DialogueTRGAT: Temporal and Relational Graph Attention Network for Emotion Recognition in Conversations

Junjun Kang[1,2] and Fang Kong[1,2](✉)

[1] Laboratory for Natural Language Processing, Soochow University, Suzhou, China
202042270510@stu.suda.edu.cn, kongfang@suda.edu.cn
[2] School of Computer Science and Technology, Soochow University, Suzhou, China

Abstract. Emotion Recognition in Conversations (ERC) is the task of identifying the emotions of utterances from speakers in a conversation, which is beneficial to a number of applications, including opinion mining over conversations, developing empathetic dialogue systems, and so on. Many approaches have been proposed to handle this problem in recent years. However, most existing approaches either focus on using RNN-based models to simulate temporal information change in the conversation or graph-based models to take the relationships between the utterances of the speakers into account. In this paper, we propose a temporal and relational graph attention network, named DialogueTRGAT, to combine the strengths of RNN-based models and graph-based models. DialogueTRGAT can better model the intrinsic structure and information flow within a conversation for better emotion recognition. We conduct experiments on two benchmark datasets(IEMOCAP, MELD), and the experimental results demonstrate the great effectiveness of our approach compared with several competitive baselines.

Keywords: Emotion recognition · Graph model · Dialogue modeling

1 Introduction

As a fundamental aspect of human communication, emotions play important roles in our daily lives and are crucial for more natural human-computer interaction. In recent years, with the development of social networks and the construction of large datasets for dialogue, emotion recognition in conversations has become an emerging task for the research community due to its applications in several important tasks such as opinion mining over conversations(Kumar et al. [7]), building an emotional and empathetic dialogue system (Majumder et al.[8], Zhou et al. [16]), and so on.

Emotion recognition in conversations aims to identify the emotion of each utterance in conversations involving two or more speakers. Different from other emotion recognition tasks, conversational emotion recognition is not only for utterances, but also depends on the context and the states of speakers. With

W. Lu et al. (Eds.): NLPCC 2022, LNAI 13551, pp. 460–472, 2022.
https://doi.org/10.1007/978-3-031-17120-8_36

the development of deep learning technologies, many approaches have been proposed to handle this problem. They can generally be divided into two categories: RNN-based methods and graph-based methods. But they all have their disadvantages. For the RNN-based methods, they use RNN-based models encoding the utterances temporally, but because RNN has long-term information propagation issues, they tend to aggregate relatively limited information from the nearest utterances for the target utterance, so can't model the long-term dependency within the conversation. For graph-based models, they adopt neighborhood-based graph convolutional networks to model conversational context. In these models, they construct relational edges to directly build the correlation between utterances, thereby alleviating the long-distance dependency issues. But they neglect the sequential characteristic of conversation.

According above discussion, in this paper, we try to combine the advantage of RNN-based models and graph-based models to complement each other. We propose a temporal and relational graph attention network, named DialogueTRGAT to model the conversation as temporal graph structure. In particular, like RNN-based models, we gather historical context information for each target utterance based on the their temporal position in dialogue. For each target utterance, it only receives information from some previous utterances and cannot propagate information backward. In order to model the inter-speaker dependency[1] and self-dependency[2] between utternaces, we follow Ishiwatari et al. [5], use the message aggregation principle of relational graph attention networks(RGAT) to aggregate context information for the target utterance based on the speaker identity between itself and the previous utterances.

Compared with the traditional static graph networks, DialogueTRGAT enables the targe utterance can indirectly attend to the remote context without having to stack too many graphical layers. And it can be seen as an extension of traditional graph neural networks with an additional focus on the temporal dimension. We argue that DialogueTRGAT can better model the flow of information in dialogue and aggregate more meaningful historical contextual information for each target utterance, leading to better emotion recognition.

2 Related Work

We generally classify related works into two categories according to the method of modeling the dialogue context.

