Metadata of the chapter that will be visualized in SpringerLink

Book Title	Artificial Intelligence	
Series Title		
Chapter Title	Developing Relationships: A Heterogeneous Graph Network with Learnable Edge Representation for Emotion Identification in Conversations	
Copyright Year	2023	
Copyright HolderName	The Author(s), under exclusive license to Springer Nature Switzerland AG	
Author	Family Name	**Li**
	Particle	
	Given Name	**Zhenyu**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
	Address	Shenzhen, China
	Division	
	Organization	Joint Lab of HIT-KONKA
	Address	Shenzhen, China
	Email	190110709@stu.hit.edu.cn
Author	Family Name	**Tu**
	Particle	
	Given Name	**Geng**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
	Address	Shenzhen, China
	Division	
	Organization	Joint Lab of HIT-KONKA
	Address	Shenzhen, China
	Email	tugeng0313@gmail.com
Corresponding Author	Family Name	**Liang**
	Particle	
	Given Name	**Xingwei**
	Prefix	
	Suffix	
	Role	
	Division	

	Organization	Konka Research Institute
	Address	Shenzhen, China
	Division	
	Organization	Joint Lab of HIT-KONKA
	Address	Shenzhen, China
	Email	liangxingwei@konka.com
Corresponding Author	Family Name	**Xu**
	Particle	
	Given Name	**Ruifeng**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
	Address	Shenzhen, China
	Division	
	Organization	Joint Lab of HIT-KONKA
	Address	Shenzhen, China
	Email	xuruifeng@hit.edu.cn

Abstract	Emotion recognition in conversations (ERC) aims to predict the emotion of utterances. Modeling context dependencies is the critical challenge of the task. Existing efforts in ERC are mainly based on the sequence and graph models. The graph models can better capture structured information than the sequence models. Unfortunately, there are few suitable aggregation strategies for ERC models based on high-dimensional edge features. Moreover, the adjustment of edge representation in graph-based models has been ignored for a long time. Based on this, we propose a learnable edge message-passing model based on a heterogeneous dialog graph. The model first calculates the attention weights between utterance nodes and between nodes and edges separately and then learns contextual utterance representations through these learnable edge representations. Additionally, we conducted our experiment on four public datasets and achieved advanced results.

Keywords (separated by '-')	Conversational emotion identification - Graph transformer - Learnable edge

Developing Relationships: A Heterogeneous Graph Network with Learnable Edge Representation for Emotion Identification in Conversations

Zhenyu Li[1,3], Geng Tu[1,3], Xingwei Liang[2,3(✉)], and Ruifeng Xu[1,3(✉)]

[1] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China
190110709@stu.hit.edu.cn
[2] Konka Research Institute, Shenzhen, China
liangxingwei@konka.com
[3] Joint Lab of HIT-KONKA, Shenzhen, China
xuruifeng@hit.edu.cn

Abstract. Emotion recognition in conversations (ERC) aims to predict the emotion of utterances. Modeling context dependencies is the critical challenge of the task. Existing efforts in ERC are mainly based on the sequence and graph models. The graph models can better capture structured information than the sequence models. Unfortunately, there are few suitable aggregation strategies for ERC models based on high-dimensional edge features. Moreover, the adjustment of edge representation in graph-based models has been ignored for a long time. Based on this, we propose a learnable edge message-passing model based on a heterogeneous dialog graph. The model first calculates the attention weights between utterance nodes and between nodes and edges separately and then learns contextual utterance representations through these learnable edge representations. Additionally, we conducted our experiment on four public datasets and achieved advanced results.

Keywords: Conversational emotion identification · Graph transformer · Learnable edge

1 Introduction

Conversation is the most common and effective way to convey information and emotion between people. With the rapid development of the Internet, social media platforms such as YouTube and Twitter have generated massive comment and conversation data, attracting more and more researchers to participate in data mining and emotion recognition [5,9]. Emotion recognition in conversations (ERC) can not only be helpful in chatbots to achieve emotional responses driven by user emotions [1,16] but also contributes to recommendation systems [17].

