreuz and Caucci 2007; Tsur, Davidov, and Rappoport 2010; Joshi, Sharma, and Bhattacharyya 2015; Peled and Reichart 2017). A detailed summary of these studies can be found in the survey compiled by Joshi, Bhattacharyya, and Carman (2017). Several work explored sarcasm in other languages such as Hindi (Bharti, Sathya Babu, and Jena 2017), Arabic (Abu Farha and Magdy 2020), Spanish (Ortega-Bueno et al. 2019), Italian (Cignarella et al. 2018), or even code-mixed (Swami et al. 2018) languages.

Sarcasm and Dialogues. Linguistic and lexical traits were the primary sources of sarcasm markers in previous investigations (Kreuz and Caucci 2007; Tsur, Davidov, and Rappoport 2010). However, in more contemporary studies, attention-based approaches are used to capture the inter- and intra-sentence interactions in the text (Tay et al. 2018; Xiong et al. 2019; Srivastava et al. 2020). In terms of conversations, Ghosh, Richard Fabbri, and Muresan (2017) harnessed attention-based RNNs to capture context and determinate sarcasm.

Sarcasm and Multimodality. Castro et al. (2019) proposed a multimodal, multiparty, English dataset called MUStARD to benchmark the task of multimodal sarcasm identification in conversation. Subsequently, Chauhan et al. (2020) devised a multi-task framework by leveraging interdependency between emotions and sarcasm to solve the task of multimodal sarcasm detection. Another work (Hasan et al. 2021) established the interdependency of humour with sarcasm by suggesting a humour knowledge enriched Transformer model for sarcasm detection. In the code-mixed scenario, Bedi et al. (2021) proposed MASAC, a multimodal, multiparty, code-mixed dialogue dataset for humour and sarcasm detection. In the bimodal setting, sarcasm identification with tweets containing images has also been well explored (Cai, Cai, and Wan 2019; Xu, Zeng, and Mao 2020; Pan et al. 2020) .

Beyond Sarcasm Detection. Sarcasm generation is another direction that practitioners are inquisitive about due to its forthright benefit in enhancing chatbot engagement. Thereby, Mishra, Tater, and Sankaranarayanan (2019) induced sarcastic utterances by presenting context incongruity through fact removal and incongruous phrase insertion. A retrieve-and-edit-based unsupervised approach for generating sarcasm was proposed by Chakrabarty et al. (2020) that exploits semantic incongruity and valence reversal to convert non-sarcastic instances to sarcastic ones. On the other hand, while detecting irony is crucial, it is insufficient to capture the cardinal connotation of the statement. Consequently, Dubey, Joshi, and Bhattacharyya (2019) examined the task of converting sarcastic utterances into their non-sarcastic counterparts using deep neural networks.

In this work, we explore the task of Sarcasm Explanation in Dialogues, the second attempt after Kumar et al. (2022). SED aims to generate natural language explanations for a disseminated multimodal sarcastic conversation. We present a new model, MOSES, which enhances the current state-of-the-art for the SED task. However, unlike Kumar et al. (2022), we perform both intrinsic and extrinsic evaluations to show the efficacy and usefulness of our model. We leverage the generated explanations to improve three affect understanding tasks – sarcasm detection, humour identification, and emotion recognition in dialogues.

Dataset

Human conversations often take place employing a variety of languages. The phenomenon of using a blend of more than one language to communicate is dubbed code-mixing. Due to the prevalence of code-mixing in today's world, we consider the WITS dataset (Kumar et al. 2022), which contains code-mixed dialogues (English-Hindi) from an Indian TV series. The dataset comprises multimodal, multiparty, code-mixed, sarcastic conversations where each sarcastic instance is annotated with a corresponding natural language code-mixed explanation.

In order to gauge the effect of sarcasm explanation on affective attributes, we augment the WITS dataset to perform sarcasm detection, humour identification, and emotion recognition on it. We create instances for sarcastic and non-sarcastic utterances with their context to perform sarcasm detection. We call this variation of the dataset sWITS. Adapted from MASAC (Bedi et al. 2021), WITS can also be mapped to annotations for humour identification, where each utterance contains a binary marker showcasing whether the utterance is amusing or not. Consequently, we map each instance in sWITS to its corresponding humour annotation. Additionally, we determine emotion labels for the instances at hand and identify the following emotions – *sadness, joy, anger*, *and neutral*. Three annotators were involved in this phase and achieved an inter-annotator agreement of 0.86. More information can be found in the supplementary. Accordingly, we obtain four variations of the dataset:

1. WITS: It contains multimodal, multiparty, code-mixed, sarcastic instances with associated explanations.
2. sWITS: It contains sarcastic and non-sarcastic instances constructed from WITS. The last utterance of each instance is marked by a binary tag indicating whether the statement contains sarcasm or not.
3. hWITS: For each instance created in sWITS, each target utterance is marked with another binary label revealing the existence of humour in it.

