Affective Dependency Graph for Sarcasm Detection

Chenwei Lou*
Joint lab of CMS-HITSZ, Harbin Institute of Technology, Shenzhen
louchenw@163.com

Bin Liang*
Joint lab of CMS-HITSZ, Harbin Institute of Technology, Shenzhen
bin.liang@stu.hit.edu.cn

Lin Gui
University of Warwick
lin.gui@warwick.ac.uk

Yulan He
University of Warwick
Yulan.He@warwick.ac.uk

Yixue Dang
China Merchants Securities Co.,Ltd.
dangyixue@cmschina.com.cn

Ruifeng Xu†
Harbin Institute of Technology, Shenzhen, Peng Cheng Lab
xuruifeng@hit.edu.cn

ABSTRACT

Detecting sarcastic expressions could promote the understanding of natural language in social media. In this paper, we revisit sarcasm detection from a novel perspective, so as to account for the long-range literal sentiment inconsistencies. More concretely, we explore a novel scenario of constructing an affective graph and a dependency graph for each sentence based on the affective information retrieved from external affective commonsense knowledge and the syntactical information of the sentence. Based on it, an Affective Dependency Graph Convolutional Network (ADGCN) framework is proposed to draw long-range incongruity patterns and inconsistent expressions over the context for sarcasm detection by means with interactively modeling the affective and dependency information. Experimental results on multiple benchmark datasets show that our proposed approach outperforms the current state-of-the-art methods in sarcasm detection.

CCS CONCEPTS

• **Information systems → Sentiment analysis**;

KEYWORDS

sarcasm detection, graph network, sentiment analysis

ACM Reference Format:
Chenwei Lou, Bin Liang, Lin Gui, Yulan He, Yixue Dang, and Ruifeng Xu. 2021. Affective Dependency Graph for Sarcasm Detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3404835.3463061

1 INTRODUCTION

Sarcasm is a common speech act in human communications, which has received much research attention [8, 9, 13, 16–18, 23]. As shown

Sarcasm: I *love* when people *ignore* me. | **Non**: *Love* you more than this.

Figure 1: Examples of Sarcasm and Non-sarcasm expression.

in Figure 1, there are two instances paired with their labels (*Sarcasm* or *Non-sarcasm*). Note that both of them contain a decisive sentiment word "*love*". While in the sarcastic example, the word discrepant "*ignore*" leads to a contradiction expression. That is, there are some incongruity expressions in sarcastic context [13].

Some early studies attempt to extract the incongruity expressions in sarcasm detection by searching a set of positive verbs and negative situations [2, 10, 26] or employing lexical features [22]. Most recent methods employ deep neural networks to capture the subtle semantic incongruity patterns [7, 32, 35]. Further, Babanejad et al. [1] leverages both affective and contextual features to extend the architecture of BERT for sarcastic expressions learning. Most existing studies, however, are largely inadequate to determine the affective dependencies in sarcastic expressions when the incongruity patterns are separated far away in the context, or easy to mistake the inessential contextual words as sarcastic descriptors. As the sarcasm example shown in Figure 1, the word "*love*" is not near to "*ignore*" in the incongruity expression.

In this work, inspired by some existing graph-based models proposed in other tasks [4, 6, 11, 20, 30, 31, 36–38], we explore a novel scenario of constructing an affective graph and a dependency graph for each instance based on the affective clues retrieved from external affective knowledge (SenticNet [3]) and the dependency tree of the sentence, so as to leverage the contextual affective dependencies of incongruity expressions in sarcasm detection. Based on it, an Affective Dependency Graph Convolutional Network (ADGCN) structure is employed to provide the long-range multi-word affective dependencies for understanding the roles of context words in the learning of incongruity expressions. The main contributions of our work can be summarized as follows:

- We are the first to exploit GCN model for drawing incongruity patterns over the context in sarcasm detection.
- A novel scenario of affective and dependency graphs construction is explored to extract the contradictory implications and incongruity expressions in sarcasm detection.
- Experimental results on a number of benchmark datasets demonstrate that our proposed method achieves the state-of-the-art performance in sarcasm detection.