RNN-Based Models: Many works capture contextual information in utterance sequences. ICON [3] uses an RNN-based memory network to model contextual information that incorporates inter-speaker and self-dependency. HiGRU [6] propose a hierarchical GRU framework, where lower-level GRU is utterance encoder and the contexts of utterances are captured by the upper-level GRU. Considering the individual speaker state change throughout the conversation, Majumder

[1] the speaker's emotions are influenced by others.
[2] emotional inertia of individual speakers.

et al. [9] propose DialogueRNN, which utilizes GRUs to update speakers' states, the global state of the conversation and emotional dynamics. DialogCRN [4] uses LSTM to encode the conversational-level and speaker-level context respectively for each utterance and proposes to apply LSTM-based reasoning modules to extract and integrate clues for emotional reasoning.

Graph-Based Models: Many works model the conversational context by designing a specific graphical structure. For example, DialogueGCN [2] models two relations between speakers: self and inter-speaker dependencies, and utilizes graph network to model the graph constructed by these relations. Base on DialogueGCN, DialogueRGAT [5] uses relational position encoding to combine position information into the graph network structure. ConGCN [15] regards both speakers and utterances as graph nodes, the context-sensitive dependence and the speaker-sensitive dependence are modeled as edges to construct graphical structure. Shen et al. [12] model the dialogue as a directed acyclic graph and use directed acyclic graph neural networks [14] to model the conversation context. Our work is closely related to the graph-based models. But like RNN-based models, our model focuses more on the temporality of information propagation in graphical models than the above-mentioned models.

3 Methodology

3.1 Problem Definition

Given the transcript of a conversation along with speaker information of each constituent utterance, the task is to identify the emotion of each utterance from several pre-defined emotions. Formally, given the input sequence of N number of utterances and corresponding speakers$\{(u_1, s_1), (u_2, s_2), \ldots, (u_N, s_N)\}$, where each utterance $u_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,T}\}$ consists of T words $u_{i,j}$ and spoken by speaker s_i, $s_i \in S$, where S is the set of the conversation speakers. The task is to predict the emotion label e_i for each target utterance u_i based on its historical context $\{u_1, u_2, ..., u_{i-1}\}$ and the corresponding speaker information.

3.2 Model

Emotion Recognition in Conversations, as a conversational utterance-level understanding task, most of the recent methods consist of three common components including (i) feature extraction for utterances (ii) conversational context encoder, and (iii) the emotion classifier. Our model also follows the paradigm. Figure 1 shows the overall architecture of our model.

Utterance-Level Feature Extraction. Convolutional Neural Networks (CNNs) are effective in learning high-level abstract representations of sentences from constituting words or n-grams. Following (Ghosal et al. [3] Hazarika et al., [9] Majumder et al.[2]), we use a single convolutional layer followed by max-pooling and a fully connected layer to obtain the feature representations for the utterances. We denote $\{h_i\}_{i=1}^{N}$, $h_i \in \mathbb{R}^{d_u}$ as the representation for N utterances.

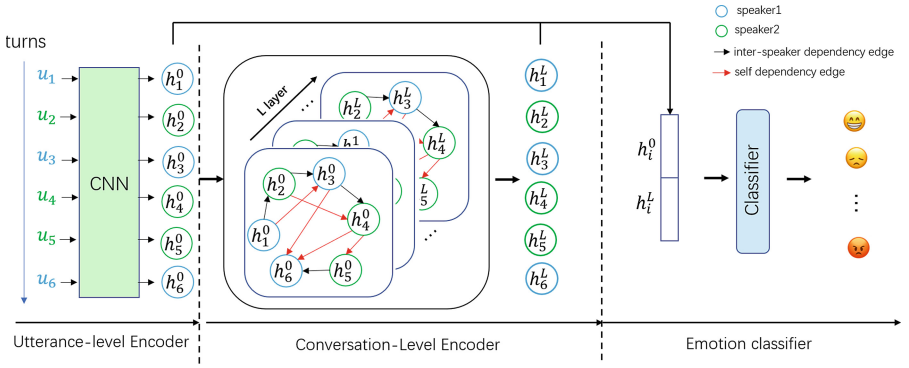


Fig. 1. The overall architecture of our model.