L. Fang et al. (Eds.): CICAI 2022, LNAI 13606, pp. 1–13, 2022.
https://doi.org/10.1007/978-3-031-20503-3_25

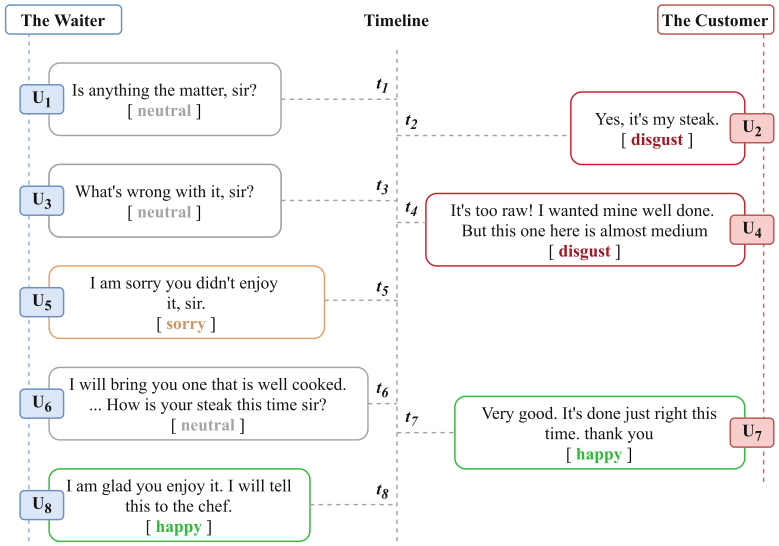

Fig. 1. An abridged dialogue between the waiter and the customer in a restaurant from DailyDialog [11]. There are two speakers: the waiter and the customer. Each utterance at time t_i is denoted by U_i. Emotion labels are marked in the form of *"[happy]."*

Compared with vanilla sentiment analysis, modeling the context in conversations is the key challenge in ERC. Fig. 1 illustrates an example of emotion recognition in a dialogue. When the emotion analysis is performed out of context, the short utterance *"Yes, it's my steak"* often obtains neutral results. Only by combining U_2 and U_4 can we know the emotional tendency of this utterance.

Existing work in ERC can be divided into two categories by context modeling: sequence-based and graph-based models. *Sequence models* [6,13,14] often use LSTMs or GRUs to extract dialog-level and speaker-level information. *Graph models* [5,7,19] pay attention to the correlation between utterances, and can model the complex contextual structure between dialog-level and speaker-level by making various associations between utterances.

Utterances information aggregation algorithm is as important as a sufficiently reasonable dialog graph modeling. The graph models above often use attention weights as edge features, and use methods such as Graph Attention Network (GAT) [22] or modified versions to aggregate information between utterances nodes weighted. However, models like SKAIG [9] use more complex common sense edge features, which is difficult for simple GAT to use complex edge features to generate effective attention weights and aggregate context information. Although the Unified Graph Transformer (UGT) [20] can allow edge features to participate in attention calculation, it still cannot allow edge features to be dynamically learned, which makes context modeling solidified. CensNet [8], NENN [24], and EGAT [23] propose to regard edges as nodes and update edges with neighbor edges and neighbor nodes. These methods need to construct edge-

to-edge subgraphs based on the original graph, which mixes the features of different edges since neighboring edges cannot be guaranteed to be similar.

To solve the above problems, we propose a new graph transformer based on the Transformer [21], Learnable Edge Message Passing Network (LEMPN), and build an abstract dialogue graph framework adapted to the model for subsequent applications. Specifically, We first use RoBERTa [12] to encode utterances in conversations. Second, we define a simple and abstract dialog graph model, which treats all utterances as nodes, and regards any edge relationships between sentences as high-dimensional learnable features. Then, we propose an attention mechanism for sentence nodes and adjacent edges, respectively. Finally, we test our model on four high-frequency used datasets: IEMOCAP [3], DailyDialog [11], EmoryNLP [25], and MELD [15]. We achieve impressive results on DailyDialog and are competitive with the baseline on MELD and EmoryNLP.

2 Related Work

Emotion recognition in conversation is still a research hotspot, arousing broad interest among NLP researchers. There are many works in context modeling and can be divided into sequence-based and graph-based models.