*The first two authors contribute equally to this work.
†Corresponding Author.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463061

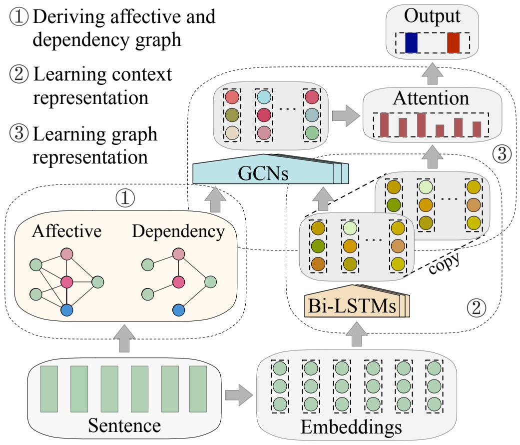


Figure 2: Architecture of the proposed ADGCN framework.

2 METHODOLOGY

In this section, we describe our proposed Affective Dependency Graph Convolutional Network (ADGCN) framework in details. As demonstrated in Figure 2, the architecture of the proposed ADGCN framework contains three main components: 1) *Deriving affective and dependency graphs*, which constructs an affective graph and a syntax-aware dependency graph for each sentence based on affective commonsense knowledge and dependency tree; 2) *Learning context representation*, which learns the vector representations of the context with bidirectional LSTMs (Bi-LSTM); 3) *Learning graph representation*, which leverages the affective dependencies of the context with multi-layer GCNs for sarcasm detection.

2.1 Deriving Affective and Dependency Graphs

To leverage the affective dependencies of the context, we explore a novel scenario of constructing an affective graph and a dependency graph for each sentence. This aims to discern the affective expressions of the contextual words and preserve global structure information of the sentence in sarcasm detection simultaneously.

Given a sentence s consists of n words $s = \{w_i\}_{i=1}^{n}$, to explore the affective expressions of the context for determining the role of contextual incongruity information in learning sarcastic expressions, we construct an affective guided graph and attain an adjacency matrix $\boldsymbol{A}^a \in \mathbb{R}^{n\times n}$, based on the affective scores of words retrieved from an external affective commonsense knowledge:

$$A^a_{i,j} = |\mathcal{S}(w_i) - \mathcal{S}(w_j)| \tag{1}$$

where $\mathcal{S}(w_i) \in [-1, 1]$ represents the affective score of word w_i retrieved from SenticNet [3]. We set $\mathcal{S}(w_t) = 0$ if w_t is not contained in the knowledge. $|\cdot|$ represents absolute value calculation. In this way, words with opposite emotions could be highly regarded. Thus the affective incongruity expressions could be propagated to discriminate the contradiction between literal expression and the authentic intention of the author in sarcasm detection.

In addition, intuitively, affective expressions generally depend on some syntactic structure, as the sarcastic clue of "*people ignore me*" shown in Figure 1. To this end, inspired by previous syntax-aware graph methods [12, 19, 27, 34], in addition to the affective graph we construct a dependency graph based on the dependency tree of the sentence[1]:

$$A^d_{i,j} = 1 \qquad \text{if } \mathcal{T}(w_i, w_j) \tag{2}$$

where $\boldsymbol{A}^d \in \mathbb{R}^{n\times n}$, whose remaining elements are 0. $\mathcal{T}(w_i, w_j)$ represents that there is a relation between w_i and w_j in the dependency tree of the sentence. Inspired by [15], we construct the undirected graph to enrich the affective and dependency information: $A_{i,j} = A_{j,i}$, and also set a self-loop for each word: $A_{i,i} = 1$.

2.2 Learning Context Representation

We embed each word of $s = \{w_i\}_{i=1}^{n}$ into an m-dimensional embedding $\boldsymbol{x}_i \in \mathbb{R}^m$ via mapping the embedding from the lookup table $\boldsymbol{X} \in \mathbb{R}^{m\times|V|}$, $|V|$ is the vocabulary size. Then we feed the embedding matrix $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]$ into bidirectional LSTMs to encode the input sentence into vector representations:

$$\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n\} = \textit{Bi-LSTM}(\boldsymbol{x}) \tag{3}$$

Where $\boldsymbol{h}_t \in \mathbb{R}^{2d_h}$ denotes the hidden representation of $\boldsymbol{x}_t$ in time step t, d_h denotes the dimensionality of hidden representation.