	WITS	sWITS		hWITS		eWITS			
	#S	#NS	#S	#NH	#H	#Ntrl	#Sad	#Joy	#Ang
Train	1792	1669	1792	2795	995	1590	1147	623	429
Val	224	213	224	362	112	196	133	87	57
Test	224	218	224	367	106	195	141	70	67
Total	2240	2100	2240	3524	1213	1981	1421	780	553

Table 1: Statistics of the sarcasm, humour, and emotion (N: Ntrl: Neutral, Ang: Anger) datasets in consideration (number of dialogue instances marked as sarcastic (#S), non-sarcastic (#NS), non-humorous (#NH), and humorous (#H).).

4. eWITS: Similar to hWITS, this variant contains emotion labels for the target utterances.

Table 1 illustrates the elementary statistics for the explained dataset variations. More details about the dataset are present in the supplementary.

Proposed Method

This section illustrates the working of our proposed model, MOSES as presented in Figure 2. The existing SED model, MAF (Kumar et al. 2022), which uses a modified version of context-aware attention (Yang et al. 2019), takes the multimodal (audio/video) vectors as context and fuses them with the text modality to generate multimodal fused text vectors. This way of multimodal fusion makes text the primary modality and treats the other signals (acoustic and visual) as secondary. Such a fusion technique might result in the downplay of the audio and video modalities. However, in the complete duration of the discourse, modalities other than text could play the deciding role in resolving the affects in consideration. Consequently, we propose using context-aware fusion in such a way that each modality gets a chance to play a pivotal role in the fusion computation.

The existing MAF module consists of an adapter-based module comprising two modules. The two modules – Multimodal Context Aware Attention (MCA2) and Global Information Fusion (GIF) together make up the Multimodal Aware Fusion (MAF) module. Given the three input signals, namely text, audio, and video, the MCA2 module effectively introduces multimodal information in the textual representations. Further, the GIF module combines the multimodal infused textual representations. We insert another module in the pipeline, Modality Spotlight (MS), which is responsible for attending to each modality by treating it as the primary modality and the rest as the context. We explain each of these modules below.

Multimodal Context-Aware Attention (MCA2). The textual modality directly interacts with the other modalities in the standard fusion scheme, which uses dot-product based cross-modal attention. The multimodal representation acts as the key and value vectors while the text serves as the query. However, such a direct fusion of multimodal information from different embedding subspaces can lead to an inefficient representation that cannot capture contextual information. Consequently, inspired by Yang et al. (2019), a context-aware attention block is used instead of dot product attention in the MCA2 module.

The multimodal context-aware attention first generates multimodal information conditioned key and value vectors using a primary representation, H which depends on the choice of dominant modality in consideration. For example, H can be obtained from any language model, such as BART (Lewis et al. 2020), if text is to be considered as the primary modality. To generate information fused vectors, the MCA2 module first needs to convert H into key, query, and value vectors, q, k, $v \in \mathbb{R}^{n\times d}$, respectively, as illustrated in Equation 1. Here, W_q, W_k, and $W_v \in \mathbb{R}^{d\times d}$ are learnable parameters where the input, of maximum length n, is represented by a vector of dimension d.

$$[qkv] = H[W_q W_k W_v] \tag{1}$$

If we consider $M \in \mathbb{R}^{n\times d_c}$ a multimodal vector, the multimodal information infused key and value vectors k_m, $v_m \in \mathbb{R}^{n\times d}$, are generated following Equation 2. Here, $\lambda \in \mathbb{R}^{n\times 1}$ decides the amount of multimodal information to capture into the primary modality.

$$\begin{bmatrix} k_m \\ v_m \end{bmatrix} = (1 - \begin{bmatrix} \lambda_k \\ \lambda_v \end{bmatrix}) \begin{bmatrix} k \\ v \end{bmatrix} + \begin{bmatrix} \lambda_k \\ \lambda_v \end{bmatrix} (M \begin{bmatrix} U_k \\ U_v \end{bmatrix}) \tag{2}$$

The parameter λ is learned using a gating mechanism, as shown in Equation 3. Note that the matrices, U_k, $U_v \in \mathbb{R}^{d_c\times d}$, and W_{k1}, W_{v1}, W_{k1}, $W_{v2} \in \mathbb{R}^{d\times 1}$, in Equations 2 and 3 are trained with the model.

$$\begin{bmatrix} \lambda_k \\ \lambda_v \end{bmatrix} = \sigma(\begin{bmatrix} k \\ v \end{bmatrix} \begin{bmatrix} W_{k_1} \\ W_{v_1} \end{bmatrix} + M \begin{bmatrix} U_k \\ U_v \end{bmatrix} \begin{bmatrix} W_{k_2} \\ W_{v_2} \end{bmatrix}) \tag{3}$$

Once the modified key and value pair is obtained, the traditional dot-product attention is used to obtain the final multimodal fused vectors.

