Relation-aware Graph Attention Networks with Relational Position Encodings for Emotion Recognition in Conversations

Taichi Ishiwatari **Yuki Yasuda** **Taro Miyazaki** **Jun Goto**
NHK Science and Technology Research Laboratories
{ishiwatari.t-fa, yasuda.y-hk, miyazaki.t-jw, goto.j-fw}@nhk.or.jp

Abstract

Interest in emotion recognition in conversations (ERC) has been increasing in various fields, because it can be used to analyze user behaviors and detect fake news. Many recent ERC methods use graph-based neural networks to take the relationships between the utterances of the speakers into account. In particular, the state-of-the-art method considers self- and inter-speaker dependencies in conversations by using relational graph attention networks (RGAT). However, graph-based neural networks do not take sequential information into account. In this paper, we propose *relational position encodings* that provide RGAT with sequential information reflecting the relational graph structure. Accordingly, our RGAT model can capture both the speaker dependency and the sequential information. Experiments on four ERC datasets show that our model is beneficial to recognizing emotions expressed in conversations. In addition, our approach empirically outperforms the state-of-the-art on all of the benchmark datasets.

1 Introduction

Interest in emotion recognition in conversations (ERC) has been increasing in various fields (Picard, 2010), because it can be used to analyze user behaviors (Lee and Hong, 2016) and detect fake news (Guo et al., 2019). With the recent proliferation of social media platforms such as Facebook, Twitter, and YouTube, as well as conversational assistants such as Amazon Alexa, there is a need to study how emotions are expressed in natural conversation.

Recent research on ERC processes the utterances of dialogues in sequence by using recurrent neural network (RNN)-based methods (Hochreiter and Schmidhuber, 1997; Chung et al., 2014; Liu et al., 2016). However, these methods are not able to process long series of information (Bradbury et al., 2016). DialogueRNN tries to make up for this problem by using an attention mechanism to focus on the relevant utterances in the entire conversation (Majumder et al., 2019). However, these methods do not take self-dependency or inter-speaker dependency into account. Table 1 shows the importance of these dependencies, as illustrated by an example dialogue depicting an argument about a job search. Because speaker A has not been able to find a job for a long time, his emotional state is consistently negative. In this way, self-dependency is critical to understanding his own emotional transitions in the conversation. On the other hand, B's emotions shift at utterance ♯4 to commiserate on A's situation. This inter-speaker dependency captures how the utterances of other speakers affect emotions.

♯	Speaker	Utterance	Emotion
1	A	I'm just so tired all the time.	Sad
2	B	Well have you been trying to get a job, look for a job or...?	Neutral
3	A	I've been looking for like eight months.	Frustrated
4	B	I know., It- It's really tough out there., It's really hard to find a job.	Frustrated
5	A	I'm tired of the same excuses., No, no you're not qualified enough, wish you had more education.	Frustrated
6	B	Well what are you looking for?, I mean–	Neutral
7	B	Well, okay. Well that's–	Neutral
8	A	Cause I went to Harvard.	Anger

Table 1: Example for contextual emotion analysis on the IEMOCAP dataset (Busso et al., 2008), which contains emotion-labeled utterances in multi-party conversations.

The state-of-the-art method, DialogueGCN (Ghosal et al., 2019), uses relational graph attention networks (RGAT) to take the dependency

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7360–7370,
November 16–20, 2020.

into account; it is inspired by relational graph convolutional networks (RGCN) (Schlichtkrull et al., 2018) and graph attention networks (GAT) (Veličković et al., 2017). This method takes into account the conversational context by using a directed graph, where the nodes denote individual utterances, the edges represent relationships between pairs of nodes (utterances), and the labels of the edges represent the types of relationships. However, graph-based neural networks do not take sequential information contained in utterances into account. Table 1 also represents the importance of the sequential information. B's emotional change at utterance ♯4 is caused by utterance ♯3 rather than ♯2 or ♯1. In this way, human emotions may depend on more immediate utterances in the temporal order, and thus it is essential to take the sequence of utterances into account.