2.3 Learning Graph Representation

Different from conventional sarcasm detection methods that treated a sentence as a word sequence and purely extracted sarcastic information from the literal or semantic content. We explore a novel Affective Dependency Graph Convolutional Network (ADGCN) framework that interactively feeding the affective and dependency graphs of the sentence into the multi-layers GCN architecture to leverage the long-range affective incongruity expressions. Each node in the l-th GCN layer is updated according to the hidden representations of its neighborhoods according to the adjacency matrices of the two graphs, the process is defined as:

$$\boldsymbol{g}^l = \text{ReLU}(\tilde{\boldsymbol{A}}^d \text{ReLU}(\tilde{\boldsymbol{A}}^a \boldsymbol{g}^{l-1} \boldsymbol{W}^l_a + \boldsymbol{b}^l_a)\boldsymbol{W}^l_d + \boldsymbol{b}^l_d) \tag{4}$$

where $\boldsymbol{g}^{l-1} \in \mathbb{R}^{n\times 2d_h}$ is the hidden graph representation evolved from the preceding GCN layer, and the original input nodes of the first GCN layer are the context representation learned by Bi-LSTMs: $\boldsymbol{g}^0 = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n]$. $\tilde{\boldsymbol{A}}$ is the normalized adjacency matrix: $\tilde{\boldsymbol{A}}_i = \boldsymbol{A}_i/(E_i + 1)$. $E_i = \sum_{j=1}^{n} A_{i,j}$ is the degree of $\boldsymbol{A}_i$. $\boldsymbol{W}^l \in \mathbb{R}^{2d_h\times 2d_h}$, $\boldsymbol{b}^l \in \mathbb{R}^{2d_h}$ are the trainable parameters of the l-th GCN layer.

Then inspired by [34], we employ a retrieval-based attention mechanism to capture the affective dependency graph-oriented features from context representations:

$$\boldsymbol{r} = \sum_{t=1}^{n} \alpha_t \boldsymbol{h}_t, \quad \alpha_t = \frac{\exp(\beta_t)}{\sum_{i=1}^{n}\exp(\beta_i)}, \quad \beta_t = \sum_{i=1}^{n} \boldsymbol{h}_t^\top \boldsymbol{g}_i^L \tag{5}$$

where $\top$ represents matrix transposition, $\boldsymbol{g}^L$ is the output of the final GCN layer. Afterward, the final sarcastic representation is fed into a fully-connected layer with softmax normalization to capture a probability distribution $\hat{\boldsymbol{y}}$ of sarcasm decision space:

$$\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{W}_o \boldsymbol{r} + \boldsymbol{b}_o) \tag{6}$$

where $\hat{\boldsymbol{y}} \in \mathbb{R}^{d_p}$ is the predicted sarcastic probability for the input sentence, d_p is the dimensionality of sarcasm labels. $\boldsymbol{W}_o \in \mathbb{R}^{d_p\times 2d_h}$ and $\boldsymbol{b}_o \in \mathbb{R}^{d_p}$ are trainable parameters.

[1] We employ spaCy toolkit to derive dependency tree of the sentence: https://spacy.io/.

Table 1: Statistics of the experimental data.

Dataset	Train		Test	
	Sarcasm	*Non*	*Sarcasm*	*Non*
IAC-V1	862	859	97	94
IAC-V2	2947	2921	313	339
Tweets-1 (Riloff)	282	1051	35	113
Tweets-2 (Ptáček)	23456	24387	2569	2634
Reddit-1 (/r/movies)	5521	5607	1389	1393
Reddit-2 (/r/technology)	6419	6393	1596	1607

2.4 Learning Objective

We minimize the cross-entropy loss via the standard gradient descent algorithm to train the model:

$$\min_{\Theta} \mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{d_p} y_i^j log \hat{y}_i^j + \lambda ||\Theta||^2 \tag{7}$$

where N is the training data size. $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$ respectively represent the ground-truth and estimated label distribution of instance i. Θ denotes all trainable parameters of the model, λ represents the coefficient of L_2-regularization.

