Just Like a Human Would, Direct Access to Sarcasm Augmented with Potential Result and Reaction

Changrong Min[a], Ximing Li*[b,c], Liang Yang[a], Zhilin Wang[b,c], Bo Xu[a], Hongfei Lin[a]
[a]School of Computer Science and Technology, Dalian University of Technology, China
[b]College of Computer Science and Technology, Jilin University, China
[c]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China
11909060@mail.dlut.edu.cn, liximing86@gmail.com
wangzl5521@mails.jlu.edu.cn, (hflin,liang,xubo)@dlut.edu.cn

Abstract

Sarcasm, as a form of irony conveying mockery and contempt, has been widespread in social media such as Twitter and Weibo, where the sarcastic text is commonly characterized as an incongruity between the surface positive and negative situation. Naturally, it has an urgent demand to automatically identify sarcasm from social media, so as to illustrate people's real views toward specific targets. In this paper, we develop a novel sarcasm detection method, namely Sarcasm Detector with Augmentation of Potential Result and Reaction (SD-APRR). Inspired by the direct access view, we treat each sarcastic text as an incomplete version without latent content associated with implied negative situations, including the result and human reaction caused by its observable content. To fill the latent content, we estimate the potential result and human reaction for each given training sample by `[xEffect]` and `[xReact]` relations inferred by the pre-trained commonsense reasoning tool COMET, and integrate the sample with them as an augmented one. We can then employ those augmented samples to train the sarcasm detector, whose encoder is a graph neural network with a denoising module. We conduct extensive empirical experiments to evaluate the effectiveness of SD-APRR. The results demonstrate that SD-APRR can outperform strong baselines on benchmark datasets.

1 Introduction

Sarcasm, as subtle figures of speech, serves many communicative purposes in human daily life (Ivanko and Pexman, 2003), commonly used to criticize an individual. Refer to the formal description of sarcasm from the Oxford English Dictionary:[1]

"A way of using words that are the opposite of what you mean in order to be unpleasant to somebody or to make fun of them."

The sarcastic text is typically characterized as an incongruity between the positive surface and negative situation (Riloff et al., 2013; Liu et al., 2022). For example, as an obvious sarcasm *"I love working for six hours every day for free"*, its surface meaning tends to be positive, conveyed by the sentiment word "love", but it corresponds to a negative situation "work for free", conveying people's complaint.

Detecting sarcasm from social media is a significant task due to the universal existence of sarcasm, but its complicated nature makes the task challenging. To resolve this task, the community has recently proposed a number of **S**arcasm **D**etection (**SD**) methods, whose major idea is to capture the incongruity characteristic of sarcasm (Joshi et al., 2017; Xiong et al., 2019; Pan et al., 2020; Agrawal et al., 2020; Li et al., 2021b; Lou et al., 2021). For example, several early SD studies express the incongruity by extracting positive-negative pairs from observable text content, such as rule-based method (Joshi et al., 2017) and neural networks with co-attention tricks (Xiong et al., 2019; Pan et al., 2020). Unfortunately, those methods cannot accurately capture the negative situations, which are mostly implied and associated with contexts and background information. To alleviate this issue, the recent arts of SD express the negative situations with external knowledge bases. From the perspective of sentiments, some SD methods employ auxiliary affective lexicons, *e.g.,* SenticNet (Cambria et al., 2020), to estimate the implied affective correlations among words and phrases of samples (Agrawal et al., 2020; Lou et al., 2021). Additionally, the SarDeCK method (Li et al., 2021b) employs the pre-trained commonsense reasoning tool COMET (Hwang et al., 2021) to infer the relations behind samples as their implied situations. Despite the promising performance, their expressions of implied negative situations are still a bit abstract and impalpable.

* Corresponding Author
[1]https://www.oed.com

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 10172–10183
July 9-14, 2023

Table 1: Examples of sarcastic texts and the corresponding potential results and human reactions reasoned by COMET.

ID	Text	Result	Human Reaction
1	I love people that make me feel so shit about myself.	gets hurt.	sad
2	Oh joy another drive by with absolutely no proof or evidence.	goes to jail.	angry
3	Sound night when your bathroom floor falls through into the kitchen sink.	gets dirty.	scared

As complicated figures of speech, we are particularly interested in **how do human beings accurately identify sarcasm?** Through referring to the prior psychological, cognitive, and linguistic literature (Gibbs, 1986; W.Gibbs, 2002; Ivanko and Pexman, 2003), we are agreeable with two significant viewpoints. First, the negative situations of sarcasm are mostly associated with certain social events (Pickering et al., 2018), and human beings can often easily identify the events with the background information in the brain. Second, from the direct access view (Giora and Fein, 1999; W.Gibbs, 2002; Ivanko and Pexman, 2003), human beings are likely to directly understand the whole sarcastic text with both literal meanings and implied negative situations, which can be easily captured by them.