Sequential and Speaker-Level Context Encoder. We model the conversation as a temporal graph structure and propose a temporal and relational graph attention network(TRGAT) to model the controversial context and gather historical context information for target utterance. Our graph structure transmit information in temporal order to imitate the process of dynamic conversation, which can preserve the temporal change information of conversation. The relational graph attention network's message aggregation principle captures both self-dependency and inter-speaker dependency for target utterance.

Graph Structure:Node: Each utterance in a conversation is represented as a node $v_i \in V$. Each node v_i is initialized with the utterance representation h_i. The representation can updated by aggregating the representations of previous utterance within a certain context window through our TRGAT layers. The updated representation is donated as h_i^l , where l denotes the number of TRGAT layers. So we also denoted h_i as h_i^0.

Edges: For each target utterance u_i, its emotion is most likely to be influenced by the utterance between the previous utterance spoken by s_i and the utterance u_{i-1}. We use these utterances as the historical window to aggregate context information for utterance u_i. We argue that it is more reasonable compared to using a fixed-size history window. We regard u_j as the latest utterance spoken by s_i before u_i ($s_j = s_i$). Then for each utterance u_τ in between u_j and u_{i-1}, we make a directed edge from u_τ to u_i. Depending on whether the speaker of u_τ is the same as the speaker of u_i, we divide the edges into two types. Formally, the above process can be expressed by the following formulas:

$$j = \max_j j < i \ \& \ s_j = s_i \tag{1}$$

$$historical\ window = [u_j, u_{j+1}, ..., u_{i-1}] \tag{2}$$

$$edges = \{u_\tau \rightarrow u_i\}_{\tau=j}^{i-1} \tag{3}$$

$$edge\ type = \begin{cases} 0 & s_\tau = s_i \\ 1 & s_\tau \neq s_i \end{cases} \qquad \tau \in [j, j+1, ..., i-1] \tag{4}$$

To ensure that the representation of the utterance node at layer l can also be informed by the corresponding representation at layer $l-1$, we add a self-loop edge to u_i. We set the edge type as 0.

Node(utterance) Representation Update Scheme: At each layer of TRGAT, We aggregate historical context information for each utterance in temporal order, and allow each utterance to gather information from neighbors(utterances in its historical window) and update their representations. So the representation of utterances would be computed recurrently from the first utterance to the last one. Follow DialogueRGAT [5], in order to model the self and inter-speaker dependency between utterances, we use the message aggregation principle of relational graph attention networks(RGAT) to aggregate context information for each utterance.

In l-th layer, for each target utterance u_i, the attention weights between u_i and u_τ and the attention weights between u_i and itself are calculated as follows:

$$e^l_{i,\tau} = LeakyReLu\left((a^l_r)^T[W^l_r h^{l-1}_i || W^l_r h^l_\tau]\right) \quad edge\ type(s_\tau, s_i) = r \in \{0, 1\} \tag{5}$$

$$e^l_{i,i} = LeakyReLu\left((a^l_0)^T[W^l_0 h^{l-1}_i || W^l_0 h^{l-1}_i]\right) \tag{6}$$

$$\alpha^l_{i,\tau} = softmax_i(e^l_{i,\tau}) \tag{7}$$

$$\alpha^l_{i,i} = softmax_i(e^l_{i,i}) \tag{8}$$

where $\alpha^l_{i,\tau}$ denotes the edge(attention) weight from u_τ to the target utterance u_i in layer l. $\alpha^l_{i,i}$ denotes self-loop edge weight for u_i in layer l, W^l_r denotes a parameterized weight matrix for edge type r in layer l. a^l_r denotes a parameterized weight vector for edge type r in layer l, W_r and a_r not shared across the layers. T represents transposition. $||$ represents the concatenation operation of vectors. A softmax function is used to obtain the incoming edges whose total weight is 1.