2.1 Sequence-based Models

Sequence models [6,13,14] often use LSTM or GRU to extract sequential context information. ICON [6] extracts the speaker-level context representations of each speaker from the dialog, then use these representations to build dialog-level context, and stores the dialog-level features into the multi-hop memory. DialogueRNN [13] uses GRU to model the dialog-level context first and then uses attention to model the speaker-level context of the listener and speaker. COSMIC [4] makes improvements based on DialogueRNN and introduces external common sense knowledge to improve model performance. From the message passing view, sequence methods are limited to updating utterances stated only with the adjacent utterances. It is difficult for them to extract complex associations between utterances, so many graph-based models have been proposed.

2.2 Graph-based Models

Graph models [5,7,19] pay attention to the correlation between utterances and can model the complex contextual relationship between dialog-level and speaker-level at a deeper level by making various associations between utterances and utterances. For example, the DialogueGCN [5] builds a directed graph based on dialogue, uses edges representing relationships to connect related utterances, and allows the edges to carry the attention scores between utterances and weighted aggregate. RGAT [7] adds position encoding on the basis of DialogueGCN. For DAG-ERC [19], a directed acyclic graph is established in terms of timing and information transmission. SKAIG [9] establishes utterances associations based

on psychological common sense implicit in dialogues, and uses a Unified Graph Transformer (UGT) [20] to aggregate information. From the message passing view, these models build a directed graph of conversations, use edges to represent the direction of message delivery, additional relation or type information, and aggregates based on computable attention edge weights using methods like GAT [22] or modified version. The edge features of SKAIG [9] have semantics and are more informative than the simple edge weights of other models. Its edge features are common sense from COMET [2], which implies psychological state change momentum.

Simple weighted aggregation cannot fully use edge features, so we propose a Learnable Edge Message Passing Network (LEMPN) based on a graph transformer, which innovatively calculates the attention weights from nodes to edges and makes edge features adaptive through learnable edge representations. As far as we know, our work is the first application of an edge learnable graph transformer in emotion identification in conversations.

3 Methodology

3.1 Problem Definition

A conversation is defined as a sequence of utterances $\{u_1, u_2, ..., u_N\}$, where N is the number of utterances in conversation. Each utterance u_i contains L_i words as $\{w_1, w_2, ..., w_{L_i}\}$. Additionally, a conversation have P people involved, and the utterance u_i expressed by its corresponding speaker $S_j \in \{S_1, S_2, ..., S_P\}$. The objective of emotion identification in conversation is to classify all utterances in a conversation to their correct emotion label $\{E_1, E_2, ..., E_M\}$, where M is the number of the labels.

3.2 Utterance Independent Encoder

RoBERTa [12] is widely used in ERC for extracting utterances features. We feed utterance $u_i = \{w_{i,1}, w_{i,2}, ..., w_{i,L_i}\}$ into RoBERTa, and obtain the hidden states of the last layer.

$$F_i = RoBERTa(w_{i,1}, ..., w_{i,L_i}) \tag{1}$$

where $F_i \in \mathbb{R}^{L_i \times d_w}$. d_w is the dimension of the hidden states of each word.

High-dimensional features will introduce more parameters, which is not conducive to feature expression. Therefore we introduce a max-pooling and linear projection operation to reduce the dimension following Li et al. [9]:

$$f_i = Linear(MaxPooling(F_i)) \tag{2}$$

where $f_i \in \mathbb{R}^{d_h}$ is the representation of the utterance, and d_h is the dimension of the representation. After all utterances are encoded, we obtain the context-independent conversation representation of conversation $C \in \mathbb{R}^{N \times d_h}$.

3.3 Dialogue Graph Modeling

In this section, we define a dialog abstract directed graph framework with edge features for introducing our message passing model below. Generally, a dialogue graph can be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $v_i \in \mathcal{V}$ is an utterance node. $e_{ij} \in \mathcal{E}$ is an directed edge from node v_i to v_j.

Vertices. The whole conversation has been encoded as C and we set the initial node feature of v_i as $h_i = f_i \in C$. The node will aggregate the contextual information from its adjacent nodes and edges, then update its features.