Based on the analysis, what we expect is to develop a novel SD method by simulating the way of human thinking. Inspired by the direct access view, we treat each sarcastic text as an incomplete version without latent content associated with implied negative situations. We can use the associated social events to express the negative situations due to their strong connection. Further, we assume the social events can be mainly expressed by the potential results and human reactions that the events produced (see examples in Table 1). Accordingly, for each given sample we can estimate its potential result and human reaction by pre-trained commonsense reasoning tools (acted as background information), and then integrate the observable text content with them as an augmented sample (acted as the whole text). Finally, we can use those augmented samples to train the sarcasm detector (just like a human would).

Upon these ideas, we propose a novel SD method, namely **S**arcasm **D**etector with **A**ugmentation of **P**otential **R**esult and **R**eaction (**SD-APRR**). Specifically, we estimate the potential result and human reaction for each training sample by `[xEffect]` and `[xReact]` relations inferred by the auxiliary commonsense reasoning tool COMET (Hwang et al., 2021), and then integrate the sample with them to generate an augmented one, dubbed as **event-augmented sample**. By analogy to (Lou et al., 2021; Liang et al., 2022), we assume that the syntactic information of event-augmented samples can intuitively imply the incongruity of sarcasm. Accordingly, we transform each event-augmented sample into a dependency graph (Nivre, 2003), and suggest a graph-based encoder to generate sample embeddings. Additionally, to resolve the noisy results and reactions inferred by COMET, we suggest a denoising module with the dynamic masking trick (Yang et al., 2021), enabling to improve the quality of sample embeddings. With those embeddings, a single-layer MLP is used as the sarcastic classifier finally. To examine the effectiveness of SD-APRR, we conduct extensive experiments on benchmark datasets. The empirical results demonstrate that SD-APRR can outperform the existing baseline methods.

The contributions of this work can be summarized as follows:

- We propose a novel SD method, named SD-APRR, with event-augmented samples formed by the auxiliary commonsense reasoning tool COMET.
- We suggest a graph-based encoder with a denoising module, enabling to generate strong sample embeddings.
- The experimental results indicate that SD-APRR can achieve competitive performance compared with existing baselines.

2 Related Works

2.1 Sarcasm Detection

Early SD methods are mostly based on special rules and evidence (Maynard and Greenwood, 2014; Bharti et al., 2015; Riloff et al., 2013). For instance, the study (Maynard and Greenwood, 2014) treats the hashtag sentiment as the key indicator of sarcasm since the hashtags are usually taken

to highlight sarcasm in Tweets; and other methods employ various evidence, such as parser-based negative phrase matching, interjections (Bharti et al., 2015), and positive-negative word pairs (Riloff et al., 2013). Some other methods form incongruity-specific embeddings for sarcastic texts, such as shape and pointedness of words (Ptáček et al., 2014), extensions of words (Rajadesingan et al., 2015), and unexpectedness (Reyes et al., 2012).

Due to the success of neural networks, the mainstream SD methods nowadays apply them to capture the incongruity between positive surface and negative situations within the sarcastic text. Early methods mainly capture the incongruity from the observable text content (Tay et al., 2018; Xiong et al., 2019; Pan et al., 2020). For instance, the methods of (Xiong et al., 2019; Pan et al., 2020) extract positive-negative word pairs and phrase pairs with co-attention tricks. However, those methods cannot fully understand the negative situation due to its implicit nature. To resolve this issue, the recent methods employ external resources to capture negative situations and further incongruities of sarcastic texts (Agrawal et al., 2020; Lou et al., 2021; Li et al., 2021b; Liu et al., 2022). For example, the ADGCN method (Lou et al., 2021) employs the affective lexicon SenticNet (Cambria et al., 2020) to represent intra-sentence affective relations; and the DC-Net method (Liu et al., 2022) exploits sentiment lexicon to separate literal meanings from texts and further estimates sentiment conflicts. Orthogonal to the aforementioned methods, our SD-APRR forms augmented samples by commonsense reasoning and treats the augmented ones as the whole versions of sarcastic texts from the direct access view (Giora and Fein, 1999; W.Gibbs, 2002; Ivanko and Pexman, 2003).

2.2 Commonsense Knowledge Graph

Large-scale commonsense knowledge graphs (Lin et al., 2019; Yin et al., 2022) can conduct reasoning for texts to infer the commonsense knowledge behind them, and they have been widely applied to a wide range of natural language processing tasks, such as dialogue generation (Sabour et al., 2022), relation classification (Hosseini et al., 2022), and emotion recognition (Li et al., 2021a). To our knowledge, some representatives include ConceptNet (Speer et al., 2017), ATOMIC (Sap et al., 2019), and TransOMCS (Zhang et al., 2020). The ConceptNet contains 3.4M entity-relation tuples and about 90% of these tuples are taxonomic and lexical knowledge, resulting in relatively smaller commonsense portion. The recent ATOMIC contains 880K of tuples with 9 relations, which covers social commonsense knowledge including effects, needs, intents, and attributes of the actors in an event. In addition, the TransOMCS contains 18.5M tuples collected from various web sources, and the relations are similar to the ConceptNet.