3 EXPERIMENTS

3.1 Experimental Data and Settings

To evaluate our proposed model, following [28], we conduct experiments on 6 benchmark datasets from 3 well-known sources:

- **IAC (Internet Argument Corpus)**: We use two versions of the dataset from [21], which are denoted as IAC-V1[2] and IAC-V2[3] respectively.
- **Tweets**: We use two datasets collected by Riloff et al. [26] and Ptáček et al. [25]. For both datasets, we retrieve tweets using the Twitter API with the provided tweet IDs[4].
- **Reddit**: We use two subsets (i.e. */r/movies* and */r/technology*) of Reddit dataset provided by [14] for sarcasm detection.

The statistics of the experimental data are reported in Table 1.

In our experiments, for non-BERT models, we utilize GloVe [24] to embed each word as a 300-dimensional embedding. The number of GCN layers is set to 3. The dimensionality of hidden representations is set to 300. The coefficient λ of L_2 regularization is set to 0.01. Adam is utilized as the optimizer with a learning rate of 0.001 to train the model, and the mini-batch size is 128 for Tweets-2 and 32 for other datasets. For BERT-based models, we use the pre-trained uncased BERT-base [5] with 768-dimensional embedding, and the learning rate is 0.00002. We perform Accuracy (Acc.) and Macro *F1-score* (F1) to measure the performance of the models[5].

3.2 Comparison Models

We compare our model, i.e. **ADGCN** and **ADGCN-BERT** (replace Bi-LSTM with BERT), with the following 13 baselines[6]. Including 1) statistic technique: **NBOW** [28]; 2) conventional neural networks: **CNN**, **LSTM**, **ATT-LSTM** [33]; 3) sarcasm detection methods: **GRNN** [35], **CNN-LSTM-DNN** [7], **SIARN** [28], **MI-ARN** [28], **SMSD** [32], **SMSD-BiLSTM** [32]; 4) BERT-based models: **BERT** [5], **ACE2-BERT-EMoSi** [1], **ACE2-BERT-EAISe** [1].

[2]https://nlds.soe.ucsc.edu/sarcasm1
[3]https://nlds.soe.ucsc.edu/sarcasm2
[4]http://api.twitter.com/
[5]The source code of this work is released at https://github.com/HLT-HITSZ/ADGCN
[6]Since there are no unified datasets among existing studies, we conduct comparison experiments of baselines on our datasets with open source code or reproduced code.

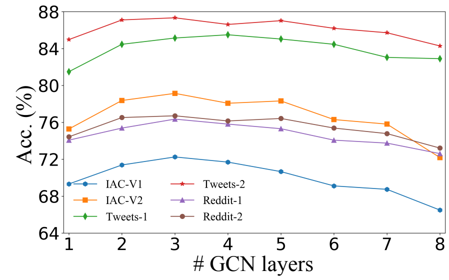


Figure 3: Impact of the number of GCN layers.

3.3 Main Experimental Results

Table 2 shows the experimental results on 6 benchmark datasets. We can observe that our proposed **ADGCN** consistently outperforms all compared baselines over both non-BERT and BERT-based models on all datasets. To be specific, the best improved results of Acc. and F1 respectively are 7.65% and 7.78% compared with the previous state-of-the-art performance. For BERT-based methods, the best improved results of Acc. and F1 respectively are 7.37% and 7.35% compared with previous state-of-the-art performance. This verifies that our proposed model, which leveraging affective dependencies of the context with a GCN architecture outstandingly improves the performance of sarcasm detection.

3.4 Ablation Study

To analyze the impact of different components of the proposed **ADGCN** bring to the performance, we conduct an ablation study and report the results in Table 3. Note that removal of affective graph sharply degrades the performance, which indicates that affective information is significant in the sarcastic expressions learning. Additionally, the graph without syntax-aware refinement also leads to a considerably poorer performance. This implies that refining the affective graph with syntax-aware information advances the model to extract the linchpin clues of incongruity expressions by affective dependencies.