A common response to this issue is to encode information about absolute position features (Vaswani et al., 2017) or relative position features (Shaw et al., 2018), where these encodings are added to nodes (utterances) or edges (relationships). However, in order to account for self- and inter-speaker dependency, our model focuses on relation types rather than nodes (utterances) and edges (relationships); thus, our position encoding also focuses on relation types.

In this paper, we propose novel position encodings (*relational position encodings*) that provide the RGAT model with sequential information reflecting relation types. By using the relational position encodings, our RGAT model can capture both the speaker dependency and the sequential information. Experiments on four ERC benchmark datasets showed that our relational position encoding outperformed baselines and state-of-the-art methods. In addition, our method outperformed both the absolute and relative position encodings.

In summary, our contributions are as follows: (1) For the first time, we apply position encodings to RGAT to account for sequential information. (2) We propose relational position encodings for the relational graph structure to reflect both sequential information contained in utterances and speaker dependency in conversations. (3) We conduct extensive experiments demonstrating that the graphical model with relational position encodings is beneficial and that our method outperforms state-of-the-art methods on four ERC datasets. (4) We also empirically demonstrate that our model is an effective representation of other positional variations with absolute or relative position encodings.

2 Related Work

Emotion Recognition in Conversation Several studies have tackled the ERC task. Hazarika et al. (2018a,b) used memory networks for recognizing humans emotion in conversation, where two distinct memory networks consider the inter-speaker interaction. DialogueRNN (Majumder et al., 2019) employs an attention mechanism for grasping the relevant utterance from the entire conversation. More related to our method is the DialogueGCN model proposed by Ghosal et al. (2019), in which RGAT is used for modeling both self-dependency and inter-speaker dependency. This model has achieved state-of-the-art performance on several conversational datasets. On the other hand, as a way of considering contextual information, Luo and Wang (2019) proposed to propagate each of the utterances into an embedded vector. Likewise, a pre-trained BERT model (Devlin et al., 2018) has been used for generating dialogue features to combine several utterances by inserting separate tokens (Yang et al., 2019).

Graph Neural Network Graph-based neural networks are used in various tasks. The fundamental model is the graph convolutional network (GCN) (Kipf and Welling, 2016), which uses a fixed adjacency matrix as the edge weight. Our method is based on RGCN (Schlichtkrull et al., 2018) and GAT (Veličković et al., 2017). The RGCN model prepares a different structure for each relation type and hence considers self-dependency and inter-speaker dependency separately. The GAT model uses an attention mechanism to attend to the neighborhood's representations of the utterances.

Position Encodings In our work, positional information is added to the graphical structure. Several studies add position encodings to several structures, such as self-attention networks (SANs) and GCN. SANs (Vaswani et al., 2017) perform the attention operation under the position-unaware assumption, in which the positions of the input are ignored. In response to this issue, the absolute position (Vaswani et al., 2017) or relative position (Shaw et al., 2018), or structure position (Wang et al., 2019) are used to capture the sequential order of the input. Similarly, graph-based neural net-