Edges. The meaning of a directed edge e_{ij} is to express the temporal, semantics, speaker, and other contextual relations between node v_i and v_j, and to indicate the direction of information transmission. The rules of edge selection between nodes depend on the specific implementation. For acyclic graphs or cyclic graphs, fully connected graphs, or graphs with limited window size, message passing can always be used to aggregate node information.

Edges can express different types of contextual relationships through their features through relative position encoding, relation embedding, etc. But recent works often ignore edges themselves can carry meaningful information at higher dimensions. Therefore, we propose the following model to fill in the field gap.

3.4 Learnable Edge Message Passing Network

Learnable Edge Message Passing Network, namely LEMPN, uses separate attention to node and edge to make better use of edge features and enable self-learning of edge features based on the former abstract graph.

Graph Transformer. Transformer [21] has proven to be very effective in NLP, and we extend its multi-head attention mechanism to node-to-node and node-to-edge in pair drawing. Specially, in layer l, given node features of an conversation $H = \{h_1, h_2, ...h_N\}$, we construct multi-head attention function from v_i to v_j as following:

$$f_{q,t}(h_i) = W_{q,t}h_i + b_{q,t} \tag{3}$$

$$f_{k,t}(h_j) = W_{k,t}h_j + b_{k,t} \tag{4}$$

$$\alpha_{ij,t} = Softmax\left(\frac{\langle f_{q,t}(h_i),\ f_{k,t}(h_j)\rangle}{\sum_{n\in\mathcal{N}(i)}\langle f_{q,t}(h_i),\ f_{k,t}(h_n)\rangle}\right) \tag{5}$$

where $f_{q,t}(h_i) \in \mathbb{R}^{d_h}$ and $f_{k,t}(h_j) \in \mathbb{R}^{d_h}$, and d_h is the dimension of node feature. $W_{q,t}, W_{k,t}, b_{q,t}, b_{k,t}$ is trainable weights and bias, and $\langle a, b\rangle = \frac{a\cdot b^\top}{\sqrt{d_t}}$, where d_t is the dimension of each head. For the t-th head attention, we use $f_{q,t}(h_i)$ and $f_{k,t}(h_j)$ to transform v_i feature h_i and v_j feature h_j into query vector and key vector. In this way, the information weight of each adjoining node during

message transmission is determined. Compared with directly processing edge features with inter-node attention weight representation or letting edge features directly participate in attention calculation, we consider that high-dimensional edge features may have semantic meanings and should be treated separately. Similar to the above, we compute the node-to-edge attention score in the same way as follows:

$$g_{k,t}(e_{ji}) = M_{k,t}e_{ji} + p_{k,t} \tag{6}$$

where $g_{k,t}(e_{ji}) \in \mathbb{R}^{d_e}$, and d_e is the dimension of edge features. $M_{k,t}, p_{k,t}$ are also trainable weight and bias. e_{ji} is a directed edge from v_j to v_i. For the t-th head attention, the key vector of edge e_{ij} is $g_{k,t}(e_{ji})$. Then, we calculate the attention weight of the v_i to all the e_{ji} pointing to it as $\beta_{ij,t}$ by using the edge feature, which supplements the information weight of each adjacent edge when the message is passed.

$$\beta_{ij,t} = Softmax\left(\frac{\left\langle f_{q,t}(h_i),\ g_{k,t}(e_{ji})\right\rangle}{\sum_{n\in\mathcal{N}(i)}\left\langle f_{q,t}(h_i),\ g_{k,t}(e_{ni})\right\rangle}\right) \tag{7}$$

Message Passing. The so-called *message passing* means that after calculating the attention weight, the information of adjacent nodes and adjacent edges is aggregated and the node feature is updated. We first compute the value matrix for nodes and edges. Then we aggregate the contextual information of adjacent nodes and edges separately, and simply add them together to construct message msg_i to v_i as follows:

$$f_{v,t}(h_j) = W_{v,t}h_j + b_{v,t} \tag{8}$$

$$g_{v,t}(e_{ji}) = M_{v,t}e_{ji} + p_{v,t} \tag{9}$$

$$msg_i = \sum_{n\in\mathcal{N}(i)} \mathrm{Concat}_{\mathrm{t}}\left(\alpha_{in,t}f_{v,t}(h_n)\right) + \sum_{n\in\mathcal{N}(i)} \mathrm{Concat}_{\mathrm{t}}\left(\beta_{in,t}g_{v,t}(e_{ni})\right) \tag{10}$$

where $f_{v,t}(h_j) \in \mathbb{R}^{d_h}$ and $g_{v,t}(e_{ji}) \in \mathbb{R}^{d_e}$. $\mathrm{Concat}_{\mathrm{t}}(\cdot)$ concats all heads. When $d_h \neq d_e$, a projection for edge features is additionally needed.

