applied sciences

MDPI

Article

Affection Enhanced Relational Graph Attention Network for Sarcasm Detection

Guowei Li , Fuqiang Lin, Wangqun Chen and Bo Liu *

School of Computer, National University of Defense Technoloy, Changsha 410073, China; liguowei@nudt.edu.cn (G.L.); linfuqiang13@nudt.edu.cn (F.L.); chenwangqun19@nudt.edu.cn (W.C.)
* Correspondence: kyle.liu@nudt.edu.cn

Abstract: Sarcasm detection remains a challenge for numerous Natural Language Processing (NLP) tasks, such as sentiment classification or stance prediction. Existing sarcasm detection studies attempt to capture the subtle semantic incongruity patterns by using contextual information and graph information through Graph Convolutional Networks (GCN). However, direct application of dependence may inevitably introduce noisy information and inferiorly in modeling long-distance or disconnected words in the dependency tree. To better learn the sentiment inconsistencies between terms, we propose an Affection Enhanced Relational Graph Attention network (ARGAT) by jointly considering the affective information and the dependency information. Specifically, we use Relational Graph Attention Networks (RGAT) to integrate relation information guided by a trainable matrix of relation types and synchronously use GCNs to integrate affection information explicitly donated by affective adjacency matrixes. The employment of RGAT contributes to information interaction of structural relevant word pairs with a long distance. With the enhancement of affective information, the proposed model can capture complex forms of sarcastic expressions. Experimental results on six benchmark datasets show that our proposed approach outperforms state-of-the-art sarcasm detection methods. The best-improved results of accuracy and F1 are 4.19% and 4.33%, respectively.

Keywords: sarcasm detection; affection; dependency learning

check for updates

Citation: Li, G.; Lin, F.; Chen, W.; Liu, B. Affection Enhanced Relational Graph Attention Network for Sarcasm Detection. *Appl. Sci.* **2022**, *12*, 3639. https://doi.org/10.3390/app12073639

Academic Editors: Valentino Santucci and Luis Javier García Villalba

Received: 2 March 2022
Accepted: 31 March 2022
Published: 4 April 2022

1. Introduction

Sarcasm is a sophisticated linguistic expression and has received a lot of research attention [1–3]. By irony, the meaning expressed is different from its real one, and such kind of incongruity presents in either an explicit or implicit way. Considering the former example in Figure 1, we can observe an apparent contradiction in the sentence "I absolutely love to be ignored" with strong contrastive emotional words "love" and "ignore", so it is imperative to mine this kind of incongruity expression in sarcastic context.

Early works attempted to extract the incongruity expressions in sarcasm detection by capturing the incongruity in-between the words [2–4] or using lexical features [5]. Recently, some deep learning-based and graph-based methods have achieved significant improvements in sarcasm detection by capturing the incongruity [6–10]. Tay et al. [6] used a self-attention-based neural network to model the incongruity on word-level explicitly. After that, Pan et al. [7] proposed to model the incongruity between sentence snippets as they contain more semantic information. Babanejad et al. [8] first attempted to alter BERT's architecture directly and trained it from scratch to build a sarcasm classifier. Later, Liang et al. [9] explored interactive GCNs to interactively learn the incongruity relations of in-modal and cross-modal graphs for determining the significant clues in sarcasm detection. Lou et al. [10] utilized affective and dependency graphs to extract the contradictory implications and incongruity expressions in sarcasm detection, which achieved the best performance at the time.

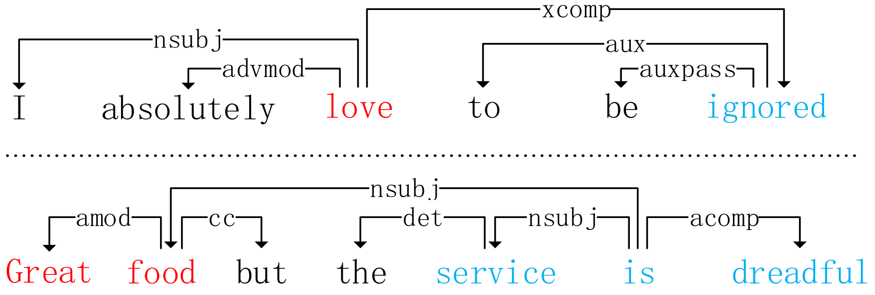
xcomp
nsubj
advmod
aux
auxpass
I absolutely love to be ignored
nsubj
amod
cc
det
nsubj
acomp
Great food but the service is dreadful

Figure 1. The figure shows two examples of tweets with dependency. The above sentence is sarcastic tweet and the below is not sarcastic. The words or snippets with incongruity are colored.

These deep learning models can capture the semantic information well but inevitably lead to misjudgment. Consider the below not sarcastic example in Figure 1, and we observe there still exists a strong contrast between "great" and "dreadful" in the sentence "Great food, but the service is dreadful", which is similar to the former sarcastic example. We will wrongly conclude it as a sarcastic expression if we only consider the semantic information. Once we consider the dependency relationship derived from the parsing tree, we will find that "great" is an adjectival modifier of "food", and "dreadful" is the adjectival complement of "service", so the dependency information can help us differentiate the falsely classified sentence. However, direct use of dependencies allocates the same weight to adjacent nodes in the parsing tree, so we apply the dependency relationships as trainable parameters to refine the importance of different relations.