It is worth noting that the attention weights between u_i and u_τ are based on the u_i's hidden state[3] in the $l-1$-th layer (h^{l-1}_i)and the u_τ's hidden state in the l-th layer(h^l_τ). The reasons are as follows: we update hidden state for each utterance based on their temporal position and the temporal position of u_τ is in front of u_i. So the hidden state for u_τ has been updated before u_i, donated h^l_τ, when updating the hidden state of u_i, we use the updated hidden state to calculate the attention weight.

Finally, a relational graph attention networks propagation module updates the representation of u_i by aggregating representations of its neighborhood $N(i)$, and an attention mechanism is used to attend to the neighborhood's representations. We define the propagation module as follows:

[3] The hidden state of utterance in layer l is equivalent to the representation of utterance in layer l.

$$h_i^l = \left(\sum_r \sum_{\tau \in N^r(i)} \alpha_{i,\tau}^l W_r^l h_\tau^l \right) + \alpha_{i,i}^l W_0^l h_i^{l-1} \qquad r \in 0,1 \qquad j <= \tau <= u-1 \tag{9}$$

where $N^r(i)$ donates the neighborhood of u_i under the edge type r.

In each layer, TRGAT can adaptively gather context information for target utterance from both the neighboring utterances and the remote utterances because of the following reason: the target utterance can directly interact with the previous utterances in the context window through directed relational edges. And each utterance in context window has gathered context information for itself, so the target utterance can indirectly attend to the remote utterances.

Let's take the conversation in Fig. 1 as an example to illustrate the update process of utterance representation. The dialogue consists of six utterances $\{u_1,u_2,u_3,u_4,u_5,u_6\}$, u_1, u_3, u_6 are spoken by s_1, u_2, u_4, u_5 are spoken by s_2. The historical context for each utterance is shown in Table 1, and the update process of utterance representation in the l-th TRGAT layer is shown in Fig. 2.

Table 1. The utterances and its historical context in conversation.

Utterance	Historical context
u_1	$\{\emptyset\}$
u_2	$\{u_1\}$
u_3	$\{u_1, u_2\}$
u_4	$\{u_2, u_3\}$
u_5	$\{u_4\}$
u_6	$\{u_2,u_3,u_4\}$

Emotion Classification. After obtaining the representations h_i^L of each utterance node through stacking TRGAT layer of L layers, we concatenate the non-contextual representation h_i^0 and the representation h_i^L as the final representation of u_i, and pass it through a feed-forward neural network and a softmax layer to get the emotion distribution:

$$H_i = h_i^0 || h_i^L \tag{10}$$

$$Z_i = ReLu(W_H H_i + b_H) \tag{11}$$

$$P_i = Softmax(W_Z Z_i + b_Z) \tag{12}$$

where W_H and W_Z denote learnable weight matrixes, and b_H and b_Z denote learnable bias vectors.