Node Update. In this part, we use a gated residual connection between layers inspired by UGT [20] to prevent our model from over-smoothing. In layer l, we calculate the v_i new feature as follows.

$$\gamma_i = Sigmoid(W_1[msg_i, o_i, msg_i - o_i]) \tag{11}$$

$$\hat{h}_i = (1-\gamma_i)msg_i + \gamma_i o_i \tag{12}$$

where o_i is an trainable linear projection of h_i.

Edge Update. From the perspective of an edge, the context of the edge e_{ji} comes from v_i and v_j. So the edge features need to be adaptive according to the changes of the nodes at both ends.

$$f_{qk}(h_i, h_j) = \frac{f_{q,t}(h_i) \circ f_{k,t}(h_j)}{\sqrt{d_t}} \tag{13}$$

$$g_{qk}(h_i, e_{ji}) = \frac{f_{q,t}(h_i) \circ g_{k,t}(e_{ji})}{\sqrt{d_t}} \tag{14}$$

$$\hat{e}_{ji} = e_{ji} + W_e[g_{v,t}(e_{ji}) f_{qk}(h_i, h_j);\ g_{v,t}(e_{ji}) g_{qk}(h_i, e_{ji})] + b_e \tag{15}$$

where $f_{qk}(h_i, h_j) \in \mathbb{R}^{d_h}$, and $g_{qk}(h_i, e_{ji}) \in \mathbb{R}^{d_e}$. $a \circ b$ is Hadamard product, namely element-wise multiply. W_e, b_e are trainable. This formula weights the point-to-point and point-to-edge aspects of edge features separately through the query and key results of adjacent points and edges, and finally aggregates them linearly.

Layer Passing. We learn point-wise feed forward network (FFN) from Transformer [21] to update node features and edge features respectively to enhance the nonlinear ability of the model. The node feature h_i and e_{ji} on layer l will be updated to h_i^+ and e_{ji}^+ as follows:

$$h_i^+ = LayerNorm(\hat{h}_i + Linear_1(ReLU(Linear_2(\hat{h}_i)))) \tag{16}$$

$$e_{ji}^+ = LayerNorm(\hat{e}_{ji} + Linear_3(ReLU(Linear_4(\hat{e}_{ji})))) \tag{17}$$

3.5 Emotion Classifier

After sufficient feature extraction by the upstream model, we use the following linear units for classification.

$$Z = Softmax(W_z H + b_z) \tag{18}$$

where H is the conversation feature matrix of the graph last layer, and $W_z \in \mathbb{R}^{M \times d_h}$, $b_z \in \mathbb{R}^M$ are trainable. The cross-entropy loss is utilized to train the model base on conversation as follows:

$$loss = -\frac{1}{N} \sum_{i=1}^{N} \sum_{e=1}^{M} z_i log(Z_i) \tag{19}$$

where z_i is the one-hot vector of utterance true emotion label.

4 Experimental Settings

4.1 Datasets

We evaluate our LEMPN on four widely used conversational emotion identification datasets. The statistics of them are shown in Table 1.

IEMOCP. [3] is a dataset of two-person conversations, which was recorded from ten actors in dyadic sessions. Each utterance in it is annotated with an emotion label as *neutral, happy, sad, angry, excited, frustrated*. The dataset is not divided into training and validation sets, so we divide it the same as Li et al. [9].

DailyDialog. [11] is a two-way dialogue in various scenarios, including 7 emotions: *neutral, happiness, sadness, anger, surprise, disgust, fear*. 83% of the data is marked as neutral.

EmoryNLP. [25] is taken from the TV series *Friends* and contains multiple conversations. Each sentence is marked with one of the *neutral, mad, sad, scared, powerful, peaceful, joyful* emotional labels.