Inspired by Ref. [10], which utilizes both affective graph and dependency graph to extract the contradictory expressions, in this paper, we explore a novel network to leverage both affection information and relation information based on contextual representations. The proposed model (ARGAT) first extracts contextual information based on BiLSTM [11] and sends the contextual embeddings to RGAT [12] and GCN [13] layers. The RGAT and GCN layer can be stacked for n layers to extract deeper features. Then, we concatenate the two outputs and put them into a traditional classifier for sarcasm prediction. Experimental results on six benchmark datasets demonstrate that our proposed method achieves the state-of-the-art performance in sarcasm detection. The main contributions of our work can be summarized as follows:

- We exploit the RGAT network to better learn the syntactic information by incorporating dependency relation information, which contributes to information interaction of structural relevant word pairs with long distances.
- A combination model of affective and relational graphs is explored to extract the incongruity in sarcasm detection.
- Experimental results on a number of benchmark datasets demonstrate that our proposed method achieves the state-of-the-art performances in sarcasm detection.

The rest of this paper is organized as follows: in Section 2, we briefly review extant related studies; in Section 3, we describe our proposed model architecture in detail. Specifically, we illustrate how BiLSTM, RGAT layer, and GCN layer works; in Section 4, we demonstrate our experimental settings and discuss experiment results; we conclude this paper in Section 5.

2. Related Works

Sarcasm is a complex linguistic phenomenon that has drawn much attention. According to the rich history of research on sarcasm detection, we roughly divide previous related works into four categories: ruled-based, feature-based machine learning, deep learning approaches, and graph-based approaches.

Sarcasm detection was originally solved by using rule-based approaches; such approaches aim to identify sarcasm with fixed patterns. Riloff et al. [3], Maynard and Greenwood [14], Davidov et al. [15] distinguished irony sentences via lexical characteristics, such as the co-occurrence of positive and negative sentiments and hashtags. Bamman and Smith [2], Joshi et al. [16] attempted to capture context incongruity by including extra-linguistic information. Lunando and Purwarianti [5], Mishra et al. [17] augmented traditional linguistic and stylistic features for sarcasm detection with the representative features obtained from readers' movement when sarcasm occurred.

However, rule-based approaches were insufficient to capture complex sarcastic texts. Researchers began to exploit feature-based machine learning methods to solve the problem [18–20]. Reyes and Rosso [18] used n-grams to search for a set of recurrent words carrying sarcasm information. Various machine learning methods were explored in Pawar and Bhingarkar [19], namely, decision tree (DT), random forest (RF), support vector machine (SVM), and Naive Bayes (NB). Farías [20] used both structural features, such as punctuation mark frequency, tweet length, uppercase character amount, and affective characteristics to detect sarcasm.

With the development of neural networks, modern deep-learning methods such as CNNs [21,22], RNNs [23,24], and Recurrent Convolutional Neural Networks(RCNNs) [25] were gradually applied in sarcasm detection. Kumar et al. [26], Duan and Zhao [27] introduced the attention mechanism to neural networks to detect sarcastic comments and revealed that the attention mechanism enhances the performance. These neural networks mainly utilized semantic information, and little structural information was involved.

A new research brand called graph neural networks was proposed to obtain global structural information and the non-continuous relation over long distances. Liang et al. [9], He et al. [28] used GCN to capture the features of the global knowledge in the satire context, while Huang and Carley [29] applied a GAT to model a dependency tree of sarcastic expression. Lou et al. [10] further leveraged both affective and contextual features to draw long-range inconsistent terms over the context for sarcasm detection. However, direct application of dependence may inevitably introduce noisy information and inferiorly in modeling long-distance or disconnected words in the dependency tree. Inspired by the works mentioned above, we propose a novel model ARGAT, which combines the RGAT with GCN to integrate structural and affective information.

3. Methodology

This section describes our proposed Affection Enhanced Relation Graph Attention Network(ARGAT) framework. As demonstrated in Figure 2, for a given input text, we first utilize BiLSTM or Bidirectional Encoder Representations from Transformers (BERT) [30] to extract hidden contextual representations. Then, these hidden representations are fed into RGAT and GCN of our proposed Biaffine-layer structures to integrate the affection and typed syntax information. After that, a biaffine attention module is adopted to reinforce the contextual features and graph features mutually. Finally, we aggregate output representations of the stacked Biaffine-layers via an attention module to align relational graph representations with the affective graph representation. In this way, the model can automatically learn the classification representation with the guidance of affection and relation information.