Table 2: Summary of important notations

Notation	Description
N	number of training samples
s	raw text of the training sample
y	category label of the training sample
M	number of word tokens in s
e^r	event result inferred by COMET
e^h	human reaction inferred by COMET
s^e	event-augmented sample
M^e	number of word tokens in s^e
$\mathcal{G}$	dependency graph of s^e
$\mathbf{A}$	adjacency matrix of $\mathcal{G}$
$\mathbf{W}_b$	parameter of Bi-LSTM
$\mathbf{W}_{n,m,f}$	parameters of the encoder
$\mathbf{W}_c$	parameter of the sarcastic classifier
$\mathbf{H}$	node embeddings of $\mathcal{G}$
z	sample embedding of s^e

3 The Proposed SD-APRR Method

In this section, we briefly describe the task definition of SD and the commonsense reasoning tool COMET. We then introduce the proposed **SD-APRR** method in more detail. For clarity, we summarize the important notations in Table 2.

Task definition. Given N labeled training samples, the goal of SD is to induce a sarcasm detector enabling to distinguish whether a text sample belongs to sarcasm or not. Formally, each training sample is represented by (s_i, y_i), where $s_i = \{w_{i1}, \cdots, w_{iM}\}$ is the raw text and $y_i \in \mathcal{Y}$ is the category label. The label space is commonly defined as $\mathcal{Y} = \{\text{sarc}, \text{non-sarc}\}$.

Brief description of COMET. The COMET (Hwang et al., 2021) is a pre-trained commonsense reasoning tool, which can infer various kinds of commonsense relations associated with the related event of a given text. It totally contains 23 commonsense relations defined in $\textbf{ATOMIC}^{20}_{20}$. For examples, `[xWant]` describes post-condition de-

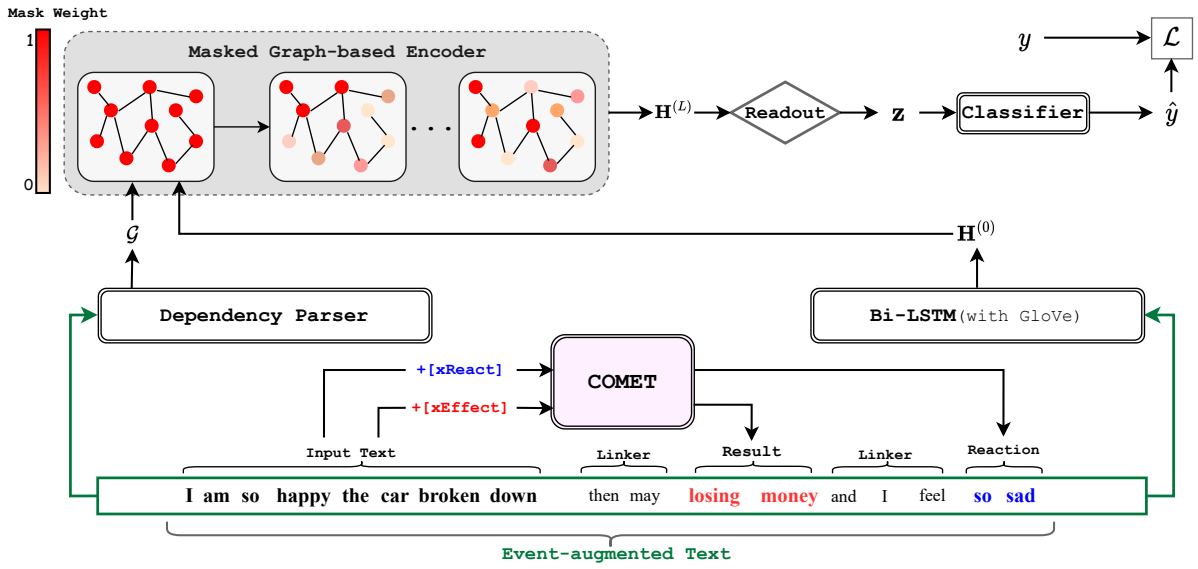

Figure 1: The framework of SD-APRR. We generate event-augmented samples by COMET, and learn the sample embeddings with the masked graph-based encoder, and finally use a single-layer MLP as the sarcastic classifier.

sires of speakers; and [xReason] gives a post-fact explanation of the cause of an event. Here, we specially introduce [xEffect] and [xReact], where [xEffect] provides social results that may occur after an event, while [xReact] provides the speakers' emotional reactions to an event. The outputs of the two relations can be directly used as the auxiliary augmentation in SD-APRR.

We declare that the COMET takes the large version of BART (Lewis et al., 2020) as the backbone, which contains 24 layers, 1024-dimensional hidden embeddings, and 16 heads for self-attention.[2] Then it was fine-tuned over $\textbf{ATOMIC}^{20}_{20}$.