MELD. [15] is also collected from *Friends*, and its emotional label is consistent with DailyDialog.

Table 1. Statistics of IEMOCAP, DailyDialog, MELD, EmoryNLP

Dataset	#Conversations			#Utterances			#Metrics
	Train	Dev	Test	Train	Dev	Test	
IEMOCAP	120		31	5810		1623	Weighted-F1
DailyDialog	11118	1000	1000	87170	8069	7740	Micro-F1 & Macro F1
EmoryNLP	659	89	79	7551	954	984	Weighted-F1
MELD	1039	114	280	9989	1109	2610	Weighted-F1

4.2 Compared Methods

We compared our model with the following methods and baselines.

Sequence-based Models. ICON [13], DialogueRNN [13], Dialogue-RNN with RoBERTa which is also implemented by Li et al. [4], HiTrans [10], DialogXL [18] and COSMIC [4].

Graph-based Models. DialogueGCN [5], RGAT [7], DAG-ERC [19], and the SKAIG [9] with UGT [20] using the same train settings to control variables.

Baselines. We also compared our work with RoBERTa [12] followed by a simple classifier and RoBERTa-Transformer, which replaces the graph transformer with transformer. They are implemented by Li et al. [4].

5 Results and Discussions

5.1 Overall Results

Overall experiment results are illustrated in Table 2. We can notice that on DailyDialog, our model achieves out-performed results, higher than the second-placed SKAIG with UGT +1.62 (2.7%) and +0.39 (0.7%) on Micro-F1 and Macro-F1, respectively. As Table 1 shows, DailyDialog has 8 to 15 times as many sentences as other datasets. Our model separately computes attention for nodes and edges while additionally learning edge features with a linear layer, which increases the parameters of the model. In this way, our model can learn better from the dataset.

On MELD, our model still performs well. Weighted F1 is +0.15 (0.2%) higher than the second-placed COSMIC model. MELD has the same sentiment label as DailyDialog, but with fewer data. The best graph-based model (SKAIG + UGT) has lower results than the best sequence-based model (COSMIC), according to Table 2. This is because MELD comes from episodes, frequent multi-party conversations, and discontinuous sampling problems, making the graph model structure on it broken and performing poorly. Our model fixes discontinuities to some extent by adaptable edge weights. But still, small datasets make our model less effective. EmoryNLP also comes from the same series as MELD, but with different tags and fewer data. Our model's Weighted F1 is slightly higher than SKAIG with UGT but lower than DAG-ERC by about -0.1 (0.3%) because its overall data size is smaller than MELD.

Table 2. Results of our method and other baselines

Methods	IEMOCAP	DailyDialog		EmoryNLP	MELD
	Weighted-F1	Micro-F1	Macro-F1	Weighted-F1	Weighted-F1
RoBERTa	55.67	55.16	48.20	37.00	62.75
+ Transformer	63.78	58.28	47.00	37.50	64.59
ICON	63.50	–	–	–	–
DialogueRNN	62.57	55.95	41.80	31.70	57.03
+ RoBERTa	64.76	57.32	49.65	37.44	63.61
HiTrans	64.50	–	–	36.75	61.94
DialogXL	65.94	54.93	–	34.73	62.41
COSMIC	65.28	58.48	51.05	38.11	65.21
DialogueGCN	64.18	–	–	–	58.10
RGAT	65.22	54.31	–	34.42	60.91
DAG-ERC	**68.03**	59.33	–	**39.02**	63.65
SKAIG + UGT	66.96	59.75	51.95	38.88	65.18
SKAIG + LEMPN	66.14	**61.37**	**52.34**	**38.91**	**65.36**

Our model doesn't perform well on IEMOCAP, with -0.82 (1.2%) lower than the baseline model SKAIG with UGT and -1.89 (2.8%) lower than the best method DAG-ERC. Considering that the amount of data in IEMOCAP is the least due to a large number of parameters, our model has less generalization ability on small datasets. Moreover, the dialogue length is much larger than other dialogues, which makes IEMOCAP rich in contextual information. Our learnable edge features can supplement contextual information in small dialogues with less contextual information but may do the opposite in long dialogues. We conducted a detailed analysis in 5.3 and confirmed our conjecture.