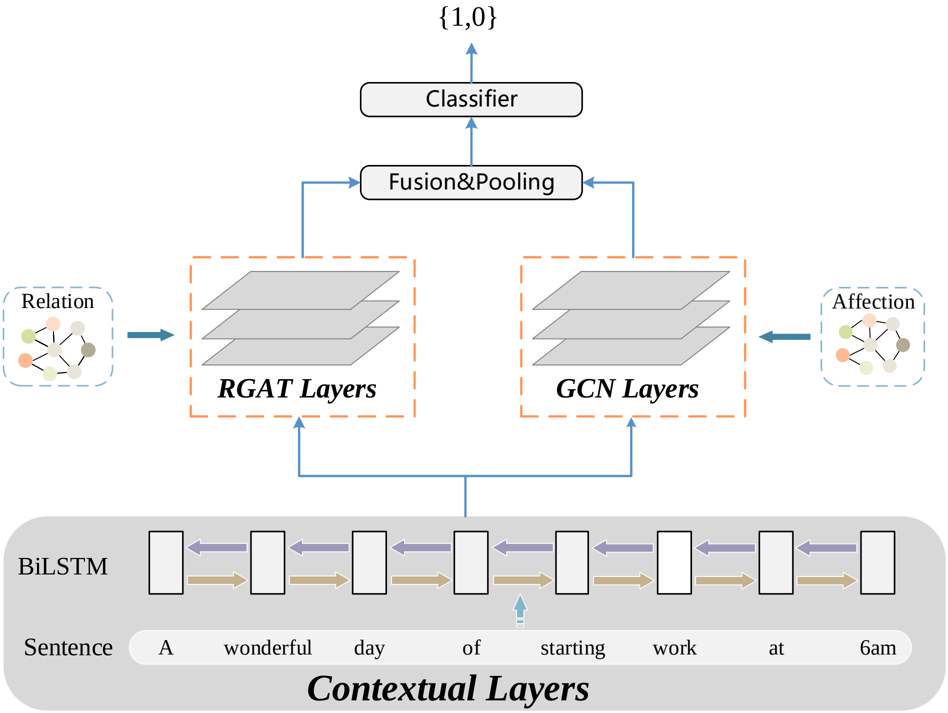

Figure 2. Architecture of the proposed ARGAT framework for sarcasm detection. It consists of a contextual encoder, RGAT layers, GCN layers and a classifier. The RGAT layers and GCN layers can be stacked to n layers.

3.1. Contextual Encoder

We use $\mathbf{x}_k$ to represent the k-th word embedding of m-dimension. Then, we feed the embedding matrix $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \cdot, \mathbf{x}_n\}$ into bidirectional LSTMs or BERT to encode the input sentence into vector representations. For the BiLSTM model:

$$\begin{aligned} \mathbf{H} &= \{\mathbf{h}_1, \mathbf{h}_2, \ldots \cdot, \mathbf{h}_n\} = \mathit{Bi\text{-}LSTM}(\mathbf{x}) \\ \mathbf{h}_t &= [\overrightarrow{\mathbf{h}}_t ; \overleftarrow{\mathbf{h}}_t] \end{aligned} \tag{1}$$

where $h_t \in R^{2d_h}$ denotes the concatenation of forward $\overrightarrow{LSTM}$ hidden representation and backward $\overleftarrow{LSTM}$ hidden representation of x_t in time step t, d_h denotes the dimensionality of hidden representation.

3.2. Relational Graph Attention Network

The graph attention network is a variant of graph neural networks, which leverages masked self-attention layers to encode graph structure. The vanilla GAT uses an adjacent matrix as structural information, thus omitting dependency label features. RGAT incorporates relational features into the attention calculation and aggregation process to obtain more informative representations. The details of the RGAT layer are illustrated in Figure 3.

Input and Output: We denote the relation between word w_i and w_j as $\mathcal{R}_{ij}$, we transform $\mathcal{R}_{ij}$ into a vector $\mathbf{r}_{ij} \in \mathbb{R}^{d_r}$, where d_r is the dimension of relation embeddings. The RGAT takes initial hidden representations $\{h_1^0, h_2^0, \ldots, h_n^0\}$ learnt by BiLSTM or BERT, the adjacent matrix $\mathbf{A}$ and the relation embeddings as input to produce a new set of word features $\{h_1^l, h_2^l, \ldots, h_n^l\}$ as its output after l iteration.

Relation-aware feature aggregation: An RGAT updates the hidden representation at lth layer by calculating a weighted sum of the neighbour states at the guidance of relation embeddings. Specifically, the aggregation process of a multi-head attention-based RGAT can be described as:

$$\mathbf{h}_i^l = \overset{Z}{\underset{z=1}{||}} \sigma(\sum_{j \in N(i)} \alpha_{ij}^{lz} (\mathbf{W}_V^{lz} \mathbf{h}_j^{l-1} + \mathbf{W}_{V_r}^{lz} r_{ij})) \tag{2}$$

where $\mathbf{W}_V^{lz} \in \mathbb{R}^{\frac{d}{Z} \times d}$ and $\mathbf{W}_{V_r}^{lz} \in \mathbb{R}^{\frac{d}{Z} \times d_r}$ are parameter matrices. α_{ij}^{lz} is the normalized attention scores calculated by the attention mechanism, which combines node-aware attention scores and relation-aware attention scores:

$$\alpha_{ij}^{lz} = \frac{exp(e_{ij}^N + e_{ij}^R)}{\sum_{j' \in \mathbf{N}(i)} exp(e_{ij'}^N + e_{ij'}^R)} \tag{3}$$

where e_{ij}^N and e_{ij}^R are node-aware attention scores and relation-aware scores, respectively. We calculate the two types of attention scores according to the equations denoted as:

$$e_{ij}^N = \begin{cases} f(h_i^{l-1}, h_j^{l-1}), & j \in \mathbf{N}(i) \\ -inf, & otherwise \end{cases} \tag{4}$$

$$e_{ij}^R = \begin{cases} f(h_i^{l-1}, r_{ij}), & j \in \mathbf{N}(i) \\ -inf, & otherwise \end{cases} \tag{5}$$

For $\mathbf{h}^l = \{\mathbf{h}_0^l, \mathbf{h}_1^l, \ldots \cdot, \mathbf{h}_n^l\}$, we employ two normalization layers to get better representations:

$$\mathbf{h}^l = Norm(\mathbf{h}^l + \mathbf{h}^{l-1}) \tag{6}$$

$$\mathbf{h}^l = Norm(\mathbf{h}^l + FFN(\mathbf{h}^l)) \tag{7}$$

where $FFN(\mathbf{x}) = Relu(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$ is a two-layer multi-layer perceptron(MLP) with the activation function *Relu*, *Norm* is a normalization layer, W_1 and W_2 are trainable parameters, and $\mathbf{h}^l$ in Equation (9) is the final output the RGAT layer at iteration l. In this way, we take both node features and relation features into consideration for better feature aggregation.