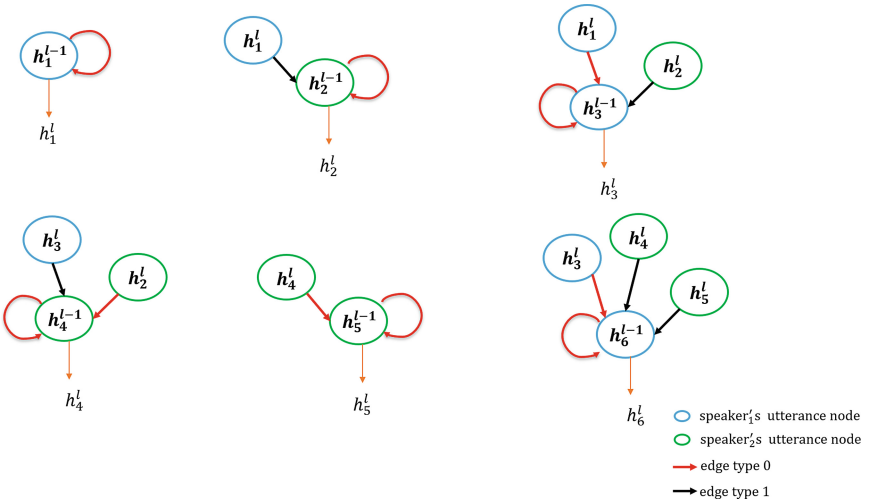


Fig. 2. Each utterance updates the hidden state according to its temporal position in the dialogue. Each subgraph represents the computational graph of the currently updated node(utterance). h_i^{l-1},h_i^l represents the hidden state of i-th utterance in layer $l-1$ and l respectively. The two speakers' utterances are colored blue and green respectively. The edges represent the direction of the flow of information. The utterance represented by the source node and the tail node of the red arrow is said by the same speaker, which is used to model self-dependency between utterances, and the utterance represented by the source node and the tail node of the black arrow is spoken by different speakers, used to model inter-speaker dependency between utterances. (Color figure online)

4 Experiment

4.1 Datasets and Evaluation Metrics

We evaluate our model on two benchmark datasets: IEMOCAP [1] and MELD [11]. Both datasets are multimodal datasets containing textual, visual, and acoustic information for every utterance of each conversation. In this work, we focus on conversational emotion recognition only from textual information. We leave multimodal dialogue emotion recognition as future work, and when comparing model performance, we also only use the performance of different models in text modalities.

The IEMOCAP dataset contains videos of dyadic conversations where actors perform improvisations or scripted scenarios. Each conversation is segmented into utterances, which are annotated with one of the six emotion labels: happy, sad, neutral, angry, excited, and frustrated.

The MELD dataset comes from the Friends TV series with multiple speakers involved in the conversations. The utterances are annotated with one of seven labels: neutral, happiness, surprise, sadness, anger, disgust, and fear.

The statistics of the two datasets are shown in Table 2. Because IEMOCAP has no validation set, we extract the validation set from the randomly shuffled

training set with the ratio of 8:2. Following [2,9], we use the F1-score to evaluate the performance for each emotion class, and use the weighted F1-score to evaluate the overall performance on the two datasets.

Table 2. Statistics of IEMOCAP, MELD

Dataset	# conversations			Avg. conversation len			# utterance		
	Train	Val	Test	Train	Val	Test	Train	Val	est
IEMOCAP	120		31	48		52	5810		1623
MELD	1038	114	280	10	10	9	9989	1109	2610

4.2 Baselines

For a comprehensive performance evaluation, we compared our model with the following baselines:

CNN: As described in Sect. 3.2, it is our utterance representation extractor and trained at the utterance-level without contextual information. **scLSTM** [10]: It captures contextual information from historical utterances by using a unidirectional LSTM. **Memnet** [13]: The current utterance is fed to a memory network, where the memories correspond to historical utterances. The output from the memory network is used as the final utterance representation for emotion classification. **DialogueRNN** [9]: It is a recurrent network that uses two GRUs to track individual speaker's states and global context during the conversation. Further, another GRU is employed to track emotional state through the conversation. **DialogueGCN** [2]: It captures self-dependency and inter-speaker dependency between utterances by using two-layer graph neural networks. For a fair comparison, we remove the directed edges from future utterances to current utterances from the original graph structure to avoid backpropagation of dialogue information. **DialogueRGAT** [5]: Based on DialogueGCN and taking the sequential information of conversation into account, DialogueRGAT propose a kind of relational position encodings that provide RGAT with sequential information. Our handling of graph structures is consistent with DialogueGCN.