3.1 Overview of SD-APRR

As depicted in Fig.1, our SD-APRR mainly consists of three components. (1) **Event-augmented samples generation**: For each raw text s_i, we employ COMET to infer its result e^r_i and human reaction e^h_i, and then concatenate them to form the corresponding event-augmented sample s^e_i. (2) **Maksed graph-based encoder**: For each event-augmented sample s^e_i, we transform it into a dependency graph $\mathcal{G}_i$, and encode $\mathcal{G}_i$ as the sample embedding $\mathbf{z}_i$ by leveraging a graph neural network encoder with dynamic masking. (3) **Sarcastic classifier**: With $\mathbf{z}_i$, we predict the category label by employing a single-layer MLP finally. In the following, we introduce each component of SD-APRR in more detail.

[2]https://huggingface.co/facebook/bart-large

3.2 Event-Augmented Samples Generation

For each raw text $s_i = \{w_{i1}, \cdots, w_{iM}\}$, we feed it into the pre-trained COMET with the [xEffect] and [xReact] relations, and treat the outputs $e^r_i = \{\overline{w}_{i1}, \cdots, \overline{w}_{i\overline{M}}\}$ and $e^h_i = \{\widetilde{w}_{i1}, \cdots, \widetilde{w}_{i\widetilde{M}}\}$ as the result and human reaction of the implied social event behind s_i. We then concatenate them to form its event-augmented version. For semantic coherence, we further leverage two linkers l^r and l^h, where l^r denotes "*then may*" for e^r_i and l^r denotes "*and I feel*" for e^h_i. Accordingly, the final event-augmented sample is formed by $s^e_i = s_i \oplus l^e \oplus e^r_i \oplus l^h \oplus e^h_i$, where $\oplus$ denotes the concatenation operator, and it totally contains M^e word tokens, where $M^e = M + \overline{M} + \widetilde{M} + 5$. We have shown the example in Fig.1.

3.3 Masked Graph-based Encoder

Given event-augmented samples $\{s^e_i\}^N_{i=1}$, we suggest a masked graph-based encoder to induce their embeddings $\{\mathbf{z}_i\}^N_{i=1}$.

3.3.1 Constructing Graphs of Samples

By analogy to (Lou et al., 2021; Liang et al., 2022), we assume that the syntactic information of event-augmented samples can intuitively imply the incongruity of sarcasm. Accordingly, we transform each s^e_i into an undirected graph $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i\}$ with the off-the-shelf dependency parsing tool,[3] where $\mathcal{V}_i$

[3]In this work, we employ the off-the-shelf syntax toolkit available at the website "https://spacy.io/".

is the set of nodes, *i.e.,* the tokens occurring in s_i^e, and $\mathcal{E}_i$ is the set of edges computed by dependency parsing. Define $\mathbf{A}_i \in \{0,1\}^{M^e \times M^e}$ as its corresponding adjacency matrix, and 1/0 denotes the component corresponds to an edge or not. Besides, each node is with self-loop.

3.3.2 Initializing Node Embeddings

For each $\mathcal{G}_i$, we initialize its node embeddings $\mathbf{H}_i^{(0)} = [\mathbf{h}_{i1}^{(0)}, \cdots, \mathbf{h}_{iM^e}^{(0)}]^\top$ by leveraging a single-layer Bi-LSTM (Hochreiter and Schmidhuber, 1997). Specifically, we represent the nodes $\mathbf{X}_i = [\mathbf{x}_{i1}, \cdots, \mathbf{x}_{iM^e}]^\top$ by the pre-trained GloVe word embeddings, and then feed $\mathbf{X}_i$ into the Bi-LSTM as follows:

$$\mathbf{H}_i^{(0)} = \text{Bi-LSTM}(\mathbf{X}_i; \mathbf{W}_b), \tag{1}$$

where $\mathbf{W}_b$ is the trainable parameter of Bi-LSTM.

3.3.3 Learning Sample Embeddings with Dynamic Masking

Given each pair $\{\mathcal{G}_i, \mathbf{H}_i^{(0)}\}$, we optimize the node embeddings $\mathbf{H}_i^{(l)} = [\mathbf{h}_{i1}^{(l)}, \cdots, \mathbf{h}_{iM^e}^{(l)}]^\top$ by a L-layer graph neural network encoder with dynamic masking (Yang et al., 2021), and then form the final sample embedding $\mathbf{z}_i$ by leveraging the readout operator with $\mathbf{H}_i$.

To be specific, the learning process of node embeddings for each layer is formulated below:

$$\mathbf{h}_{ij}^{(l)} = \text{ReLU}\left(\mathbf{W}_n^{(l)} m_{ij}^{(l)} \left[\mathbf{h}_{ij}^{(l-1)} \oplus \mathbf{h}_{\mathcal{N}(ij)}^{(l-1)}\right]\right), \quad j = 1, \cdots, M^e, \ \ l = 1, \cdots, L, \tag{2}$$

where $\mathbf{W}_n = \{\mathbf{W}_n^{(l)}\}_{l=1}^L$ are the trainable parameters; $m_{ij} \in [0,1]$ is the mask weight of the j-th node, used to capture the possible noisy e_i^r and e_i^h inferred by COMET; $\mathcal{N}(ij)$ denotes the neighbor set of the j-th node; and $\mathbf{h}_{\mathcal{N}(ij)}^{(l-1)} = \sum_{k \in \mathcal{N}(ij)} m_{ik}^{(l-1)} \mathbf{h}_{ik}^{(l-1)}$ is the weighted sum of the neighbors of the j-th node.