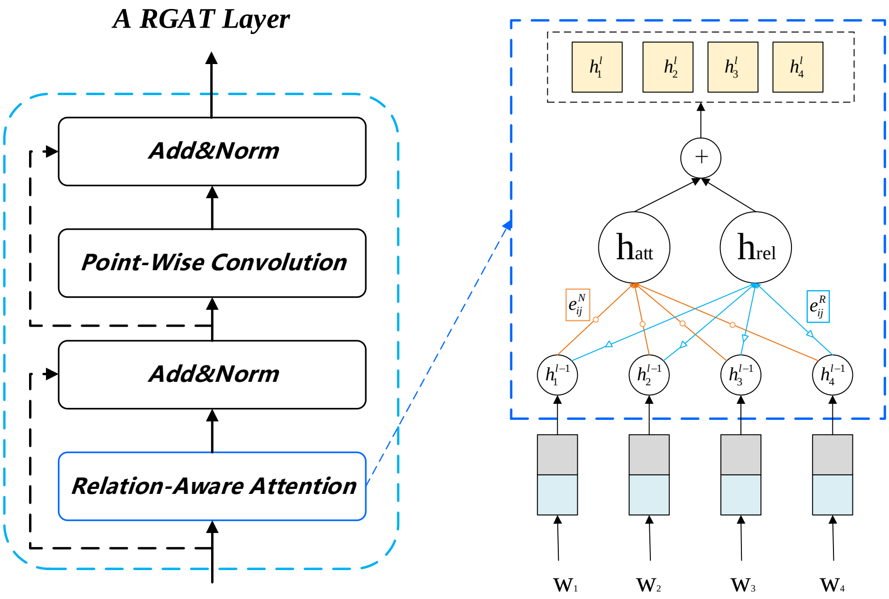

Figure 3. Details of an RGAT layer and its corresponding relation-aware attention operation.

3.3. Affective Graph Convolutional Network

We leverage the affection information to explore the contextual incongruity of sarcastic expressions. Given a sentence s consisting of n words $s = \{w_i\}_{i=1}^{n}$, we construct an affective guided graph and corresponding adjacency matrix $A_{i,j}^{a} \in R^{n\times n}$, and the scores of words are obtained from SenticNet [31]:

$$\mathbf{A}_{i,j}^{a} = abs\big(\mathbf{S}(w_i) - \mathbf{S}(w_j)\big) \tag{8}$$

where $S(w_i) \in [-1, 1]$ represents the affective score retrieved from SenticNet. Note that $S(w_i) = 0$ if w_i is not contained in the knowledge obtained from SenticNet. abs() represents absolute value calculation.

After obtaining the affective adjacency matrix, we feed it into the GCN architecture to leverage the long-range affective incongruity expressions, and the process is defined as:

$$\mathbf{g}^{l} = ReLU(\tilde{\mathbf{A}}^{d} ReLU(\tilde{\mathbf{A}}^{a}\mathbf{g}^{l-1}\mathbf{W}_{a}^{l} + \mathrm{b}_{a}^{l})\mathbf{W}_{d}^{l} + \mathrm{b}_{d}^{l}) \tag{9}$$

where $g^{l} \in \mathbb{R}^{n\times 2d_h}$ is the hidden graph representation in the l-th GCN layer, and the initial input nodes of the first GCN layer are the contextual representations learnt by BiLSTM or BERT, i.e., $\mathbf{g}^{0} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$. $\tilde{\mathbf{A}}^{a}$ is the normalized affective adjacency matrix: $\tilde{\mathbf{A}}_{i}^{a} = \mathbf{A}_{i}^{a}/(\mathbf{E}_i - 1)$. $\mathbf{E}_i = \sum_{j=1}^{n} A_{i,j}^{a}$ is the degree of adjacency matrix $\mathbf{A}_{i}^{a}$. $\mathbf{W}^{l} \in \mathbb{R}^{2d_h\times 2d_h}$, $\mathbf{b}^{l} \in \mathbb{R}^{2d_h}$ are the trainable parameters of the l-th GCN.

Then, we apply a normalization layer to extract higher-level features:

$$\mathbf{g}^{l} = Norm(\mathbf{g}^{l} + \mathbf{g}^{l-1}) \tag{10}$$

3.4. Classification Model

Before sending features into the classifier, we fuse two types of hidden representations by a concatenative score function as follows:

$$\mathbf{r} = pool\Big(\sigma(\mathbf{W}_r[\mathbf{h}^{l'};\mathbf{g}^{l'}] + \mathbf{b}_r)\Big) \tag{11}$$

where $\mathbf{W}_r \in \mathbb{R}^{2d_h\times 4d_h}$ is the trainable parameter, [;] means the concatenation operation, *pool*—here, we apply average-pooling function to reserve salient features and reduce dimensions. Then, we send $\mathbf{r}$ to a fully connected network, which takes the fused representation $\mathbf{r}$ as input and computes the probability of sarcastic and non-sarcastic classes:

$$\hat{\mathbf{y}} = softmax(\mathbf{W}_o\mathbf{r} + \mathbf{b}_o) \tag{12}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{d_p}$ is the predicted sarcastic probability for the input sentence, d_p is the dimensionality of sarcasm labels. $\mathbf{W}_o \in \mathbb{R}^{d_p\times 2d_h}$, and $\mathbf{b}_o \in \mathbb{R}^{d_p}$ are trainable parameters. We use the cross-entropy loss as our objective and try to minimize the loss via standard gradient descent algorithm during training the model:

$$min\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{d_p} y_i^j log\hat{y}_i^j + \lambda||\Theta||^2 \tag{13}$$

where N is the trainable data size. y_i and $\hat{y}_i$ represent the ground-truth and predicted label distribution of sentence i. Θ denotes all trainable parameters of the model, and λ represents the hyperparameter of L_2-regularization.