Modality Spotlight (MS). We discussed how we can generate multimodal infused vector representation considering one modality as primary and the rest as context. Our work deals with three modalities – text, acoustic, and visual. The spotlight module is responsible for treating each of these modalities as the primary modality at a time and generating the corresponding fused vectors. For instance, if text is considered the primary modality, then we need to calculate two multimodal fused vectors, H_{Ta} and H_{Tv}, such as audio and video, play the role of context in the representations, respectively. Similarly, when audio and video are considered the primary source of information, H_{tA} and H_{tV} are calculated. Note that we do not calculate H_{Av} or H_{aV} because we are dealing with a textual generation task where the textual information plays the preliminary role.

Apart from bi-modal interactions, we also deal with tri-modal interactions in our work, where all three modalities are infused using the GIF module. Unlike bi-modal fusion, it is unfair to let text be the only primary modality in the tri-modal fusion. Consequently, we compute three tri-modal vectors, H_{Tav}, H_{tAv}, and H_{taV}, such that text, audio, and video individually play the primary role, respectively.

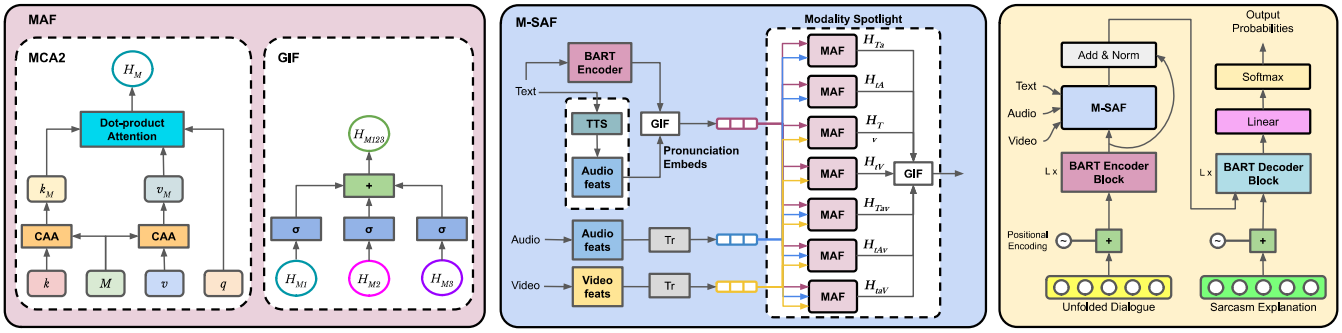

Figure 2: MOSES: The MAF model captures acoustic and visual hints using the Multimodal Context Aware Attention (MCA2) and combines them using Global Information Fusion (GIF). Each modality is kept in spotlight using the Modality Spotlight (MS) module. To capture the subjectivity in the code-mixed spellings, we propose *pronunciation embeddings*.

Global Information Fusion (GIF). The GIF module is responsible for combining the information from multiple modalities together in an efficient manner. G gates are used to control the amount of information disseminated by each modality, where $2 \leq G \leq 3$ is the number of modalities to fuse. For instance, if we calculate the interaction between the text and audio modalities with text being the primary source of information, we will first need to calculate the gated information from the audio representation using Equation 4.

$$g_a = [H \oplus H_a]W_a + b_a \tag{4}$$

where W_a and b_a are learnable matrices, and $\oplus$ denotes vector concatenation. The final representation to be passed on to the next encoder layer will be obtained using Equation 5.

$$H_{Ta} = H + g_a \odot H_a \tag{5}$$

On similar lines, if we are to calculate the tri-modal representation keeping the text as the primary modality, we first compute the gated vector for audio and video and then compute a weighted combination of the three modalities. The following sequence of equations illustrates this process,

$$g_a = [H \oplus H_a]W_a + b_a$$
$$g_v = [H \oplus H_v]W_v + b_v$$
$$H_{Tav} = H + g_a \odot H_a + g_v \odot H_v$$

Likewise, we calculate the following set of vectors: H_{Ta}, H_{tA}, H_{Tv}, H_{tV}, H_{Tav}, H_{tAv}, and H_{taV}. Further, another GIF module is used to conglomerate these seven vectors, as shown in Equation 6.

$$H_{all} = g_t \odot H + g_{Ta} \odot H_{Ta} + g_{tA} \odot H_{tA} + g_{Tv} \odot H_{Tv} + g_{tV} \odot H_{tV} + g_{Tav} \odot H_{Tav} + g_{tAv} \odot H_{tAv} + g_{taV} \odot H_{taV} \tag{6}$$

Experiment and Results

This section illustrates the feature extraction strategy we use and the baseline systems to which we compare our model, followed by the results we obtain for the SED task. We use the standard generative metrics – ROUGE-1/2/L (Lin 2004), BLEU-1/2/3/4 (Papineni et al. 2002), and METEOR (Denkowski and Lavie 2014) to capture the syntactic and semantic performance of our system. Details about the execution process and the hyperparameters used are mentioned in the supplementary.