5.2 Ablation Studies

In Table 3, we investigated the effectiveness of node-to-node with node-to-edge attention and learnable edge features.

First, we test the case "Edge Feature through FFN," which is equivalent to updating an edge feature using an FFN without node features involved. Interestingly, the trends on the two datasets are consistent, with a decrease of 2.53 (4.12%) and 1.4 (2.1%), respectively. This shows that FFN needs a good combination of edge feature learning and representation to obtain good results. From another point of view, enhancing the nonlinearity of edge features with FFN may not be the best choice.

Table 3. Ablation studies on best-performed dataset. Metrics for DailyDialog is Micro-F1, and Weighted-F1 For MELD.

	Node + Edge attention			Node Attention	UGT
	Learnable edge feature	Edge feature through FFN	Unlearnable Edge Feature	Unlearnable edge feature	
DailyDialog	**61.37**	58.84	59.98	59.67	59.75
MELD	**65.36**	63.96	64.96	64.63	65.18

Second, we test the case "Unlearnable Edge Feature," where the edge features are fixed and completely unlearnable. Comparing the results, we can find that the unlearnable edge feature reduces the evaluation indicators of the two datasets by 1.39 (2.3%) and 0.4 (0.6%), respectively. Both DailyDialog and MELD are based on short daily dialogues, and we believe that learnable edge features can fully capture the underlying contextual information in such cases.

Third, we do tests when only using node-to-node attention without node-to-edge attention. UGT calculates the attention weights as $Attention(h_i, h_i + e_{ji})$ and this attention mechanism is between node-edge separate attention and node attention only. Comparing the results, we find that the absence of node-to-edge attention leads to a 0.31 (0.5%) and 0.33 (0.5%) drop, respectively, on both datasets.

5.3 Error Analysis

In this part, we analyze from the bottom up which part reduces the model effect. According to Table 4, we can find out that our model results are 0.38 (0.6%) higher than the SKAIG with UGT baseline with a separate attention mechanism only. However, the addition of FFN caused a substantial drop in our model results by about 3.19 (4.7%), which is the highest reduction of all the four datasets. In addition, even though we introduce learnable edge features after removing FFN, the model performance still drops by about 0.83 (1.2%).

Table 4. Error analysis on the best and worst-performing datasets. Metrics are Micro-F1 for DailyDialog and Weighted-F1 for IEMOCAP. Data with * are trained and predicted without FFN.

	UGT	Node attention + Edge attention		
		Unlearnable edge feature	Edge feature through FFN	Learnable edge feature
DailyDialog	59.75	59.98	58.84	**61.37**
IEMOCAP	66.96	**67.34**	64.15	66.54*

Combining all the experimental results, we find that FFN always causes the results to drop no matter in the largest dataset DailyDialog or the smallest dataset, IEMOCAP. We speculate that it may not be appropriate to eliminate edge feature nonlinearity by FFN. Also, the average conversation length of 48 is much longer than the 12 of other datasets. We believe that the context information span of long dialogue is larger than that of short dialogue, and our edge feature update is limited to adjacent nodes, which is more suitable for capturing short-distance context information. Therefore, it can be necessary to adapt the edge learning approach to the dialogue form.

5.4 Conclusion

In this paper, we propose a Learnable Edge Message Passing Model based on a heterogeneous dialog graph, which calculates the node-to-node and node-to-edge attentions separately and updates edge features. The results show our attention mechanisms are generally effective. The learnable edge feature performs differently on different datasets and may need to be improved according to the types of datasets. Although there are few related works in the field, we will strive to find better edge learning methods and use external data to build better graph models in the future.

Acknowledgments. This work was partially supported by the National Nat- ural Science Foundation of China (61876053, 62006062, 62176076), Shenzhen Foundational Research Funding (JCYJ20200109113441941 and JCYJ2021032411 5614039), Joint Lab of HIT and KONKA.