4. Experiments

4.1. Datasets

To conduct the experiments in a fair condition, we keep the statistics of the Experimental data to be consistent with [10]. We conduct experiments on six benchmark datasets of three different sources to evaluate our model, and each dataset is divided into a train set and test set. The details are shown in Table 1.

- IAC(Internet Argument Corpus): The dataset is from a forum used for political debating and voting, which is characterized by long sentences with satire style. We use two versions of the dataset from [32], which are denoted as IAC-V1 (https://nlds.soe.ucsc.edu/sarcasm1, in 1 January 2022) and IAV-V2 (https://nlds.soe.ucsc.edu/sarcasm2, in 1 January 2022), respectively.
- Tweets: We use two datasets collected by [3,33]. We get all the tweets through Twitter API with the provided tweet IDs (http://api.twitter.com/, in 1 January 2022)
- Reddit: We use two subsets (i.e., movies and technology) of the Reddit dataset (http://nlp.cs.princeton.edu/SARC, in 1 January 2022) provided by [34] for sarcasm detection.

Table 1. Statistics of the experimental data.

Dataset	Train		Test	
	Sarcasm	None	Sarcasm	None
IAC-V1	862	859	97	94
IAC-V2	2947	2921	313	339
Tweets-1 (Riloff)	282	1051	35	113
Tweets-2 (Ptáček)	23,456	24,387	2569	2634
Reddit-1 (movies)	5521	5607	1389	1393
Reddit-2 (technology)	6419	6393	1596	1607

4.2. Baselines

We compare our proposed model with the following algorithms:

- **NBOW** Tay et al. [6] use a simple neural bag-of-words baseline that sums all the word embeddings and passes the summed vector into a simple logistic regression layer.
- **CNN** is a vanilla Convolutional Neural Network with max-pooling.
- **GRNN** Zhang et al. [35] extracts local syntactic and semantic information with a Bidirectional Gated Recurrent Unit.
- **CNN-LSTM-DNN** Ghosh and Veale [11] combines CNN, LSTM, and Deep Neural Network via stacking for prediction.
- **ATT-LSTM** Yang et al. [36] adopt a LSTM model with a neural attention mechanism applied to all the LSTM hidden outputs.
- **SIARN** [6] is an attention-based neural model that looks in-between instead of across.
- **MIRAN** [6] is a Multi-dimensional Intra-Attention Recurrent Network based on the intuition of compositional learning by leveraging intra-sentence relationships.
- **SAWS** Pan et al. [7] proposes a novel model based on self-attention mechanism of weighted snippets.
- **ADGCN** Lou et al. [10] proposed a GCN-based model to draw long-range incongruity patterns and inconsistent expressions over the context for sarcasm detection by means of interactively modeling the affective and dependency information.

We can roughly divide the baselines into two categories, the first four models are basic models, the next four models are attention-based models, and the last model is the graph-based model.

4.3. Settings

We implement our model in PyTorch [37] and use spaCy to deal with tokenization and dependency parsing of sentences. All experiments are running on NVIDIA GeForce RTX 3090 GPU. We adopt a similar experimental setting as previous work [10]. Concerning the BiLSTM-based contextual encoder, 300-dimensional GloVe [38] vectors are applied for word representation. The dropout rate on input word embedding is 0.1. The dimension of relation embeddings is set to 30. The dimension of hidden representations is set to 300. The coefficient λ of L_2 regularization is set to 0.00001. The Adam optimizer with a learning rate of 0.001 is adopted for model training. The iteration of the ARGAT layer is set to 3. For the RGAT layer, we set five attention heads. The mini-batch size is 128 for Tweets-2 and 32 for the other five datasets. The maximum sentence length is set to 80. We perform precision, recall, macro F1 score, and accuracy (Acc.) to measure the performance of models.

4.4. Results

Tables 2–4 show the experimental results on six benchmark datasets. We observe that our model achieves state-of-the-art performance in all six datasets; the best-improved results of the four metrics are more than 4%, compared with the previous state-of-the-art version. It is worth noting that, 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 (the results of basic models are from Tay et al. [6]). Experiments show that our proposed model, which simultaneously considers the affection information and relational dependencies, contributes to the performance of sarcasm detection.

Table 2. Experimental results on IAC datasets. The best results are in bold and the results of four basic models are from [6].