Feature Extraction

The primary challenges for generating vector representations for the instances in WITS come from the code-mixed and multimodal aspects of the dataset. We alleviate these by proposing intelligent feature extraction methods.

Text: The textual input in WITS is present in romanised code-mixed format. Thereby, it may contain terms with the same meaning but varying spellings that are phonetically identical. For instance, the word *"main"* in Hindi (translating to *"I"* in English) can be written as *"main"* or *"mein"*. To capture the similarity between all these spelling variations, we propose using *Pronunciation Embeddings (PE)* that capture the phonetic equivalence between the words of the input text. We convert the text into a standard speech format using python's gTTS library[1]. This converted audio does not contain any tone or pitch variation for any term and thus, sounds the same for phonetically similar terms. We then extract the audio features from this converted speech using the method described below. This pronunciation vector is fused with the text representation, obtained from any encoder model like BART, using the GIF module to obtain the final text representation.

Audio: We use 154 dimension features capturing the loudness, and Mel Frequency Cepstral Coefficient (MFCCs) for each instance using the eGeMAPS model (Eyben et al. 2016). A Transformer encoder (Vaswani et al. 2017) is used for further processing of these features.

Video: These features are obtained using ResNext-101 (Hara, Kataoka, and Satoh 2018). A resolution of 720 pixels, a window length of 16, and a frame rate of 1.5 are used to obtain 2048 dimensional visual features. Analogous to the acoustic representation, a Transformer encoder captures the sequential conversation context in the resultant vectors.

Comparative Systems

We use various established sequence-to-sequence (seq2seq) models to obtain the most promising textual representations for the discourse. **RNN**: The openNMT4[2] implementation of

[1]https://pypi.org/project/gTTS/
[2]https://github.com/OpenNMT/OpenNMT-py

	Model	R1	R2	RL	B1	B2	B3	B4	M
Textual	**RNN**	29.22	7.85	27.59	22.06	8.22	4.76	2.88	18.45
	Transformer	29.17	6.35	27.97	17.79	5.63	2.61	0.88	15.65
	PGN	23.37	4.83	17.46	17.32	6.68	1.58	0.52	23.54
	mBART	33.66	11.02	31.5	22.92	10.56	6.07	3.39	21.03
	BART	36.88	11.91	33.49	27.44	12.23	5.96	2.89	26.65
Multimodal	**MAF-TA**	38.21	14.53	35.97	30.58	15.36	9.63	5.96	27.71
	MAF-TV	37.48	15.38	35.64	30.28	16.89	10.33	6.55	28.24
	MAF-TAV	39.69	17.1	37.37	33.2	18.69	12.37	8.58	30.4
	MOSES-TA	38.27	14.53	35.72	31.57	16.37	9.66	6.06	29.27
	MOSES-TV	39.62	16.78	37.48	32.69	17.76	11.01	6.89	31.65
	MOSES-TAV	40.88	18.33	38.38	33.27	18.87	12.6	8.8	31.41
	MOSES	**42.17**	**20.38**	**39.66**	**34.95**	**21.47**	**15.47**	**11.45**	**32.37**

Table 2: Experimental results (Abbreviation: R1/2/L: ROUGE1/2/L; B1/2/3/4: BLEU1/2/3/4; M: METEOR; PGN: Pointer Generator Network). Final row denotes MOSES including the pronunciation and spotlight modules.

the RNN seq2seq architecture is used to obtain the results. **Transformer** (Vaswani et al. 2017): Explanations are generated using the vanilla Transformer encoder-decoder model. **Pointer Generator Network (PGN)** (See, Liu, and Manning 2017): A combination of generation and copying mechanisms is used in this seq2seq architecture. **BART** (Lewis et al. 2020): We use the base version of this denoising autoencoder model. It has a bidirectional encoder with an autoregressive left-to-right decoder built on standard machine translation architecture. **mBART** (Liu et al. 2020): Trained on multiple large-scale monolingual corpora, mBART follows the same objective and architecture as BART[3].

Results

Textual: Table 2 shows the results obtained when textual systems are used to obtain the generated explanations. We notice that while PGN delivers us with the least performance across most metrics, BART-based representations outperform the rest by providing the best performance across the majority of all evaluation metrics.

Pronunciation Embeddings (PE): Due to the subjective nature of how other languages (Hindi, in our case) are written in a romanised format, the spellings of the words come from their phonetic understanding. To resolve the ambiguity between the same words with differing spellings, we propose to use pronunciation embeddings. As illustrated in Table 2, we observe that by adding the PE component to the model with the help of the GIF module, the performance of text-based systems jumps by an average of $\sim 4\%$ across all evaluation metrics.