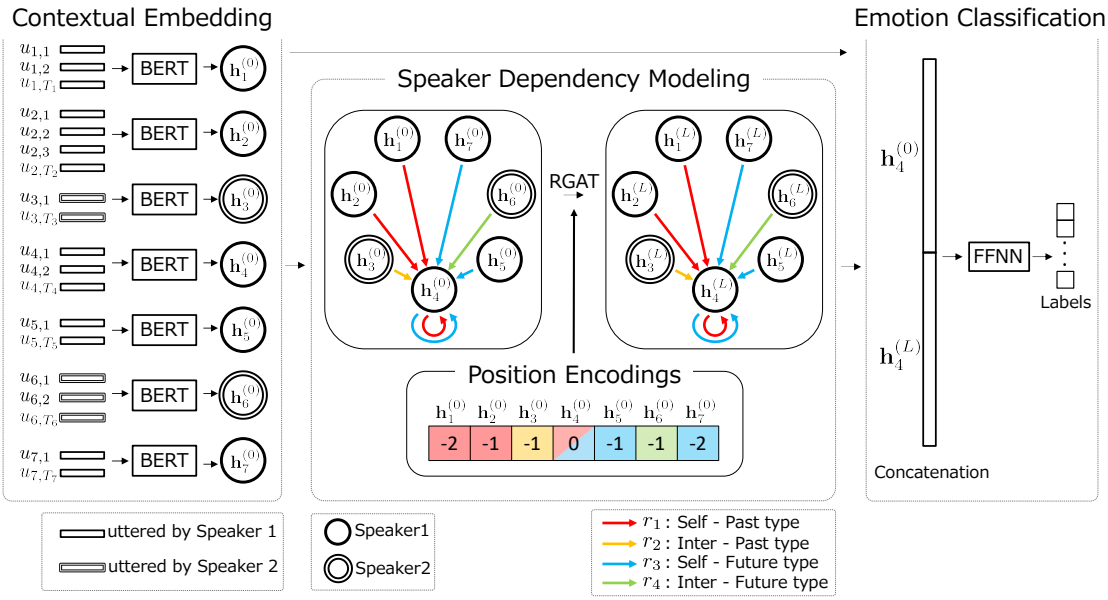

Figure 1: Our entire framework. First, we obtain a contextual embedding for each utterance by using BERT. Then, we modify this embedding by using RGAT to consider speaker dependency. The position encodings in the RGAT structure take sequential information into account. Finally, after concatenating the contextual embedding to the output embedding through RGAT, we classify the concatenated vector into emotion labels by using a fully connected feed-forward network.

works do not take sequential information. In the design of proteins, the relative spatial structure between proteins is modeled in order to account for the complex dependencies in the protein sequence and is applied to the edges of the graph representations (Ingraham et al., 2019).

3 Method

First, we define the problem of the ERC task. The task is to recognize emotion labels (*Happy*, *Sad*, *Neutral*, *Angry*, *Excited*, and *Frustrated*) of utterances $u_1, u_2, \cdots, u_N$, where N denotes the number of utterances in a conversation. Let s_m for $m = 1, \cdots, M$ be a collection of speakers in a given conversational dataset, where M denotes the number of speakers. The utterance u_i is uttered by speaker s_m, where m is the correspondence between the utterance and its speaker.

Our framework consists of three components - contextual utterance embedding, speaker dependency modeling with position encodings and emotion classification. The entire model architecture is shown in Figure 1. Although our method is based on the DialogueGCN (Ghosal et al., 2019) model, it considers the positional information contained in utterances in a sequential conversation as described in Section 3.2.3, whereas the DialogueGCN model does not.

3.1 Contextual Utterance Embedding

We generate contextual utterance features from the tokens by following the method in (Luo and Wang, 2019). First, every utterance $u_1, u_2, \cdots, u_N$ is tokenized by the BPE tokenizer (Sennrich et al., 2015), i.e., $u_i = (u_{i,1}, u_{i,2}, \cdots, u_{i,T_i})$, where T_i denotes the number of tokens. The tokens are embedded through WordPiece embeddings (Wu et al., 2016). The pre-trained *uncased BERT-Base*[1] model converts the token embeddings into contextualized token representations, which can be converted to the vector representations via max pooling, so that they are regarded as the contextual utterance embeddings $\mathbf{h}_i^{(0)} \in \mathbb{R}^{D_m}$ for $i = 1, \cdots, M$, where D_m denotes the dimension of the utterance embeddings. This BERT model is fine-tuned through a training process.

3.2 Speaker Dependency Modeling with Position Encodings

Graph-based neural networks are used to capture the speaker dependency features of conversations. We design relational graph attention net-

[1] See https://github.com/google-research/bert for details.

works to capture both self-dependency and inter-speaker dependency of utterances. In addition, we introduce an attention mechanism to attend to the neighborhood's representations of the utterances. Furthermore, novel position encodings (relational position encodings) are added to the graph to account for the sequential information contained in utterances.

3.2.1 Graphical Structure

We introduce the following notation: we denote directed and multi-graphs as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ with a node (utterance) $v_i \in \mathcal{V}$ and a labeled edge (relation) $(v_i, r, v_j) \in \mathcal{E}$, where $r \in \mathcal{R}$ is a relation type.