Model	IAC-V1				IAC-V2			
	Precision (%)	Recall (%)	F1 (%)	Acc. (%)	Precision (%)	Recall (%)	F1 (%)	Acc. (%)
NBOW	57.17	57.03	57.00	57.51	66.01	66.03	66.02	66.09
CNN	58.21	58.00	57.95	58.55	68.45	68.18	68.21	68.56
GRNN	56.21	56.21	55.96	55.96	62.26	61.87	61.21	61.37
CNN-LSTM-DNN	55.50	54.60	53.31	55.96	64.31	64.33	64.31	64.38
ATT-LSTM	58.98	57.93	57.23	59.07	70.04	69.62	69.63	69.96
SIARN	63.94	63.45	60.52	62.69	72.17	71.81	71.85	72.10
MIARN	63.88	63.71	63.18	63.21	72.92	72.93	72.75	72.75
SAWS	66.22	65.65	65.60	66.13	73.25	73.40	73.43	73.55
ADGCN	68.08	68.08	68.06	68.06	76.96	76.98	76.97	76.99
ARGAT (**proposal**)	**72.26**	**72.26**	**72.25**	**72.25**	**78.41**	**78.19**	**78.21**	**78.22**

Table 3. Experimental results on Tweets datasets. The best results are in bold and the results of four basic models are from [6].

Model	Tweets (Riloff)				Tweets (Ptacek)			
	Precision (%)	Recall (%)	F1 (%)	Acc. (%)	Precision (%)	Recall (%)	F1 (%)	Acc. (%)
NBOW	71.28	62.37	64.13	79.23	80.02	79.06	79.43	80.39
CNN	71.04	67.13	68.55	79.48	82.13	79.67	80.39	81.65
GRNN	66.32	64.74	65.40	76.41	82.06	81.02	82.43	82.20
CNN-LSTM-DNN	69.76	66.62	67.81	78.72	79.65	79.12	79.20	79.94
ATT-LSTM	69.76	66.62	67.81	78.72	81.62	81.45	81.56	81.56
SIARN	73.82	73.26	73.24	82.31	82.62	82.51	82.59	82.59
MIARN	73.34	68.34	70.10	80.77	82.34	82.72	82.78	82.78
SAWS	74.69	74.08	74.34	81.72	83.25	83.40	83.43	83.55
ADGCN	74.81	76.22	75.45	81.75	83.85	83.85	83.85	83.86
ARGAT (**proposal**)	**83.19**	**76.24**	**79.78**	**85.81**	**84.28**	**84.28**	**84.28**	**84.28**

We observe that basic models are much worse than attention-based and graph-based models. Those models only capture local semantic information, which is insufficient to recognize complex irony expressions or long-term incongruity between words. The attention-based models achieve a slight improvement compared with the basic models, demonstrating the effectivity of the attention mechanism. The SAWS shows that sentence snippets capture more useful sarcastic information than words among the attention-based methods. The graph-based methods (i.e., ADGCN and ARGAT), which simultaneously utilize the affection and semantic information, show the strong power to capture incongruity among word pairs. The performances of graph-based models suggest that affective information plays an essential role in sarcasm detection. Noting that all models get the best results in Tweets sets, and the performances decrease when the length of sentences is too long (IAC) or too short (Reddit), this means that more ideas should be explored when the sentence contains little information or redundant information. The proposed ARGAT achieves comparison performances in both the small dataset (Riloff) and the big dataset (Ptacek). In contrast, the others only perform well in the big dataset, indicating that the graph-based method captures more features for sarcasm detection.

Table 4. Experimental results on Reddit datasets. The best results are in bold and the results of four basic models are from [6].

Model	Reddit (/r/Movies)				Reddit (/r/Technology)			
	Precision (%)	**Recall (%)**	**F1 (%)**	**Acc. (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**	**Acc. (%)**
NBOW	67.33	66.56	66.82	67.52	65.45	65.62	65.52	66.55
CNN	65.97	65.97	65.97	66.24	65.88	62.90	62.85	66.80
GRNN	66.16	66.16	66.16	66.42	66.56	66.73	66.66	67.65
CNN-LSTM-DNN	68.27	67.87	67.95	68.50	66.14	66.73	65.74	66.00
ATT-LSTM	68.11	67.87	67.94	68.37	68.20	68.78	67.44	67.22
SIARN	69.59	69.48	69.52	69.84	69.35	70.05	69.22	69.57
MIARN	69.68	69.37	69.54	69.90	68.97	69.30	69.09	69.91
SAWS	71.79	71.77	71.76	71.77	72.50	72.45	72.45	72.48
ADGCN	74.48	74.58	74.47	74.48	75.59	75.59	75.58	75.59
ARGAT (**proposal**)	**75.82**	**75.82**	**75.81**	**75.82**	**76.13**	**76.13**	**76.13**	**76.13**

4.5. Impact of Stacked Number of RGAT and GCN

To study the effect of the number of RGAT layers and GCN layers on the performance of the proposed model, we adjust the layer number from 1 to 7 and recode the results on six datasets in Figure 4. We initially set the number of GCN layers to 3 and then adjust the number of RGAT layers from 1 to 7. Note that three RGAT layers perform overall better than other layers, and thus we set the number of RGAT layers as 3. One RGAT layer performs unsatisfactorily on all datasets, which indicates that simple network structure is insufficient to exploit decent sarcastic features. Additionally, when the layer is greater than 3, the performance fluctuates and declines with the increasing number. This implies that increasing the number of RGAT layers is likely to reduce the model's learning capabilities due to the sharp increase of model parameters. Then, we fix the number of RGAT layers as three and vary the number of GCN layers from 1 to 7, and results show that three GCN layers perform the best overall. According to the experimental results, we set the number of RGAT and GCN layers to 3. The depths of RGAT and GCN layers have less impact on large datasets. The accuracy of Ptacek stays in a small range when the number of layers changes. Conversely, it fluctuates heavily in IAC-1 and Riloff datasets.