The update process of the mask weights for each layer is formulated below:

$$m_{ij}^{(l)} = \text{Sigmoid}\left(\mathbf{W}_m^{(l)} \hat{\mathbf{h}}_{ij}^{(l-1)} \oplus \mathbf{W}_f^{(l)} \mathbf{h}_{\mathcal{N}(ij)}^{(l-1)}\right) \quad j = 1, \cdots, M^e, \ \ l = 1, \cdots, L, \tag{3}$$

where $\mathbf{W}_m = \{\mathbf{W}_m^{(l)}\}_{l=1}^L$ and $\mathbf{W}_f = \{\mathbf{W}_f^{(l)}\}_{l=1}^L$ are the trainable parameters; and $\hat{\mathbf{h}}_{ij}^{(l-1)} = m_{ij}^{(l-1)} \mathbf{h}_{ij}^{(l-1)}$.

After obtaining the node embeddings $\mathbf{H}_i^{(L)}$ of the last layer, we can form the sample embedding $\mathbf{z}_i$ by leveraging the readout operator as follows:

$$\mathbf{z}_i = \frac{1}{M^e} \sum\nolimits_{i=1}^{M^e} \mathbf{h}_i^{(L)} \tag{4}$$

3.4 Sarcastic Classifier and Training Objective

Given the sample embeddings $\{\mathbf{z}_i\}_{i=1}^N$, we employ a single-layer MLP as the sarcastic classifier. For each $\mathbf{z}_i$, we predict its category label $\hat{y}_i$ by the following equation:

$$\hat{y}_i = \text{Softmax}\left(\mathbf{W}_c \mathbf{z}_i\right), \tag{5}$$

where $\mathbf{W}_c$ is the trainable parameter of the sarcastic classifier.

Consider N training pairs $\{(\mathbf{z}_i, y_i)\}_{i=1}^N$, we can formulate the full objective of SD-APRR with respect to all trainable parameters $\mathbf{W} = \{\mathbf{W}_b, \mathbf{W}_n, \mathbf{W}_m, \mathbf{W}_f, \mathbf{W}_c\}$:

$$\mathcal{L}(\mathbf{W}) = \sum_{i=1}^{N} \mathcal{L}_{\text{CE}}(y_i, \hat{y}_i) + \lambda \|\mathbf{W}\|^2, \tag{6}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss; $\|\cdot\|$ denotes the ℓ_2-norm; and $\lambda \in [0,1]$ is the regularization coefficient.

4 Experiment

4.1 Experimental Settings

Datasets. To thoroughly evaluate the performance of SD-APRR, we conduct experiments on four publicly available SD datasets with different scales. Their statistics of are shown in Table 3, and they are briefly described below:

- **SemEval18** is collected in SemEval 2018 Task 3 Subtask A (Van Hee et al., 2018).
- **iSarcasm** (Oprea and Magdy, 2020) consists of tweets written by participants of an online survey and thus is for intented sarcasm detection.
- **Ghosh** (Ghosh and Veale, 2016) is collected from Twitter and leverages hashtag to automatically annotate samples.
- **IAC-V2** (Abbott et al., 2016) is sourced from online political debates forum.[4] Compared with other datasets, the samples of IAC-V2 are relatively longer and more normative.

[4] http://www.4forums.com/political/

Table 3: Statistics of the benchmark datasets. #Avg.Len: the average length of samples. %Sarcasm: the proportion of sarcastic samples.

Datasets	#Train	#Test	#Avg.Len	%Sarcasm
SemEval18	3,398	780	17.4	49%
iSarcasm	3,116	887	27.3	18%
Ghosh	33,373	4,121	12.7	45%
IAC-V2	5,216	1,043	68.3	50%

Table 4: The experimental results of all comparing methods in terms of Accuracy (Acc) and Macro-F1 (F1). The best results are represented in **bold**. The second-best results are underlined.