Nodes Representation Each utterance in a conversation is represented as a node $v_i \in \mathcal{V}$. Each node v_i is initialized with the contextual utterance embeddings $\mathbf{h}_i^{(0)}$. Through a stack of graphical layers, this embedding is modified by aggregating their neighborhood's representations, described as $\mathbf{h}_i^{(L)}$, where L denotes the number of graphical layers.

Labeled Edges Representation Following the state-of-the-art method (Ghosal et al., 2019), the labeled edges depend on two aspects: (a) speakers dependency - this depends upon both self-dependency and inter-speaker dependency. In detail, the former indicates how utterance u_i of speaker s_m influences s_m's other utterances (including itself). On the other hand, the latter describes how utterance u_i of speaker s_m influences the other speaker $s_{k \neq m}$'s utterances; (b) temporal dependency - this also depends on temporal turns in conversation. Namely, it relies upon whether one utterance u_j is uttered in the past or future of the target utterance u_i. While the future dependencies are not used in on-going conversation, the ERC task is an offline system. Furthermore, as past utterances plausibly influence future utterances, the converse may help the model fill in some missing information like the speaker's background. For these reasons, we take the converse influence into account, referring to (Ghosal et al., 2019).

Accordingly, there are four relation types of edges: (1) self - past type, (2) inter - past type, (3) self - future type, and (4) inter - future type, described as (r_1, r_2, r_3, r_4). Note that this is in contrast to the 8 types used by DialogueGCN[2].

In addition, the window sizes p and f represent the number of past or future utterances from a target utterance in a neighborhood where each utterance u_i has an edge with the p utterances (i.e. $u_{i-1}, u_{i-2}, \cdots, u_{i-p}$), the f utterances (i.e. $u_{i+1}, u_{i+2}, \cdots, u_{i+f}$), and itself. An appropriate window size has to be determined because a small window makes each utterance connect to too small a neighborhood while an immense window size makes the calculation very expensive. Although the window size can be different for each type, we determine the same window size for each relation.

3.2.2 Edge Weight

We introduce an edge weight by using an attention mechanism. Although our attention mechanism is based on the GAT (Veličković et al., 2017) model, it is independent for each relational type r:

$$\alpha_{ijr} = \mathrm{softmax}_i\Big(\mathrm{LRL}\big(\mathbf{a}_r^T[W_r\mathbf{h}_i || W_r\mathbf{h}_j]\big)\Big) \tag{1}$$

where α_{ijr} denotes the edge weight from a target utterance i to its neighborhood j under relational type r, W_r denotes a parametrized weight matrix for the attention mechanism, $\mathbf{a}_r$ denotes a parametrized weight vector, and $\cdot^T$ represents transposition. After applying LeakyReLU nonlinearity (LRL), a softmax function is used to obtain the incoming edges whose sum total weight is 1.

3.2.3 Position Encodings

We propose relational position encodings for the relational graph attention networks. Our position encodings are based on the relative position since it is appropriate for graph-based neural networks. The target utterance feature is connected to its neighborhood by an edge in the graph. Therefore, in order to account for the sequential information between them, we need to consider the distance from the target to its neighborhood, which is undoubtedly the relative distance between utterances. Furthermore, we follow the speaker dependency modeling described in 3.2.1 and use relational graph attention networks. It is necessary that the sequential information depends on the relation type r. In summary, we use a different relative distance for each relation type, which is re-

[2]The type of DialogueGCN depends on 2 distinct speakers and therefore implies 2×4 distinct relation types, which indicates that both the speaker dependency and the temporal dependency are prepared for each distinct speaker.