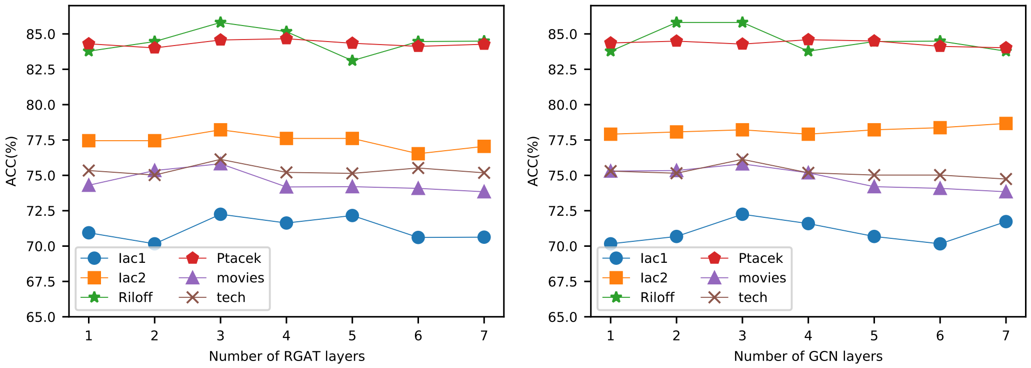

Figure 4. Impact of the number of stacked layers. The left figure fixes the GCN layers to 3 and records the model's performance with a different number of RGAT layers. The right figure fixes the RGAT layers to 3 and records the results of the model with a different number of GCN layers.

4.6. Ablation Study

To analyze the impact that different components of the proposed ARGAT bring to the performance, we conduct an ablation study and report the results in Table 5. Note that removing the GCN structure from the syntactic information sharply degrades the performance, indicating that affective information is significant in learning sarcastic expressions. Additionally, removing the RGAT structure with affection information refinement leads to considerably poorer performance. This implies that both dependency relation information and affection information help extract the linchpin clues of incongruity expressions.

Table 5. Accuracy results of ablation study. The results of the proposed model are in bold. $\mathcal{R}$ denotes the RGAT structure, $\mathcal{A}$ denotes affective GCN structure.

Model	IAC-V1	IAC-V2	Tweets-1	Tweets-2	Reddit-1	Reddit-2
ARGAT	**72.25**	**78.22**	**85.81**	**84.28**	**75.82**	**76.13**
w/o $\mathcal{R}$	71.11	77.45	82.43	84.09	74.04	74.93
w/o $\mathcal{A}$	70.06	76.99	81.76	83.87	73.15	73.40

5. Conclusions

This paper proposes a graph-based structure that jointly utilizes affective information and dependency relation information to learn long-distance incongruity in sarcasm detection. The GCN and RGAT structures in the proposed model effectively capture inconsistent relations according to corresponding graph information of affection and dependency. Extensive experiments demonstrate that our proposed ARGAT model outperforms strong state-of-the-art baselines such as ADGCN and MIARN. We plan to introduce external knowledge into the model to solve the insufficiency in short sentences in sarcasm detection for future work.

Author Contributions: Conceptualization, G.L. and B.L.; Data curation, G.L.; Methodology, F.L., W.C. and B.L.; Visualization, F.L. and W.C.; Writing—original draft, G.L.; Writing—review and editing, F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gibbs, R.W. On the psycholinguistics of sarcasm. *J. Exp. Psychol. Gen.* **1986**, *115*, 3–15. [CrossRef]
2. Bamman, D.; Smith, N.A. Contextualized Sarcasm Detection on Twitter. In Proceedings of the International AAAI Conference on Web and Social Media (ICWSM), Oxford, UK, 26–29 May 2015.
3. Riloff, E.; Qadir, A.; Surve, P.; Silva, L.D.; Gilbert, N.; Huang, R. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), Seattle, WA, USA, 26 August 2013.
4. González-Ibáñez, R.I.; Muresan, S.; Wacholder, N. Identifying Sarcasm on Twitter: A Closer Look. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL), Portland, OR, USA, 23 June 2011.
5. Lunando, E.; Purwarianti, A. Indonesian social media sentiment analysis with sarcasm detection. In Proceedings of the 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Depok, Indonesia, 23–25 October 2013; pp. 195–198.
6. Tay, Y.; Luu, A.T.; Hui, S.C.; Su, J. Reasoning with Sarcasm by Reading In-Between. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), Melbourne, Australia, 15–20 July 2018; Gurevych, I., Miyao, Y., Eds.; Association for Computational Linguistics: Melbourne, Australia, 2018; Volume 1: Long Papers; pp. 1010–1020. [CrossRef]