References

1. Adikari, A., De Silva, D., Alahakoon, D., Yu, X.: A cognitive model for emotion awareness in industrial chatbots. In: Proceedings of INDIN, vol. 1, pp. 183–186 (2019)
2. Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y.: COMET: commonsense transformers for automatic knowledge graph construction. In: Proceedings of ACL, pp. 4762–4779 (2019)
3. Busso, C., et al.: IEMOCAP: interactive emotional dyadic motion capture database. Lang. Resour. Eval. **42**(4), 335–359 (2008)
4. Ghosal, D., Majumder, N., Gelbukh, A., Mihalcea, R., Poria, S.: Cosmic: commonsense knowledge for emotion identification in conversations. In: Findings of ACL: EMNLP 2020, pp. 2470–2481 (2020)
5. Ghosal, D., Majumder, N., Poria, S., Chhaya, N., Gelbukh, A.: DialogueGCN: a graph convolutional neural network for emotion recognition in conversation. In: Proceedings of EMNLP-IJCNLP, pp. 154–164 (2019)
6. Hazarika, D., Poria, S., Mihalcea, R., Cambria, E., Zimmermann, R.: Icon: interactive conversational memory network for multimodal emotion detection. In: Proceedings of EMNLP, pp. 2594–2604 (2018)
7. 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 EMNLP, pp. 7360–7370 (2020)
8. Jiang, X., Ji, P., Li, S.: CensNet: convolution with edge-node switching in graph neural networks. In: Proceedings of IJCAI, pp. 2656–2662 (2019)
9. Li, J., Lin, Z., Fu, P., Wang, W.: Past, present, and future: conversational emotion recognition through structural modeling of psychological knowledge. In: Findings of ACL: EMNLP 2021, pp. 1204–1214 (2021)
10. Li, J., Ji, D., Li, F., Zhang, M., Liu, Y.: HiTrans: a transformer-based context-and speaker-sensitive model for emotion detection in conversations. In: Proceedings of COLING, pp. 4190–4200 (2020)
11. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: DailyDialog: a manually labelled multi-turn dialogue dataset. In: Proceedings of IJCNLP, pp. 986–995 (2017)
12. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
13. Majumder, N., Poria, S., Hazarika, D., Mihalcea, R., Gelbukh, A., Cambria, E.: DialogueRNN: an attentive RNN for emotion detection in conversations. In: Proceedings of AAAI, vol. 33, pp. 6818–6825 (2019)
14. Poria, S., Cambria, E., Hazarika, D., Majumder, N., Zadeh, A., Morency, L.P.: Context-dependent sentiment analysis in user-generated videos. In: Proceedings of ACL, pp. 873–883 (2017)
15. Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., Mihalcea, R.: Meld: a multimodal multi-party dataset for emotion recognition in conversations. In: Proceedings of ACL, pp. 527–536 (2019)
16. Poria, S., Majumder, N., Mihalcea, R., Hovy, E.: Emotion recognition in conversation: research challenges, datasets, and recent advances. IEEE Access **7**, 100943–100953 (2019)
17. Shen, T., et al.: PEIA: personality and emotion integrated attentive model for music recommendation on social media platforms. In: Proceedings of AAAI, vol. 34, pp. 206–213 (2020)

18. Shen, W., Chen, J., Quan, X., Xie, Z.: DialogXL: All-in-one XLNet for multi-party conversation emotion recognition. In: Proceedings of AAAI, vol. 35, pp. 13789–13797 (2021)
19. Shen, W., Wu, S., Yang, Y., Quan, X.: Directed acyclic graph network for conversational emotion recognition. In: Proceedings of ACL-IJCNLP, pp. 1551–1560 (2021)
20. Shi, Y., Huang, Z., Wang, W., Zhong, H., Feng, S., Sun, Y.: Masked label prediction: unified massage passing model for semi-supervised classification. In: Proceedings of IJCAI (2021)
21. Vaswani, A., et al.: Attention is all you need. In: NIPS 30 (2017)
22. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. Stat **1050**, 20 (2017)
23. Wang, Z., Chen, J., Chen, H.: EGAT: edge-featured graph attention network. In: Proceedings of ICANN, pp. 253–264 (2021)
24. Yang, Y., Li, D.: NENN: Incorporate node and edge features in graph neural networks. In: Proceedings of ACML, pp. 593–608 (2020)
25. Zahiri, S.M., Choi, J.D.: Emotion detection on tv show transcripts with sequence-based convolutional neural networks. In: Workshops of AAAI (2018)