	$\mathbf{h}_1^{(0)}$	$\mathbf{h}_2^{(0)}$	$\mathbf{h}_3^{(0)}$	$\mathbf{h}_4^{(0)}$	$\mathbf{h}_5^{(0)}$	$\mathbf{h}_6^{(0)}$	$\mathbf{h}_7^{(0)}$	$\mathbf{h}_8^{(0)}$
Absolute Position	1	2	3	4	5	6	7	8
Relative Position	-3	-2	-1	0	1	2	3	4
Relational Position	-2	-1	-1	0	-1	-1	-2	-3

Figure 2: Example of relational positions. The relational position depends on each relational type, and the background color represents the relational type from the target utterance $\mathbf{h}_4$. These positions, which are based on the relative distance, are different for each relation.

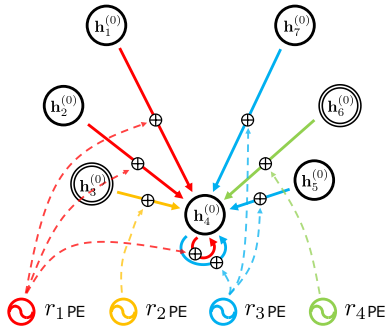

Figure 3: Illustration of relational position encodings. The encodings, which are composed of four representations, are added to the edges in a graph for each relation. "PE" denotes the position encodings.

ferred to as relational position encodings. Figure 2 illustrates the idea of relational positions.

We compare two types of relational position encoding, i.e., a fixed function and a learned representation (Gehring et al., 2017). As the fixed positional function, we define its representation as

$$PE_{ijr} = \begin{cases} max(-p, min(p, j-i)) & r = 1, where\ j \in \mathcal{N}^1(i) \\ max(-p, min(p, j-i)) & r = 2, where\ j \in \mathcal{N}^2(i) \\ max(-f, min(f, j-i)) & r = 3, where\ j \in \mathcal{N}^3(i) \\ max(-f, min(f, j-i)) & r = 4, where\ j \in \mathcal{N}^4(i) \end{cases} \quad (2)$$

where PE_{ijr} denotes the relational distance from a target utterance i to its neighborhood j under relational type r. The maximum relational position is clipped to a size of p or f, which denotes the window size of past or future utterances. $\mathcal{N}^r(i)$ denotes the neighborhood of the target i under relation type r. As the learned representations, we use one-layer feed-forward neural networks for positional embeddings, whose argument is the relational fixed function.

Our relational position is based on the relative position; thus, it can be added to the edge weight, as illustrated in Figure 3. We redefine the attention weight in (1) as

$$\alpha_{ijr} = \mathrm{softmax}_i\Big(\mathrm{LRL}\big(\mathbf{a}_r^T[W_r\mathbf{h}_i||W_r\mathbf{h}_j] + \mathrm{PE}_{ijr}\big)\Big) \quad (3)$$

To add position encodings to the edge weight, our relational position has the same scalar dimension as the edge weight. Because it is a scalar value, it may have limited ability to express positional information. In future studies, we will increase the dimension of the position encodings.

3.2.4 RGAT

A graphical propagation module modifies the representation of a node $\mathbf{h}_i^{(l)}$ by aggregating representations of its neighborhood $\mathcal{N}^r(i)$, and an attention mechanism is used to attend to the neighborhood's representations. The features $\mathbf{h}_{ir}^{(l-1)}$ under relation r are summed to compose the output embedding of a node $\mathbf{h}_i^{(l)}$. Through a stack of graphical layers l, the representation of a node changes within its l-hop neighborhood. We define the propagation module as follows:

$$\mathbf{h}_{ir}^{(l-1)} = \sum_{j \in \mathcal{N}^r(i)} \alpha_{ijr}^{(l-1)} W_r^{(l-1)} \mathbf{h}_j^{(l-1)} \quad (4)$$

$$\mathbf{h}_i^{(l)} = \sum_{r=1}^{R} \mathbf{h}_{ir}^{(l-1)} \quad (5)$$

where $W_r^{(l-1)}$ denotes a learnable weight matrix for each relation r. In addition, We apply multi-head attention to the aggregation module in (4) and concatenate its outputs. After this propagation module in (5), we use layer normalization with learnable affine transform parameters.