7. Pan, H.; Lin, Z.; Fu, P.; Wang, W. Modeling the Incongruity Between Sentence Snippets for Sarcasm Detection. In Proceedings of the ECAI 2020—24th European Conference on Artificial Intelligence, Santiago de Compostela, Spain, 29 August–8 September 2020; Giacomo, G.D., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J., Eds.; IOS Press: Santiago de Compostela, Spain, 2020; Volume 325, pp. 2132–2139. [CrossRef]
8. Babanejad, N.; Davoudi, H.; An, A.; Papagelis, M. Affective and Contextual Embedding for Sarcasm Detection. In Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020), Barcelona, Spain, 8–13 December 2020; pp. 225–243. [CrossRef]
9. Liang, B.; Lou, C.; Li, X.; Gui, L.; Yang, M.; Xu, R. Multi-Modal Sarcasm Detection with Interactive In-Modal and Cross-Modal Graphs. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20 October 2021; pp. 4707–4715. [CrossRef]
10. Lou, C.; Liang, B.; Gui, L.; He, Y.; Dang, Y.; Xu, R. Affective Dependency Graph for Sarcasm Detection. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Montréal, QC, Canada, 11–15 July 2021; pp. 1844–1849. [CrossRef]
11. Ghosh, A.; Veale, T. Fracking Sarcasm using Neural Network. In Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@NAACL-HLT 2016, San Diego, CA, USA, 16 June 2016; Balahur, A., der Goot, E.V., Vossen, P., Montoyo, A., Eds.; The Association for Computer Linguistics: Berlin, Germany, 2016; pp. 161–169. [CrossRef]
12. Busbridge, D.; Sherburn, D.; Cavallo, P.; Hammerla, N.Y. Relational Graph Attention Networks. *arXiv* **2019**, arXiv:1904.05811.
13. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907.
14. Maynard, D.; Greenwood, M.A. 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 2014), Reykjavik, Iceland, 26–31 May 2014; pp. 4238–4243.
15. Davidov, D.; Tsur, O.; Rappoport, A. Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In Proceedings of the CoNLLco 2010, Uppsala, Sweden, 15–16 July 2010.
16. Joshi, A.; Sharma, V.; Bhattacharyya, P. 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 (ACL), Beijing, China, 26–31 July 2015.
17. Mishra, A.; Kanojia, D.; Nagar, S.; Dey, K.; Bhattacharyya, P. Harnessing Cognitive Features for Sarcasm Detection. *arXiv* **2016**, arXiv:abs/1701.05574.
18. Reyes, A.; Rosso, P. Making objective decisions from subjective data: Detecting irony in customer reviews. *Decis. Support Syst.* **2012**, *53*, 754–760. [CrossRef]
19. Pawar, N.; Bhingarkar, S. Machine Learning based Sarcasm Detection on Twitter Data. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; pp. 957–961.
20. Farías, D.I.H. Irony and Sarcasm Detection on Twitter: The Role of Affective Content. *Proces. Leng. Natural* **2019**, *62*, 107–110.
21. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1746–1751. [CrossRef]
22. Das, D.; Clark, A.J. Sarcasm Detection on Flickr Using a CNN. In Proceedings of the 2018 International Conference on Computing and Big Data, Tibet, China, 20–22 April 2018.
23. Porwal, S.; Ostwal, G.; Phadtare, A.; Pandey, M.; Marathe, M. Sarcasm Detection Using Recurrent Neural Network. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 14–15 June 2018; pp. 746–748.
24. Hiai, S.; Shimada, K. Sarcasm Detection Using RNN with Relation Vector. *Int. J. Data Warehous. Min.* 2019, *15*, 66–78. [CrossRef]
25. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
26. Kumar, A.; Narapareddy, V.T.; Veerubhotla, A.S.; Malapati, A.; Neti, L.B.M. Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM. *IEEE Access* **2020**, *8*, 6388–6397. [CrossRef]
27. Duan, S.; Zhao, H. Attention Is All You Need for Chinese Word Segmentation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; pp. 3862–3872. [CrossRef]
28. He, S.; Guo, F.; Qin, S. Sarcasm Detection Using Graph Convolutional Networks with Bidirectional LSTM. In Proceedings of the 2020 3rd International Conference on Big Data Technologies, Qingdao, China, 18–20 September 2020.
29. Huang, B.; Carley, K.M. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. *arXiv* **2019**, arXiv:abs/1909.02606.
30. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:abs/1810.04805.
31. Cambria, E.; Li, Y.; Xing, F.Z.; Poria, S.; Kwok, K. SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 105–114. [CrossRef]

32. Lukin, S.M.; Walker, M.A. Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue. *arXiv* **2017**, arXiv:abs/1708.08572.
33. Ptácek, T.; Habernal, I.; Hong, J. Sarcasm Detection on Czech and English Twitter. In Proceedings of the COLING 2014, 25th International Conference on Computational Linguistics, Technical Papers, Dublin, Ireland, 23–29 August 2014; Hajic, J., Tsujii, J., Eds.; ACL: Baltimore, MD, USA, 2014; pp. 213–223.
34. Khodak, M.; Saunshi, N.; Vodrahalli, K. A Large Self-Annotated Corpus for Sarcasm. *arXiv* **2018**, arXiv:1704.05579.
35. Zhang, M.; Zhang, Y.; Fu, G. Tweet Sarcasm Detection Using Deep Neural Network. In Proceedings of the COLING 2016, 26th International Conference on Computational Linguistics, Technical Papers, Osaka, Japan, 11–16 December 2016; Calzolari, N., Matsumoto, Y., Prasad, R., Eds.; ACL: Osaka, Japan, 2016; pp. 2449–2460.
36. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.J.; Hovy, E.H. Hierarchical Attention Networks for Document Classification. In Proceedings of the NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; Knight, K., Nenkova, A., Rambow, O., Eds.; The Association for Computational Linguistics: Osaka, Japan, 2016; pp. 1480–1489. [CrossRef]
37. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R., Eds.; pp. 8024–8035.
38. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, 25–29 October 2014; Moschitti, A., Pang, B., Daelemans, W., Eds.; ACL: Baltimore, MD, USA, 2014; pp. 1532–1543. [CrossRef]