Datasets	SemEval18		iSarcasm		Ghosh		IAC-V2	
Metric	Acc	F1	Acc	F1	Acc	F1	Acc	F1
NBOW	66.2%	65.1%	75.4%	55.1%	76.1%	75.6%	68.1%	68.0%
Bi-LSTM	70.8%	69.2%	79.1%	57.9%	78.6%	78.2%	77.2%	77.1%
SIARN	68.2%	67.0%	78.1%	57.4%	79.1%	78.6%	74.2%	74.1%
MIARN	68.5%	67.8%	79.4%	57.3%	79.1%	78.6%	75.6%	75.4%
SAWS	69.9%	68.9%	76.8%	57.5%	78.8%	78.5%	76.2%	76.2%
ADGCN	71.7%	70.1%	79.2%	58.5%	79.7%	79.5%	78.0%	78.0%
DC-Net	70.8%	69.6%	78.8%	58.7%	80.2%	78.6%	78.0%	77.9%
SarDeCK	71.7%	70.2%	78.1%	59.6%	**83.4%**	**83.0%**	77.5%	77.5%
SD-APRR	**72.2%**	**70.7%**	**80.3%**	**61.2%**	82.6%	82.3%	**78.8%**	**78.8%**

Baselines. We select a number of recent baseline methods for comparison. They are briefly described below:

- **NBOW**: A traditional SD method that represents samples by the averages of word embeddings.
- **Bi-LSTM**: A SD method that sequentially encodes sarcastic texts with Bi-LSTM.
- **SIARN** and **MIARN** (Tay et al., 2018): Two RNN-based SD methods that capture the incongruity by using the single-dimensional and multi-dimensional intra-sentence attentions, respectively. We implement in-house codes.
- **SAWS**[5] (Pan et al., 2020): A CNN-based SD method that cuts each text sample into snippets and uses self-attention to re-weight them.
- **ADGCN**[6] (Lou et al., 2021): A GCN-based SD method that builds affective and dependency graphs with SenticNet to capture the incongruity in a long distance.
- **DC-Net**[7] (Liu et al., 2022): A BERT-based SD method that respectively encodes literal meanings and implied meanings by the external sentiment lexicon.
- **SarDeCK**[8] (Li et al., 2021b) A BERT-based SD method that uses the COMET to derive dynamic commonsense knowledge and fuses the knowledge to enrich the contexts with attention.

Implementation details. In the experiments, except the BERT-based methods, we apply the 300-dimensional GloVe embeddings[9] to represent the words initially. The dimension of the Bi-LSTM output is set to 300, and the layer number of the masked graph-based encoder is set to 3. For all neural network-based methods, the batch size is set to 32. We take Adam as the optimizer, and the learning rate is set to 0.001. The regularization coefficient λ is set to 0.01. Besides, we use the Xavier Uniform to initialize the parameters. For the BERT-based methods, the number of training epochs is set to 6, while for other methods, the epoch number is fixed to 100 with an early stopping mechanism (Lou et al., 2021). In terms of all datasets, the splitting of training and testing is shown in Table 3. We independently run all comparing methods 5 times and report the average results.

[5]https://github.com/marvel2120/SAWS
[6]https://github.com/HLT-HITSZ/ADGCN
[7]https://github.com/yiyi-ict/ dual-channel-for-sarcasm
[8]https://github.com/LeqsNaN/SarDeCK
[9]https://nlp.stanford.edu/projects/glove/

Table 5: The ablation results of SD-APRR in terms of Accuracy (Acc) and Macro-F1 (F1). The best results are represented in **bold**.

Datasets	SemEval18		iSarcasm		Ghosh		IAC-V2	
Metric	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SD-APRR	72.2%	**70.7%**	**80.3%**	**61.2%**	82.6%	82.3%	**78.8%**	**78.8%**
w/o Result	**72.7%** ↑	70.0% ↓	79.1% ↓	59.2% ↓	82.0% ↓	81.3% ↓	78.6% ↓	78.6% ↓
w/o Reaction	70.9% ↓	70.4% ↓	78.6% ↓	59.5% ↓	82.1% ↓	81.8% ↓	78.5% ↓	78.5% ↓
w/o Masking	71.5% ↓	70.6% ↓	79.8% ↓	59.8% ↓	**83.0%** ↑	**82.7%** ↑	78.0% ↓	78.0% ↓

Table 6: The ablation results of the BERT-based SD-APRR over Ghosh and IAC-V2 in terms of Accuracy (Acc) and Macro-F1 (F1). The best results are represented in **bold**.

Datasets	Ghosh		IAC-V2	
Metric	Acc	F1	Acc	F1
SD-APRR (BERT)	**82.2%**	**79.9%**	**80.1%**	**80.0%**
w/o Result	81.7% ↓	79.2% ↓	79.5% ↓	**80.0%**
w/o Reaction	81.8% ↓	78.9% ↓	79.2% ↓	79.4% ↓

Evaluation metrics. By convention, we employ Accuracy and Macro-F1 as the evaluation metrics in our experiments.