Multimodality: After we obtain the representation for the code-mixed text by fusing textual representation with pronunciation embeddings, we move on to adding multimodality to the system. We experimented with an established SED method (MAF-TAV) to estimate the effect of multimodality. Table 2 exhibits that while the addition of acoustic signals does not result in a performance gain, the addition of visual cues boosts the performance by $\sim 1\%$ across all metrics. This phenomenon can be attributed to the fact that audio alone may cause confusion while understanding sarcasm, and visual hints may help in such times (Case III in Figure 1). Thereby, improving the visual feature representations can be one of the future directions. Finally, when we add all multimodal signals together, we observe the best performance yet with an average increase of further $\sim 1\%$ across majority metrics.

[3]https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt

Model	R1	R2	RL	B1	B2	B3	B4	M
BART	36.88	11.91	33.49	27.44	12.23	5.96	2.89	26.65
+concat	17.22	1.7	14.12	13.11	2.11	0.0	0.0	9.34
+DPA	36.43	13.04	33.75	28.73	14.02	8.0	4.89	25.6
+MCA2	36.37	13.85	34.92	28.49	14.34	9.0	6.16	25.75
+ GIF	39.69	17.1	37.37	33.2	18.69	12.37	8.58	30.4
+ PE	40.88	18.33	38.38	33.27	18.87	12.6	8.8	31.41
+ MS (MOSES)	**42.17**	**20.38**	**39.66**	**34.95**	**21.47**	**15.47**	**11.45**	**32.37**

Table 3: Ablation results on MOSES (Abbreviation: DPA: Dot Product Attention).

Modality Spotlight: As hypothesised, we obtain the best performance for sarcasm understanding when all the three modalities are used in tandem. However, earlier methods for SED provided limelight to only textual representations (Kumar et al. 2022). We argue that especially in the case of sarcasm, multimodal signals such as audio and video might play the principal role in many instances. To comprehend this rotating importance of modalities, we use the spotlight module that aims to treat each modality as the primary modality while calculating the final representation. We observe an increase of $\sim 2\%$ across all evaluation metrics as shown in Table 2. These results directly support our hypothesis of the effect of multimodality in sarcasm analysis.

Ablation Study

To highlight the importance of all modules in consideration, we perform extensive ablation studies on the WITS dataset. Table 3 shows the results when we add the different proposed modules to our system sequentially. The first row highlights the BART model's results for the text modality which results in a ROUGE-2 of 11.91%. As illustrated, the use of naive trimodal concatenation ($T \oplus A \oplus V$) of text, audio, and video representations produces a noisy fusion resulting in decreased performance (-10.2% ROUGE-2). Next, we try with the standard dot-product attention, which, being a comparatively smarter way of multimodal fusion, results in a slightly improved performance over the text-only modality ($+2\%$ ROUGE-2). Further, adding the multimodal context-aware attention module (MCA2) and replacing standard dot-product attention, produces a further performance boost by $\sim 1\%$ across all metrics, signifying the importance of the intelligent fusion that the MCA2 module provides us. The performance is increased even more when the GIF module is introduced to compute the final multimodal vector representation ($+4\%$ ROUGE-2), signifying the positive effect gated fusion has on efficient multimodal representations. Next, we incorporate pronunciation embeddings (PE) into the model and observe another performance boost across majority metrics ($\sim 1\%$), suggesting that we can obtain better code-mixed representations by reducing the spelling am-

Dialogue	Ground Truth	MAF	MOSES
KISMI: Bas na Sahil bhai, meri firki kheech rahe ho na!? *(Enough brother Sahil, are you teasing me?!)* SAHIL: Nahi, nahi, kya hai ki, mere CD ki collection mein na, ye train ke awaaj vali CD nahi hai... *(No no, see I don't have train's sound in my CD collection...)*	Sahil Kismi ko taunt maarta hai kyuki use rail gaadi ki awaaj sunni hai. *(Sahil taunts Kismi that she wants to hear the sound of a train)*	Sahil Kismi ko taunt maarta hai ki use pasand nahi. *(Sahil taunts Kismi that he doesn't like)*	Sahil Kismi ko taunt maarta hai kyuki use rail gaadi ki awaaj sunni hai. *(Sahil taunts Kismi that she wants to hear the sound of a train)*
MONISHA: Say hello to Tommy the dog. *(Say hello to Tommy the dog.)* MAYA: Tumne iss kutte ka naam Tommy the dog rakha? *(Did you name your dog Tommy the dog?)*	Maya monisha ko tana marti hai kyunki usne apne kutte ka naam tommy the dog rakha hai. *(Maya taunts Monisha on naming her dog Tommy the dog.)*	Maya kehti hai ki uske kutte ka naam tommy the dog rakha hai. *(Maya says that her dog's name is Tommy the dog.)*	Maya taunts monisha kyunki usne apne kutte ka naam tommy the dog rakha hai. *(Maya taunts Monisha that she has named her dog Tommy the dog.)*

Table 4: Generated samples from test set. The last utterance is the sarcastic utterance for each dialogue.

	mBART	BART	MAF	MOSES
Source	75	77.23	**91.07**	90.17
Target	45.33	52.67	46.42	**56.69**

Table 5: Accuracy for the sarcasm source and target for BART-based systems.