3.3 Emotion Classification

After obtaining the representations $\mathbf{h}_i^{(L)}$ of each node through the speaker dependency modeling with relational position encodings, we concatenate the contextual utterance embeddings $\mathbf{h}_i^{(0)}$ and the representation of $\mathbf{h}_i^{(L)}$. The concatenated vector is classified by using a fully connected feed-forward network, which consists of two linear transformations with a ReLu activation between them:

$$\mathrm{Classifier}(\mathbf{x}) = \max(0, \mathbf{x}W_1 + \mathbf{b}_1)W_2 + \mathbf{b}_2 \quad (6)$$

where W_1 and W_2 denote learnable weight matrixes, and $\mathbf{b}_1$ and $\mathbf{b}_2$ denote learnable bias vectors.

Datasets	Conversations			Utterances			Classes	Evaluation Metrics
	train	validation	test	train	validation	test		
IEMOCAP	108	12	31	5320	490	1623	6	Weighted-F1
MELD	1038	114	280	9989	1109	2610	7	Weighted-F1
EmoryNLP	713	99	85	9934	1344	1328	7	Weighted-F1
DailyDialog	11118	1000	1000	87170	8069	7740	7	Micro-F1

Table 2: Dataset descriptions.

4 Experimental Settings

4.1 Datasets

We evaluated our method on four ERC benchmark datasets of various sizes. Training, validation, and test data distributions are reported in Table 2.

IEMOCAP (Busso et al., 2008) is an audio-visual database consisting of recordings of ten speakers in dyadic conversations. The utterances are annotated with one of six emotional labels: happy, sad, neutral, angry, excited, or frustrated.

MELD (Poria et al., 2018) is a multimodal multi-party emotional conversational database created from scripts of the TV series *Friends*. The utterances are annotated with one of seven labels: neutral, happiness, surprise, sadness, anger, disgust, or fear.

EmoryNLP (Zahiri and Choi, 2018) was also collected from *Friends*' TV scripts. It contains different sizes and different types of annotations from those of MELD. The emotion labels include neutral, sad, mad, scared, powerful, peaceful, and joyful.

DailyDialog (Li et al., 2017) is a multi-turn daily dialogue dataset, which contains human-written daily communications. The emotion labels are the same as the ones used in MELD.

4.2 Evaluation Metrics

For DailyDialog, following (Zhong et al., 2019), we calculated the micro-averaged F1 score excluding the majority class (neutral), due to it being an extremely high majority (over 80% occupancy in both training and test sets). For the rest of the datasets, we followed (Zhong et al., 2019; Ghosal et al., 2019) and used the weighted-average F1 score.

4.3 Baselines and State-of-the-Art

For a comprehensive performance evaluation, we compared our model with the following baseline and state-of-the-art methods:

CNN (Kim, 2014) This is a convolutional neural network trained at the utterance-level without contextual information.

CNN+cLSTM (Poria et al., 2017) This model extracts utterance features by using a CNN and captures contextual information from surrounding utterances by using a bi-directional long short term memory (LSTM).

BERT_BASE (Devlin et al., 2018) This BERT-based model extracts contextual information from single sentences and uses it as input. After obtaining the sentence feature, it is classified with emotion labels. We used this model as a contextual utterance feature extractor (Section 3.1).

KET (Zhong et al., 2019) This is the state-of-the-art model for the EmoryNLP and DailyDialog benchmark datasets. KET considers contextual information by using hierarchical self-attention and leverages external commonsense knowledge by using a context-aware graph attention mechanism.

DialogueRNN (Majumder et al., 2019) This model uses a CNN to extract textual information. It uses three GRUs to account for the context and the speakers' features and track the emotional state.

DialogueGCN (Ghosal et al., 2019) This is the state-of-the-art model for the IEMOCAP and MELD datasets. DialogueGCN extracts textual utterance features by using a CNN and extracts sequential contextual features by using a GRU. Further, it captures self-dependency and inter-speaker dependency by using two-layer graph neural networks, which consists of one layer RGAT and one layer GCN.