4.2 Results and Analysis

The main results of all comparing methods are reported in Table 4, and we draw the following observations: (1) First, it can be clearly seen that our SD-APRR can achieve the highest scores of both Accuracy and Macro-F1 in most settings, where it ranks the first over SemEval18, iSarcasm, and IAC-V2, and ranks the second over Ghosh. (2) Second, we observe that SD-APRR mostly outperforms the recent strong baseline SarDeCK, which also employs COMET to generate auxiliary commonsense relations. A major difference between SarDeCK and SD-APRR is that the former integrates training samples with their corresponding commonsense results of COMET at the embedding level, while the latter treats the augmentations of raw training texts and inferred commonsense results of COMET as the whole raw texts. So the improvements to SarDeCK indirectly indicate that the direct access view may be a better perspective for SD. (3) Third, compared with ADGCN that is also based on graph neural networks, our SD-APRR achieves significant improvements over all datasets. This indicates that leveraging contextually inferred results and reactions can be a more efficient way for SD than leveraging context-free affective lexicons in a static way. (4) Finally, SD-APRR, ADGCN, DC-Net, and SarDeCK consistently perform better than NBOW, Bi-LSTM, SIARN, MIARN, and SAWS, the methods without external resources. The results support the previous statement that understanding sarcasm heavily relies on human background information.

4.3 Ablation Study

We conduct ablation studies to examine the effectiveness of the augmentations of results, augmentations of human reactions, and the denoising module. The results are reported in Table 5. Overall, when removing the results (**w/o Result**) and the reactions (**w/o Reaction**), the performance of SD-APRR show a decline on all datasets. This indicates that the potential results enable the SD-APRR to have extra explainable contexts to understand the negativity inside the negative situations. Meanwhile, human reactions provide explicit emotional clues that can be related to the negative situations during graph learning. However, when removing the denoising module (**w/o Masking**), the performance of SD-APRR decreases across the IAC-V2 dataset. This is because samples in the Ghosh are short texts, and their syntactical information may not be accurately captured, leading the masked graph-based encoder skips nodes related to the sarcasm by mistake.

Additionally, we replace the masked graph-based encoder with BERT (Devlin et al., 2019), and further compare this BERT-based version of SD-APRR with its ablative versions (**w/o Result** and **w/o Reaction**). Due to the space limitation, we report the results on two datasets, *i.e.,* Ghosh with relatively more training samples and IAC-V2 with longer text lengths. The results are shown in Table 6. We can observe that the full version performs the best compared with the ablative versions. These results further indicate the augmentations of results and human reactions inferred by COMET can improve the classification performance even

Table 7: The visualization of mask weights of example training samples. The words in red are with much lower values of mask weights.

ID	Texts
1	People that start construction work before 7am on a Sunday need to just f**k off and die thanks, then may get fired from job and I feel angry.
2	Love to live in this very cool country where a ten thousand dollar medical bill is low, then may go to see a doctor and I feel happy.
3	the most exciting thing to look forward to this weekend. work work work, then may get a promotion and I feel happy.

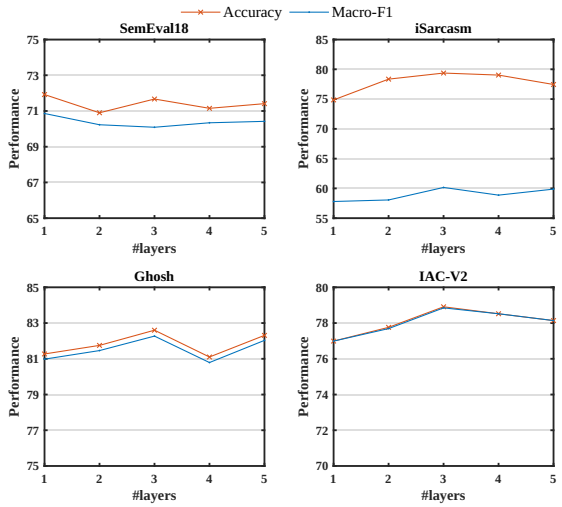

Figure 2: The impact of layer numbers of the masked graph-based encoder.

with a different encoder.

4.4 The Impact of Layer Numbers of the Masked Graph-based Encoder

We now investigate the impact of the layer number L of the masked graph-based encoder across benchmark datasets. We present the results with different values of $L \in \{1, 2, 3, 4, 5\}$ in Fig 2. We can observe that SD-APRR performs the best results across the SemEval18 dataset when $L = 1$, while achieving the best results across the other datasets when $L = 3$. The reason may be that the positive surfaces and the negative situations in the SemEval18 dataset are close to each other on the dependency graph, so the two terms can be associated together through low-order message-passing. While for the other three datasets, SD-APRR requires higher-order message passing to model the incongruity between the two terms. In practice, we suggest $L = 3$ as the default setting.

4.5 Visualization of Mask Weights.

To qualitatively visualize the impact of mask weights, we randomly select several examples and show the words with lower mask weights of the final layer of the masked graph-based encoder. The visualization is shown in Table 7. We use the red color to demonstrate the word tokens with lower mask weights. From the table, we observe that the encoder can effectively eliminate semantically irrelevant tokens, such as *"gets fired"* and *"see doctor"*, and wrong speakers' reactions, such as the term of *"happy"* in the second and the third cases. Besides, we observe that some sarcasm-irrelevant parts in the original texts can also be captured, *e.g.*, the stop words *"on"*, *"to"*, *"is"*.