4.3 Implementation Settings

We use the following settings to optimize the model parameters during training: the dimension of initial utterance representation is set to 100, 600 for IMEOCAP and MELD respectively. In each TRGAT layer, the size of hidden states is the same as the utterance representation dimension. To prevent our model from overfitting, we adopted drop out after each TRGAT layer and the dropout rate is 0.4. We employed AdamW as the optimizer for model learning and the learning rate is 0.0005. We used the standard cross-entropy loss as the loss function to train the model. On both datasets, we train 100 epochs on the training set and the

batch size is 32, saving the model parameters with the best overall performance on the validation set, and finally report the performance on the test set.

For the TRGAT layer size L, we let $L = 3$ for the overall performance comparison by default, but we also carried out experiments with different layer size in Sect. 4.5 to explore how it influence the overall performance.

4.4 Experimental Results

Tabel 3 and Table 4 present the results of IEMOCAP and MELD testing sets, respectively.

Table 3. Performance comparison on the IEMOCAP dataset. The evaluation metrics is F1 for each class. Average(w) = Weighted F1, † denotes results refer to the original paper. * denotes the re-implement results.

Models	Emotion classes						Average (w)
	Happy	Sad	Neutral	Angry	Excited	Frustrated	
CNN	29.9	53.8	40.1	52.4	50.1	55.8	48.2
scLSTM†	34.4	60.9	51.8	56.7	57.9	58.9	54.9
Memnet†	33.5	61.8	52.8	55.4	58.3	59.0	55.1
DialogueRNN†	35.5	69.9	55.3	61.9	62.2	59.4	58.8
DialogueGCN*	36.2	74.1	**56.2**	63.9	62.0	61.7	60.2
DialogueRGAT*	37.1	72.4	56.0	65.8	**62.4**	60.4	60.7
Ours	**39.1**	**75.8**	55.1	**67.2**	61.2	**61.7**	**62.6**

Table 4. Performance comparison on the MELD dataset.

Models	Emotion Classes							Average (w)
	Neutral	Suprise	Fear	Sad	Joy	Disgust	Anger	
CNN	74.9	45.5	3.7	21.1	49.4	8.2	34.5	55.0
scLSTM†	73.8	47.7	5.4	25.1	51.3	5.2	38.4	55.9
Memnet†	72.8	49.4	8.8	24.6	48.3	3.1	42.3	55.6
DialogueRNN†	73.5	49.4	1.2	23.8	50.7	1.7	41.5	55.9
DialogueGCN*	73.1	50.1	8.8	**26.2**	50.3	6.3	39.5	55.7
DialogueRGAT*	74.6	**52.3**	7.2	24.5	**51.5**	7.1	40.9	56.1
Ours	**75.5**	50.2	**10.4**	25.9	51.1	**9.2**	**42.6**	**57.8**

IEMOCAP: In Table 3, our model performs better than all compared models on IMOCAP dataset. Our model attains the best overall performance with improvement over the strongest RNN-based baseline DialogueRNN (+3.8% weight-f1) and the strongest graph-based baseline DialogueRGAT(+1.9% weight-f1).

From the experiment results, the graph-based models(DialogueGCN, DialogueRGAT) perform better than the RNN-based model (DialogueRNN). Perhaps DialogueRNN employs gated recurrent unit (GRUs) to model conversational context, GRUs-based modeling methods can be problematic for many long conversations in IEMOCAP dataset. In contrast, DialogueGCN and DialogueRGAT try to overcome this issue by constructing relational edges to directly model the correlation between utterances. Our model acts like a combination of RNN-based and graph-based models and can better model conversational context.

MELD: For the conversations in MELD dataset, it contains an average of 10 utterances and many conversations containing more than 5 speakers. So this makes the interaction between speakers more difficult than IEMOCAP which only consists of dyadic conversations. So under this circumstance, graph-based models' advantage in encoding context is not that important. So we found that the difference in results between RNN-base models and graph-based models is not as contrasting as it is in the case of IEMOCAP. The overall performance is not significantly different.