3.5 Impact of GCN Layers

To investigate the impact of the number of GCN layers on the performance of our proposed **ADGCN**, we vary the number from 1 to 8 and report the results in Figure 3. Note that 3-layer GCN performs overall better than other layers, and thus we set the number of GCN layers as 3. One GCN layer performs unsatisfactorily on all datasets, which indicates inadequate network structure is insufficient to exploit decent sarcastic features. Additionally, when the layer greater than 3, the performance fluctuates and tends to decline with the increasing number. This implies that roughly increasing the number of GCN layers is vulnerable to slash the learning ability of the model due to the sharp increase of model parameters.

Table 2: Main experimental results on different datasets. Average scores over 10 runs are reported. Best scores are in bold.

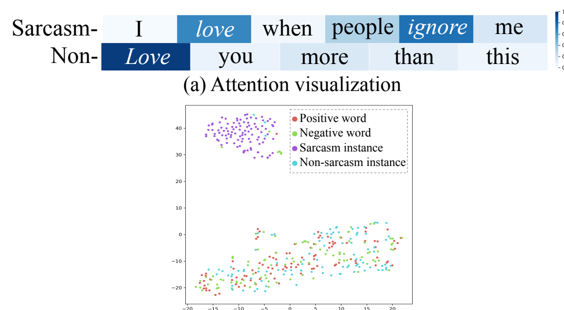
Model	IAC-V1		IAC-V2		Tweets-1		Tweets-2		Reddit-1		Reddit-2	
	Acc. (%)	*F1 (%)*	*Acc. (%)*	*F1 (%)*	*Acc. (%)*	*F1 (%)*	*Acc. (%)*	*F1 (%)*	*Acc. (%)*	*F1 (%)*	*Acc. (%)*	*F1 (%)*
NBOW [28]	59.63	59.38	67.55	67.74	74.86	63.39	74.23	74.13	69.66	69.66	65.61	65.56
CNN	60.62	60.27	70.15	69.95	78.64	66.47	80.77	80.76	70.21	70.16	68.00	67.91
LSTM	60.52	60.45	71.10	70.84	79.33	67.62	80.79	80.78	70.66	70.59	68.80	68.16
ATT-LSTM [33]	63.45	63.18	65.46	65.33	80.70	69.23	81.56	81.56	70.50	70.44	68.62	68.55
GRNN [35]	63.87	62.44	72.23	70.92	79.10	68.35	81.18	80.14	71.55	70.47	67.15	67.14
CNN-LSTM-DNN [7]	66.49	66.46	76.99	67.93	76.49	67.80	79.74	79.20	71.17	71.14	67.62	67.34
SIARN [28]	64.24	63.79	74.98	74.95	79.12	67.47	83.59	83.59	70.66	70.58	68.55	68.51
MIARN [28]	64.45	63.89	75.84	75.80	79.19	67.11	83.78	83.78	70.72	70.68	68.48	68.44
SMSD [32]	65.13	65.07	72.19	72.13	78.11	67.18	81.25	81.24	69.58	69.55	68.94	68.90
SMSD-BiLSTM [32]	64.50	64.40	71.44	71.36	78.92	67.75	78.92	78.90	69.84	69.75	69.06	69.00
ADGCN (**ours**)	**72.25**	**72.20**	**79.14**	**79.13**	**85.14**	**77.01**	**87.33**	**87.33**	**76.35**	**76.31**	**76.71**	**76.69**
BERT [5]	68.95	68.88	78.41	78.40	83.38	76.08	86.37	86.36	76.89	76.87	77.42	77.41
ACE2-BERT-EMoSi [1]	66.49	66.48	76.75	76.65	81.76	72.12	86.58	86.58	74.64	74.62	76.30	76.35
ACE2-BERT-EAISe [1]	68.06	67.98	77.25	77.10	81.76	73.39	86.60	86.60	74.73	74.70	76.37	76.36
ADGCN-BERT (**ours**)	**76.32**	**76.23**	**82.37**	**82.36**	**88.16**	**81.91**	**90.31**	**89.54**	**80.68**	**80.63**	**80.77**	**80.77**

Table 3: Accuracy results of ablation study. $\mathcal{A}$ denotes affective graph, $\mathcal{S}$ denotes syntax-aware refinement.