	Coherency	On topic	Capturing sarcasm
mBART	2.57	2.66	2.15
BART	2.73	2.56	2.18
MAF	3.03	3.11	2.77
MOSES	**3.96**	**3.27**	**3.10**

Table 6: Human evaluation.

biguities. Finally, our entire model with modality spotlight included produces the best performance, verifying the necessary use of each module discussed.

Result Analysis

Quantitative Analysis. MOSES is evaluated on its ability to capture sarcasm source and target in the generated explanations. We compare MOSES with mBART, BART, and MAF. Table 5 shows that BART performs better than mBART for both source and target detection. The inclusion of multimodal signals, even without pronunciation embeddings and modality spotlight, improves the source identification performance by $\sim 14\%$. MOSES is able to detect the sarcasm source most efficiently, resulting in an improvement of $\sim 4\%$ over the next best result. Consequently, we can relate the presence of multimodal capabilities to capture speaker-specific peculiarities more efficiently, resulting in better source/target identification.

Qualitative Analysis. We sample a few dialogues from the test set of WITS and show their generated explanations by MOSES and the best baseline, MAF along with the ground-truth explanations in Table 4. We show one of the many instances where our model generates the correct explanation for the given sarcastic instance in the first row. The last row, highlights a case where the generated explanation is not syntactically similar to the ground-truth explanation but resembles it semantically. To evaluate the semantic similarity properly, we perform a human evaluation.

Human Evaluation. We sample a total of 25 random instances from the test set and ask 20 human evaluators[4] to evaluate the generated explanations (on a scale of 1 to 5) on the following basis: **Coherence:** Checks the generated explanation for correct structure and grammar. **On topic:** Measures the extent to which the generated explanation revolves around the dialogue topic. **Capturing sarcasm:** Estimates the level of emitted sarcasm being captured in the generated output.

We show the average score for the human evaluation parameters in Table 6. As illustrated, the proposed MOSES model exhibits more coherent, on topic, and sarcasm related explanations. However, there is still a scope for improvement, which can be taken up as future work.

[4]The evaluators are fluent in English and their age ranges in 25-30 years.

Understanding Affects with Explanation

We study three understanding tasks in dialogues – sarcasm detection, humour identification, and emotion recognition using sWITS, hWITS, and eWITS, respectively. A trained SED system is used to obtain the explanations for all the instances present in these datasets. We show the qualitative analysis of the generated explanations by MOSES in the supplementary. To verify our hypothesis that sarcasm explanation helps affect understanding, we perform experiments with and without explanations, as explained in the subsequent sections.

Sarcasm Detection. We take a base RoBERTa model (Liu et al. 2019) and perform the task of sarcasm detection over sWITS. The experimentation is performed using three setups as described below:

1. When we do not provide any utterance explanation to the input dialogue.
2. When we provide utterances appended with their generated explanation at the training time. Plain dialogues are given at the testing time in this case.
3. When dialogue instances are appended with their corresponding explanations during train and test time.

Model	Use of Expl		Sarcasm				Humor				Emotion		
	Train	Test	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
None	**0**	**0**	0.57	0.68	0.62	0.57	0.69	0.78	0.73	0.87	0.8	0.78	0.78
MAF	**1**	**0**	0.58	0.73	0.65	0.6	0.57	**0.87**	0.69	0.81	0.78	0.78	0.78
	1	**1**	0.66	0.77	0.71	0.68	0.73	0.71	0.72	0.87	0.78	**0.81**	0.79
MOSES	**1**	**0**	0.65	0.71	0.68	0.66	**0.84**	0.63	0.72	**0.89**	0.79	0.78	0.78
	1	**1**	**0.70**	**0.83**	**0.76**	**0.73**	0.72	0.77	**0.75**	0.88	**0.81**	0.80	**0.80**

Table 7: Experimental results on RoBERTa base when explanations generated by MOSESand MAF are used for completing the respective tasks. The first row indicates the performance without explanation.

Table 7 illustrates the results we obtain for all the settings for MOSES and the best baseline, MAF. As can be seen, RoBERTa obtains 62% F1 score when we do not use any explanations. However, with the use of the generated explanations by MOSES during the train time, we obtain an improvement of 6% F1-score. On the other hand, the best performance is achieved by the last case, where the input instances are appended with their corresponding explanations both at the train and test time, with an increase of 8% F1-score. Consistent to the results obtained by MOSES's generation, MAF also reports an improved performance over no explanation model. However, the improvement shown by MAF is not at par with the improvement obtained by MOSES. These results directly support our hypothesis that utterance explanations can assist an efficient detection of sarcasm in the input instances.