But our models still outperform all baseline methods that suggest the efficacy of our context-modeling method. Compared with the best baseline model DialogueRGAT, our model attains +1.7% weight-f1 improvement in overall performance. In addition, our models perform the best on the two minority classes fear and disgust, this demonstrates the capability of our models in recognizing minority emotion classes.

4.5 Model Analysis

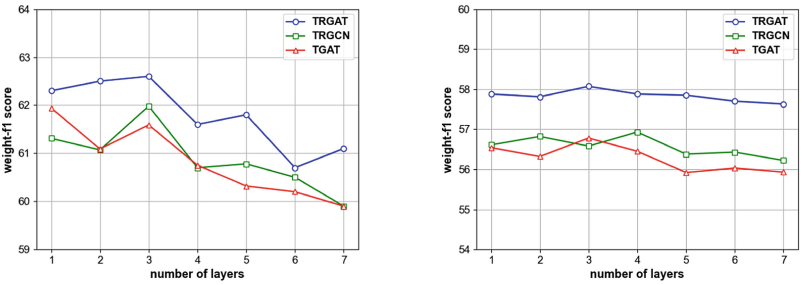
Pre-trained Models as Utterance Feature Extractor. With the outstanding performance of pre-trained models in natural language understanding tasks, pre-trained models are often used as utterance feature extractor in recent works. We replace the CNN-based extractor described in Sect. 3.2 with the Roberta-based extractor to demonstrate the effectiveness of our method regardless of what utterance feature extractor is used. The experimental results are shown in Table 5. From the results, all models can gain remarkable improvement by employing the powerful extractor. Our method attains comparable results compared with the state-of-the-art model DAG+Roberta [14] on IEMOCAP dataset. Meanwhile, our model also achieves comparable results with the best baseline models on the MELD dataset.

Number of TRGAT Layers. We further explore the relationship between model performance and number of TRGAT layer, and whether using RGAT's message aggregation principle to aggregate contextual information for each utterance outperforms other graph networks? Here, we use the message aggregation principle used in Graph Attention Network (GAT) [16] and Relational Graph Convolutional Network (RGCN) [13] as a comparative experiment. we denoted the two layers as TGAT and TRGCN. As shown in Fig. 3, we set different

Table 5. Performance comparison of different models using roberta as feature extractor on IEMOCAP and MELD datasets.

Models	IEMOCAP	MELD
Roberta	63.4	62.9
DialogueRNN+Roberta	64.8	**63.6**
DialogueGCN+Roberta	64.9	63.0
DialogueRGAT+Roberta	66.4	62.9
DAG+Roberta	**68.0**	**63.6**
Ours+Roberta	67.9	63.3

TRGAT layers on IEMOCAP and MELD datasets to compare the performance with TGAT and TRGCN.



(a) IEMOCAP dataset (b) MELD dataset

Fig. 3. Test results of TRGAT, TRCN, and TGAT on the IEMOCAP dataset and MELD dataset by different numbers of network layers.

For static graph neural network (GNN) based models such as DialogueGCN and DialogueRGAT, the only way to receive information from remote utterances for an utterance is to stack several GNN layers. However, in our model, at every layer of TRGAT, we can gather remote utterance information indirectly for each utterance by considering the timing of aggregated information. So rather than stacking many TRGAT layers, we can attain competitive performance with few layers on both datasets. Meanwhile, when stacking more TRGAT layers on the IEMOCAP dataset, the model suffers from performance degradation, which is not obvious on the MELD dataset. We believe when the number of TRGAT layers increases, the number of parameters of the model also increases, the IMEOCAP dataset is relatively small and over-fitting occurs. And RGAT's message aggregation principle perform better than GAT and RGCN. Compared with RGAT, GAT's message aggregation principle don't take the relation of the edge into consideration, so it don't model the self-dependency and inter-speaker dependency when gather historical context information for

the utterance. Compared with RGCN, RGAT can more flexibly determine the importance of historical utterances to current utterances through an attention mechanism.