Model	IAC-V1	IAC-V2	Tweets-1	Tweets-2	Reddit-1	Reddit-2
ADGCN	**72.25**	**79.14**	**85.14**	**87.33**	**76.35**	**76.71**
w/o $\mathcal{A}$	69.03	75.52	81.67	82.28	73.15	71.89
w/o $\mathcal{S}$	71.13	77.20	83.16	85.31	75.76	73.78

3.6 Visualization

To qualitatively demonstrate how affective dependency graph improves the performance of sarcasm detection, we present a visualization analysis in Figure 4. We first visualize the attention scores of typical sarcasm/non-sarcasm examples learned by our proposed ADGCN in Figure 4 (a) to analyze how the proposed ADGCN draws the affective dependencies in sarcastic/non-sarcastic expressions learning by interactively modeling both affective and dependency information of the context. Note that due to the proposed ADGCN, the affective auxiliary syntactic dependency information enhances the incongruous words from sarcasm sentences by attention signals. Hence, the weighted sum representation of sarcasm instances would neither be similar to positive words nor negative words. On the contrary, for the non-sarcasm instances, the representations will be similar to the affective words since the ADGCN only focuses on few congruous words. Thus the representations of non-sarcasm instances should be mixed with affective words but separated with sarcasm instances. To further investigate the difference of sarcastic/non-sarcastic representations, in Figure 4 (b), we show the t-SNE [29] visualization of intermediate sarcasm and non-sarcasm representations, which adhere to the hidden representations of affective words derived by Bi-LSTM layers. Note that a significantly clear separation between sarcasm representations and affective word vectors is represented, while the distribution of non-sarcasm representations is quite overlapping with affective words. This further indicates that our proposed ADGCN effectively represents non-sarcasm instances by attaching them to affective words, and derive the sarcasm representations according to the contradictory affective dependencies and incongruity expressions.



(a) Attention visualization

(b) Vectors visualization

Figure 4: Results of visualization.

4 CONCLUSION

In this paper, we propose a novel scenario of constructing an affective graph and a dependency graph for each sentence to learn the long-range contradictory implications and incongruity expressions in sarcasm detection. More concretely, an affective dependency graph convolutional network (ADGCN) framework is exploited to draw incongruity patterns and inconsistent sentiment expressions over the context in the learning of sarcastic features by interactively modeling both affective and syntactical information of the context. Experimental results on multiple benchmark datasets show that our proposed model significantly outperforms state-of-the-art baseline methods in sarcasm detection.

ACKNOWLEDGMENTS

This work was partially supported by National Natural Science Foundation of China (61632011, 61876053), Guangdong Province Covid-19 Pandemic Control Research Funding (2020KZDZX1224), Shenzhen Foundational Research Funding (JCYJ20180507183527919, JCYJ20180507183608379), Joint Lab of Lab of China Merchants Securities and HITSZ, and UK EPSRC (grant no. EP/T017112/1, EP/V048597/1). Yulan He is supported by a Turing AI Fellowship funded by the UK Research and Innovation (UKRI) (grant no. EP/V020579/1).

REFERENCES

[1] Nastaran Babanejad, Heidar Davoudi, Aijun An, and Manos Papagelis. 2020. Affective and Contextual Embedding for Sarcasm Detection. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 225–243. https://doi.org/10.18653/v1/2020.coling-main.20

[2] David Bamman and Noah Smith. 2015. Contextualized Sarcasm Detection on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media* 9, 1. https://ojs.aaai.org/index.php/ICWSM/article/view/14655

[3] 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 (CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 105–114. https://doi.org/10.1145/3340531.3412003

[4] Ying Chen, Wenjun Hou, Shoushan Li, Caicong Wu, and Xiaoqing Zhang. 2020. End-to-End Emotion-Cause Pair Extraction with Graph Convolutional Network. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 198–207. https://doi.org/10.18653/v1/2020.coling-main.17

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[6] Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A Neural Multi-digraph Model for Chinese NER with Gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1462–1467. https://doi.org/10.18653/v1/P19-1141

[7] 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*. Association for Computational Linguistics, San Diego, California, 161–169. https://doi.org/10.18653/v1/W16-0425

[8] Raymond W. Gibbs. 1986. On the Psycholinguistics of Sarcasm. *Journal of Experimental Psychology: General* 115, 1 (1986), 3–15. https://doi.org/10.1037/0096-3445.115.1.3

[9] Raymond W Gibbs. 2007. On the psycholinguistics of sarcasm. *Irony in language and thougt: A cognitive science reader* (2007), 173–200.