Humour Identification. Another RoBERTa base is used to perform humour identification on hWITS. As for sarcasm detection, humour identification is also evaluated for the three setups described in the previous section. Table 7 illustrates the results obtained for the described setups. When no explanations are used during the training or testing time, we get an F1-score of 73%. This score is comparable to the performance we get when input instances are appended with their corresponding explanations generated by MOSES at the training time. This performance is boosted by 3% when the explanations are provided at the train/test time. However, it is important to note that the explanations generated by the MAF model resulted in a slightly decreased performance indicating the superiority of MOSES.

Emotion Recognition. Table 7 illustrates the results obtained for the task of emotion recognition on eWITS. We see the same value for the weighted F1 when we add explanations during the training phase of the system for both MAF and MOSES. However, when explanations assist both the training and testing phase, we observe an increase of 2% in the weighted F1 score for MOSESand 1% increase for MAF, indicating the positive effect explanations deliver for emotion recognition. Performance analysis for sarcastic and non-sarcastic instances can be found in the supplementary.

Error Analysis

Quantitative. To capture the improvement exhibited by explanations in affect understanding, we show the confusion matrices emitted by the understanding models with and without using explanations. Table 8 illustrates these matrices – and as can be seen, the methods with explanation obtains higher true positive rate with a decreased false positive and false negative rates for majority of the classes among sarcasm, humour, and emotion labels.

	NS	S
NS	137/100	81/117
S	39/70	185/153

(a) Sarcasm detection on sWITS.

	NH	H
NH	335/330	32/37
H	24/23	82/83

(b) Humour identification on hWITS.

	Neutral	Sadness	Joy	Anger
Neutral	148/137	13/23	18/19	16/16
Sadness	5/2	62/66	3/2	0/0
Joy	7/5	10/9	120/124	4/3
Anger	0/9	0/1	8/9	50/48

(c) Emotion recognition on eWITS.

Table 8: Confusion matrix of the systems with and without (with/without) explanations.

Dialogue	**MAYA:** And this time I thought lets have a theme party! *(And this time I thought lets have a theme party!)* **MONISHA:** Animals! Hum log sab animals banenge! *(Animals! Let us all be animals this time!)* **MAYA:** Mai hiran, Sahil horse, and Monisha chhipakalee! *(I'll be a deer, Sahil a horse, and Monisha can be a lizard!)*		
Exp	Maya Monisha ko animal keh ke taunt maarti hai. *(Maya taunts Monisha by calling her an animal)*		
	Sarcasm	**Humour**	**Emotion**
GT	1	0	Anger
w/o Exp	0	1	Neutral
w Exp	1	0	Anger

Table 9: True and predicted labels for the three affect tasks with and without using MOSES's explanation.

Qualitative. While quantitative results confirm that explanations assist in identifying affects efficiently, qualitative analysis can further corroborate this hypothesis. Table 9 shows one instance from the test set where the presence of explanation helps for all affective tasks in question. More such examples can be found in the supplementary.

Conclusion

The inability of existing systems to understand sarcasm results in a performance gap for various affect understanding tasks like emotion recognition, humour identification, and sarcasm detection. To mitigate this issue, we proposed MOSES to explore the task of Sarcasm Explanation in Dialogues (SED). MOSES takes multimodal code-mixed sarcastic conversation instances as input and results in a natural language explanation describing the sarcasm present in it. We further explored the effect of the generated explanations on three dialogue-based affect understanding tasks – sarcasm detection, humour identification, and emotion recognition. We observed that explanations improved the performance of all three tasks, thus verifying our hypothesis.

Acknowledgements

The authors acknowledge the support of the ihub-Anubhuti-iiitd Foundation, set up under the NM-ICPS scheme of the DST, and CAI-IIITD.

References

Abu Farha, I.; and Magdy, W. 2020. From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 32–39.

Bedi, M.; Kumar, S.; Akhtar, M. S.; and Chakraborty, T. 2021. Multi-modal Sarcasm Detection and Humor Classification in Code-mixed Conversations. *IEEE Transactions on Affective Computing*.

Bharti, S. K.; Sathya Babu, K.; and Jena, S. K. 2017. Harnessing Online News for Sarcasm Detection in Hindi Tweets. In Shankar, B. U.; Ghosh, K.; Mandal, D. P.; Ray, S. S.; Zhang, D.; and Pal, S. K., eds., *Pattern Recognition and Machine Intelligence*, 679–686.

Cai, Y.; Cai, H.; and Wan, X. 2019. Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2506–2515.

Castro, S.; Hazarika, D.; Pérez-Rosas, V.; Zimmermann, R.; Mihalcea, R.; and Poria, S. 2019. Towards Multimodal Sarcasm Detection (An _Obviously_ Perfect Paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4619–4629.