5 Conclusion and Limitations

In this paper, we propose a novel SD method, entitled SD-APRR, which expresses negative situations of sarcasm by the potential results and human reactions of the associated events. We employ the COMET to estimate the results and human reactions, and form event-augmented samples with them. We employ those augmented samples as the whole sarcastic texts from the direct access view. We suggest a masked graph-based encoder, enabling to generate discriminative sample embeddings. Experimental results demonstrate that our SD-APRR can achieve competitive performance compared with the existing baseline methods.

We demonstrate two limitations: (1) The datasets used in this work are mostly collected from social media. In the future, we plan to collect sarcastic texts from various sources, such as the literature and films, and conduct more experiments with them. (2) Our exploration of sarcasm theories still has some space to improve. Though the incongruity theory is the mainstream in the community, there are other theories worthy to investigate in the future.

Acknowledgment

We would like to acknowledge support for this project from the National Natural Science Foun-

dation of China (No.62076046), and the Young Scientists Fund of the National Natural Science Foundation of China (No.62006034).

References

Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. 2016. Internet argument corpus 2.0: An SQL schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4445–4452.

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.

Santosh Kumar Bharti, Korra Sathya Babu, and Sanjay Kumar Jena. 2015. Parsing-based sarcasm sentiment recognition in twitter data. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, page 1373–1380.

Erik Cambria, Yang Li, Frank Z. Xing, Soujanya Poria, and Kenneth Kwok. 2020. Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, page 105–114.

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, NAACL-HLT 2019*, pages 4171–4186.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169.

Raymond W Gibbs. 1986. On the psycholinguistics of sarcasm. *Journal of experimental psychology: General*, 115(1):3.

Rachel Giora and Ofer Fein. 1999. Irony: Context and salience. *Metaphor and Symbol*, 14(4):241–257.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Pedram Hosseini, David A. Broniatowski, and Mona Diab. 2022. Knowledge-augmented language models for cause-effect relation classification. In *Proceedings of the First Workshop on Commonsense Representation and Reasoning (CSRR 2022)*, pages 43–48.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 6384–6392.

Stacey L Ivanko and Penny M Pexman. 2003. Context incongruity and irony processing. *Discourse processes*, 35(3):241–279.

Aditya Joshi, Pushpak Bhattacharyya, and Mark James Carman. 2017. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5):73:1–73:22.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, pages 7871–7880.

Jiangnan Li, Zheng Lin, Peng Fu, and Weiping Wang. 2021a. Past, present, and future: Conversational emotion recognition through structural modeling of psychological knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1204–1214.

Jiangnan Li, Hongliang Pan, Zheng Lin, Peng Fu, and Weiping Wang. 2021b. Sarcasm detection with commonsense knowledge. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3192–3201.

Bin Liang, Hang Su, Lin Gui, Erik Cambria, and Ruifeng Xu. 2022. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-Based Systems*, 235:107643.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839.

Yiyi Liu, Yequan Wang, Aixin Sun, Xuying Meng, Jing Li, and Jiafeng Guo. 2022. A dual-channel framework for sarcasm recognition by detecting sentiment conflict. In *Findings of the Association for Computational Linguistics: NAACL*, pages 1670–1680.

Chenwei Lou, Bin Liang, Lin Gui, Yulan He, Yixue Dang, and Ruifeng Xu. 2021. Affective dependency graph for sarcasm detection. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1844–1849.

Diana Maynard and Mark Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings*

of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 4238–4243.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160.

Silviu Oprea and Walid Magdy. 2020. iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289.

Hongliang Pan, Zheng Lin, Peng Fu, and Weiping Wang. 2020. Modeling the incongruity between sentence snippets for sarcasm detection. In *ECAI 2020*, pages 2132–2139.

Bethany Pickering, Dominic Thompson, and Ruth Filik. 2018. Examining the emotional impact of sarcasm using a virtual environment. *Metaphor and Symbol*, 33(3):185–197.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on Czech and English Twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, page 97–106.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12. Applications of Natural Language to Information Systems.

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.

Sahand Sabour, Chujie Zheng, and Minlie Huang. 2022. CEM: commonsense-aware empathetic response generation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11229–11237.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3027–3035.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451.

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.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 39–50.

Raymond W.Gibbs. 2002. A new look at literal meaning in understanding what is said and implicated. *Journal of pragmatics*, 34(4):457–486.

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.

Mingqi Yang, Yanming Shen, Heng Qi, and Baocai Yin. 2021. Soft-mask: Adaptive substructure extractions for graph neural networks. In *Proceedings of the Web Conference 2021*, page 2058–2068.

Da Yin, Li Dong, Hao Cheng, Xiaodong Liu, Kai-Wei Chang, Furu Wei, and Jianfeng Gao. 2022. A survey of knowledge-intensive nlp with pre-trained language models. *arXiv preprint arXiv:2202.08772*.

Hongming Zhang, Daniel Khashabi, Yangqiu Song, and Dan Roth. 2020. Transomcs: From linguistic graphs to commonsense knowledge. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4004–4010.

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.