4.4 Other Settings

We used cross entropy as a training loss for our approach on all datasets. The learning rate was decreased in accordance with a cosine annealing schedule (Loshchilov and Hutter, 2016). We set

Models	IEMOCAP	MELD	EmoryNLP	DailyDialog
CNN	48.18	55.86	32.59	49.34
CNN+cLSTM	54.95	56.87	32.89	50.24
BERT_BASE	53.31	56.21	33.15	53.12
KET	59.56	58.18	34.39	53.37
DialogueRNN	62.75	57.03	31.70	50.65
DialogueGCN	64.18	58.10	-	-
Ours	**65.22**	**60.91**	**34.42**	**54.31**

Table 3: Performance of our method, baseline, and state-of-the-art methods on the three test sets (the values in the table are in terms of the evaluation metrics listed in Table 2). Bold font denotes the best performance. "-" signifies that no results were reported for the given dataset. "Ours" denotes our methods, which are composed of a BERT model and RGAT with relational position encodings. The position representations were learned.

initial learning rates of $4e$-5 in the BERT structure and $2e$-3 in the RGAT structure and used the Adam optimizer (Kingma and Ba, 2014) under the scheduled learning rate with a batch size of 1. The number of dimensions of the contextual embeddings and utterance representations was set to 768, and the size of the internal hidden layer in the emotion classification module was set to 384. We used 8-head attention for calculating the edge weight of RGAT and set 0.1 as the dropout rate in the BERT structure. We also carried out experiments with different contextual past window sizes p and future window sizes f, $(1, 1)$, $(2, 2)$, $(3, 3)$, $(10, 10)$, (all, all), and RGAT layers, $1, 2, 3$. We selected either a concatenated function or a summation function as a mixing operation in the emotion classification module, as described in 3.3. We chose the hyper-parameter that achieved the best score on each dataset by using development data. All of the presented results are averages of 5 runs. We conducted all experiments on a CentOS server using Xeon(R) Gold 6246 CPU with 512GB of memory, and we used Quadro RTX 8000 GPU with 48GB of memory.

5 Results and Discussion

5.1 Comparison with Baselines and State-of-the-Art

We compared the performance of our approach with those of the baselines and state-of-the-art methods listed in Table 3. We have quoted the results for the baselines and state-of-the-art results reported in (Zhong et al., 2019; Ghosal et al., 2019), except for the results of BERT_BASE on IEMOCAP.

For IEMOCAP, our model obtained a weighted average F1 score of 65.22%, outperforming DialogueGCN by more than 1 point. Furthermore, it achieved a weighted average F1 score of 60.91% on the MELD dataset, outperforming DialogueGCN by more than 2 points. For EmoryNLP, it achieved a weighted average F1 score of 34.42%. It achieved a micro-averaged F1 score of 54.31% on the DailyDialog dataset, improving recognition performance over the baselines and KET model by around 1 point. From these results, we can see that adding our position encodings caused an improvement over the baselines, KET, and DialogueGCN on all datasets. Further, it is obvious that our approach is robust across datasets having varying training-data sizes, conversation lengths, and numbers of speakers.

5.2 Analysis of the Experimental Results

Let us investigate the importance of our model components by analyzing the predicted emotional labels, as shown in Table 4. The results of the model using BERT without speaker dependency modeling are listed on row ♯0, while the results of DialogueRNN, as described in Section 4.3, are on row ♯1. The results of DialogueGCN, as described in Section 4.3, are reported in ♯2. The results of the BERT and RGAT model without position encodings are on row ♯3, and those of our model are on ♯4. Note that DialogueGCN's RGAT differs from our model in terms of its graphical structure and relational types.