Chakrabarty, T.; Ghosh, D.; Muresan, S.; and Peng, N. 2020. R^3: Reverse, Retrieve, and Rank for Sarcasm Generation with Commonsense Knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7976–7986.

Chauhan, D. S.; S R, D.; Ekbal, A.; and Bhattacharyya, P. 2020. Sentiment and Emotion help Sarcasm? A Multi-task Learning Framework for Multi-Modal Sarcasm, Sentiment and Emotion Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4351–4360.

Cignarella, A. T.; Frenda, S.; Basile, V.; Bosco, C.; Patti, V.; Rosso, P.; et al. 2018. Overview of the evalita 2018 task on irony detection in italian tweets (ironita). In *Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2018)*, 1–6. CEUR-WS.

Denkowski, M.; and Lavie, A. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 376–380.

Dubey, A.; Joshi, A.; and Bhattacharyya, P. 2019. Deep Models for Converting Sarcastic Utterances into Their Non Sarcastic Interpretation. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 289–292.

Eyben, F.; Scherer, K. R.; Schuller, B. W.; Sundberg, J.; André, E.; Busso, C.; Devillers, L. Y.; Epps, J.; Laukka, P.; Narayanan, S. S.; and Truong, K. P. 2016. The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *IEEE Transactions on Affective Computing*, 190–202.

Ghosh, D.; Richard Fabbri, A.; and Muresan, S. 2017. The Role of Conversation Context for Sarcasm Detection in Online Interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 186–196.

Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6546–6555.

Hasan, M. K.; Lee, S.; Rahman, W.; Zadeh, A.; Mihalcea, R.; Morency, L.-P.; and Hoque, E. 2021. Humor Knowledge Enriched Transformer for Understanding Multimodal Humor. *Proceedings of the AAAI Conference on Artificial Intelligence*, 12972–12980.

Joshi, A.; Bhattacharyya, P.; and Carman, M. J. 2017. Automatic Sarcasm Detection: A Survey. *ACM Comput. Surv.*, 1–22.

Joshi, A.; Sharma, V.; and Bhattacharyya, P. 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)*, 757–762.

Kreuz, R.; and Caucci, G. 2007. Lexical Influences on the Perception of Sarcasm. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, 1–4.

Kumar, S.; Kulkarni, A.; Akhtar, M. S.; and Chakraborty, T. 2022. When did you become so smart, oh wise one?! Sarcasm Explanation in Multi-modal Multi-party Dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5956–5968.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81.

Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 726–742.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mishra, A.; Tater, T.; and Sankaranarayanan, K. 2019. A Modular Architecture for Unsupervised Sarcasm Generation. 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), 6144–6154.

Olkoniemi, H.; Ranta, H.; and Kaakinen, J. K. 2016. Individual differences in the processing of written sarcasm and metaphor: Evidence from eye movements. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 42(3): 433–450.

Ortega-Bueno, R.; Rangel, F.; Hernández Farıas, D.; Rosso, P.; Montes-y Gómez, M.; and Medina Pagola, J. E. 2019. Overview of the task on irony detection in Spanish variants. In *Proceedings of the Iberian languages evaluation forum (IberLEF 2019), co-located with 34th conference of the Spanish Society for natural language processing (SEPLN 2019). CEUR-WS. org*, 229–256.

Pan, H.; Lin, Z.; Fu, P.; Qi, Y.; and Wang, W. 2020. Modeling Intra and Inter-modality Incongruity for Multi-Modal Sarcasm Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1383–1392.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 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.

Peled, L.; and Reichart, R. 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)*, 1690–1700.

Roberts, R. M.; and Kreuz, R. J. 1994. Why Do People Use Figurative Language? *Psychological Science*, 5(3): 159–163.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1083.

Srivastava, H.; Varshney, V.; Kumari, S.; and Srivastava, S. 2020. A Novel Hierarchical BERT Architecture for Sarcasm Detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, 93–97. Association for Computational Linguistics.

Swami, S.; Khandelwal, A.; Singh, V.; Akhtar, S. S.; and Shrivastava, M. 2018. A corpus of english-hindi code-mixed tweets for sarcasm detection. *arXiv preprint arXiv:1805.11869*.

Tay, Y.; Luu, A. T.; Hui, S. C.; and Su, J. 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)*, 1010–1020.

Tsur, O.; Davidov, D.; and Rappoport, A. 2010. ICWSM — A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews. *Proceedings of the International AAAI Conference on Web and Social Media*, 162–169.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Xiong, T.; Zhang, P.; Zhu, H.; and Yang, Y. 2019. Sarcasm Detection with Self-Matching Networks and Low-Rank Bilinear Pooling. In *The World Wide Web Conference*, 2115–2124.

Xu, N.; Zeng, Z.; and Mao, W. 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*, 3777–3786.

Yang, B.; Li, J.; Wong, D. F.; Chao, L. S.; Wang, X.; and Tu, Z. 2019. Context-Aware Self-Attention Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 387–394.