5 Conclusion

In this paper, we propose a temporal and relational graph attention network, named DialogueTRGAT, for emotion recognition in conversation. DialogueTRGAT gathers context information for each utterance based on their temporal position in dialogue and uses the message aggregation principle of relational graph attention networks (RGAT) to aggregate historical context information for each utterance. So it acts like a combination of the RNN-based model and graph-based model. We think it is a more effective way to model the information flow within conversations and can gains more meaningful context cues for each utterance for better emotion recognition. Extensive experiments were conducted and compared with previously proposed methods, our resulting model is more competitive.

Acknowledgments. The authors would like to thank the anonymous reviewers for the helpful comments. This work was supported by Projects 61876118 under the National Natural Science Foundation of China, the National Key RD Program of China under Grant No.2020AAA0108600 and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1. Busso, C., et al.: Iemocap: interactive emotional dyadic motion capture database. Lang. Resour. Eval. **42**(4), 335–359 (2008)
2. Ghosal, D., Majumder, N., Poria, S., Chhaya, N., Gelbukh, A.: Dialoguegcn: a graph convolutional neural network for emotion recognition in conversation. arXiv preprint arXiv:1908.11540 (2019)
3. Hazarika, D., Poria, S., Mihalcea, R., Cambria, E., Zimmermann, R.: ICON: interactive conversational memory network for multimodal emotion detection. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2594–2604. Association for Computational Linguistics, Brussels, Belgium, Oct-Nov 2018. https://doi.org/10.18653/v1/D18-1280, https://aclanthology.org/D18-1280
4. Hu, D., Wei, L., Huai, X.: Dialoguecrn: contextual reasoning networks for emotion recognition in conversations. arXiv preprint arXiv:2106.01978 (2021)
5. Ishiwatari, T., Yasuda, Y., Miyazaki, T., Goto, J.: Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7360–7370 (2020)
6. Jiao, W., Yang, H., King, I., Lyu, M.R.: Higru: hierarchical gated recurrent units for utterance-level emotion recognition. arXiv preprint arXiv:1904.04446 (2019)
7. Kumar, A., Dogra, P., Dabas, V.: Emotion analysis of twitter using opinion mining. In: 2015 Eighth International Conference on Contemporary Computing (IC3), pp. 285–290. IEEE (2015)

8. Majumder, N.: Mime: mimicking emotions for empathetic response generation. arXiv preprint arXiv:2010.01454 (2020)
9. Majumder, N., Poria, S., Hazarika, D., Mihalcea, R., Gelbukh, A., Cambria, E.: Dialoguernn: an attentive rnn for emotion detection in conversations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6818–6825 (2019)
10. Poria, S., Cambria, E., Hazarika, D., Majumder, N., Zadeh, A., Morency, L.P.: Context-dependent sentiment analysis in user-generated videos. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 873–883 (2017)
11. Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., Mihalcea, R.: Meld: a multimodal multi-party dataset for emotion recognition in conversations. arXiv preprint arXiv:1810.02508 (2018)
12. Shen, W., Wu, S., Yang, Y., Quan, X.: Directed acyclic graph network for conversational emotion recognition. arXiv preprint arXiv:2105.12907 (2021)
13. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
14. Thost, V., Chen, J.: Directed acyclic graph neural networks. arXiv preprint arXiv:2101.07965 (2021)
15. Zhang, D., Wu, L., Sun, C., Li, S., Zhu, Q., Zhou, G.: Modeling both context-and speaker-sensitive dependence for emotion detection in multi-speaker conversations. In: IJCAI, pp. 5415–5421 (2019)
16. Zhou, L., Gao, J., Li, D., Shum, H.Y.: The design and implementation of xiaoice, an empathetic social chatbot. Comput. Linguist. **46**(1), 53–93 (2020)