[10] Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying Sarcasm in Twitter: A Closer Look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 581–586. https://www.aclweb.org/anthology/P11-2102

[11] Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 241–251. https://doi.org/10.18653/v1/P19-1024

[12] Binxuan Huang and Kathleen Carley. 2019. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. 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)*. Association for Computational Linguistics, Hong Kong, China, 5469–5477. https://doi.org/10.18653/v1/D19-1549

[13] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing Context Incongruity for Sarcasm Detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, 757–762. https://doi.org/10.3115/v1/P15-2124

[14] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A Large Self-Annotated Corpus for Sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

[15] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 28th International Conference on Computational Linguistics*.

[16] Roger Kreuz and Gina Caucci. 2007. Lexical Influences on the Perception of Sarcasm. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*. Association for Computational Linguistics, Rochester, New York, 1–4. https://www.aclweb.org/anthology/W07-0101

[17] Roger J Kreuz and Sam Glucksberg. 1989. How to be sarcastic: The echoic reminder theory of verbal irony. *Journal of experimental psychology: General* 118, 4 (1989), 374.

[18] Amit Kumar Jena, Aman Sinha, and Rohit Agarwal. 2020. C-Net: Contextual Network for Sarcasm Detection. In *Proceedings of the Second Workshop on Figurative Language Processing*. Association for Computational Linguistics, Online, 61–66. https://doi.org/10.18653/v1/2020.figlang-1.8

[19] Bin Liang, Rongdi Yin, Lin Gui, Jiachen Du, and Ruifeng Xu. 2020. Jointly Learning Aspect-Focused and Inter-Aspect Relations with Graph Convolutional Networks for Aspect Sentiment Analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 150–161. https://doi.org/10.18653/v1/2020.coling-main.13

[20] Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 505–514. https://doi.org/10.18653/v1/2020.acl-main.48

[21] Stephanie Lukin and Marilyn Walker. 2013. Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media*. Association for Computational Linguistics, Atlanta, Georgia, 30–40. https://www.aclweb.org/anthology/W13-1104

[22] Edwin Lunando and Ayu Purwarianti. 2013. Indonesian social media sentiment analysis with sarcasm detection. In *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 195–198. https://doi.org/10.1109/ICACSIS.2013.6761575

[23] Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* 2, 1–2 (Jan. 2008), 1–135. https://doi.org/10.1561/1500000011

[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[25] 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*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 213–223. https://www.aclweb.org/anthology/C14-1022

[26] 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*. Association for Computational Linguistics, Seattle, Washington, USA, 704–714. https://www.aclweb.org/anthology/D13-1066

[27] Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6578–6588. https://doi.org/10.18653/v1/2020.acl-main.588

[28] 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)*. Association for Computational Linguistics, Melbourne, Australia, 1010–1020. https://doi.org/10.18653/v1/P18-1093

[29] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[30] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 3229–3238. https://doi.org/10.18653/v1/2020.acl-main.295

[31] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2021), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

[32] 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 (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 2115–2124. https://doi.org/10.1145/3308558.3313735

[33] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 1480–1489. https://doi.org/10.18653/v1/N16-1174

[34] Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. 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)*. Association for Computational Linguistics, Hong Kong, China, 4568–4578. https://doi.org/10.18653/v1/D19-1464

[35] Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet Sarcasm Detection Using Deep Neural Network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, 2449–2460. https://www.aclweb.org/

anthology/C16-1231

[36] Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 334–339. https://doi.org/10.18653/v1/2020.acl-main.31

[37] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

[38] Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun. 2019. Graph Neural Networks with Generated Parameters for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1331–1339. https://doi.org/10.18653/v1/P19-1128