As shown in the table, our method did not achieve the best score for almost all labels. However, interestingly, it achieved a state-of-the-art average F1 score, which is the target metric on the dataset. A possible reason for this performance is that our method consists of effective components. Each component of BERT and RGAT with position encodings worked well for each label. As a result, these components led to a strong average performance. Each effective component is explained

♯	Models	Background Components: Contextual Utterance Embedding	Background Components: Speaker Dependency Modeling	Happy	Sad	Neutral	Angry	Excited	Frustrated	Average
0	BERT_BASE	BERT	×	37.09	59.53	51.73	54.33	54.26	55.83	53.31
1	DialogueRNN	CNN, GRU		33.18	78.80	59.21	**65.28**	**71.86**	58.91	62.75
2	DialogueGCN	CNN, GRU	RGAT	42.75	**84.54**	63.54	64.19	63.08	**66.99**	64.18
3	Ours(without PE)	BERT	RGAT	50.69	76.78	**65.85**	59.66	64.04	62.37	64.36
4	Ours	BERT	RGAT with PE	**51.62**	77.32	65.42	63.01	67.95	61.23	**65.22**

Table 4: Weighted average F1 scores of ours (with or without PE), baseline, and state-of-the-art methods for each label in the IEMOCAP dataset. Bold font denotes the best performance. "Average" denotes the weighted average F1 score. The variations of their background components are shown in the third and fourth columns.

as follows:

Effect of Speaker Dependency We observed that DialogueGCN and ours (with or without PE) achieved an F1 score of more than 60% on *Frustrated*, higher than the other methods. This may be due to the well-functioning RGAT model. On the IEMOCAP dataset, the utterances often keep on influencing the other utterances through self and inter-speaker dependency; thus, the same label continues in these utterances. Most of the labels in this case are annotated with *Frustrated*. Because of the speaker dependency modeling, these consecutive utterances can be well classified using RGAT.

Effect of Contextual Information Ours (with or without PE) achieved an F1 score of more than 50% on *Happy*, outperforming the other baselines by around 10 points. On the dataset, the *Happy* label appears in several utterances including particular words like 'love' or 'great'. The BERT model with RGAT may have led to better performance. Due to the representational power afforded by its bi-directional context modeling, the BERT model may have functioned well in these utterances. Note that the combination of BERT and RGAT is probably essential because the samples of *Happy* are also influenced by speaker dependency, as compared with ♯0.

Effect of Sequential Feature Our position encodings contributed to the strong performance on the *Sad* and *Angry* labels, our model with PE outperformed our model without PE (♯3 and ♯4). The two labels often appear in the utterances influenced by the other immediate utterances. As the RGAT with position encodings not only captures self and inter-speaker dependency but clearly distinguishes between immediate and far utterances; thus, it possibly performs well on these utterances. Despite its strong performance, our model did not outperform DialogueGCN and DialogueRNN on these labels (♯1, ♯2, and ♯4). A possible explanation is that these label's utterances are mainly influenced by the immediately preceding utterances; thus, RNN-based models such as GRU may be more adequate for these two labels.

From these results, we can see that each component of our method functioned successfully on each label. Our method achieved a state-of-the-art average F1 score. Moreover, it was useful on any label; thus, it is a well-balanced method.

Other Analyses We analyzed other aspects of our models. We observed that our model misclassified some samples of *Excited* as *Happy*. The cause of this issue may be due to the similarity of the sentences these labels appear in. There is almost no difference in the meanings of sentences, so our method may have had difficulty distinguishing these labels. In future work, we will utilize additional audio and visual information to help our model by taking voice tones and facial expressions into account.

5.3 Model Variations

We evaluated the importance of our relational position encoding and studied the positional variations on the IEMOCAP dataset. The experimental results are reported in Table 5.

To make comparisons with the other position encoding methods, absolute and relative position representations were prepared; these are referred to as node-based position encodings and edge-based position encodings, respectively. Inspired by (Vaswani et al., 2017), we added node-based position encoding to the nodes (utterances) at the bottoms of the RGAT layers. Similarly, edge-based position encoding was added to the edges in the graph. We also compared two types of posi-

♯	Position Encodings (PE)	Type	Average
0	-	-	64.36
1	Node-based PE	*fixed*	63.95
2		*learn*	64.95
3	Edge-based PE	*fixed*	63.97
4		*learn*	64.59
5	Relational PE	*fixed*	63.99
6		*learn*	**65.22**

Table 5: Impact of various position encodings components on the IEMOCAP dataset. The base model using BERT and RGAT without position encodings is shown in ♯0. "*fixed*" and "*learn*" denote a fixed function and a learned representation respectively.

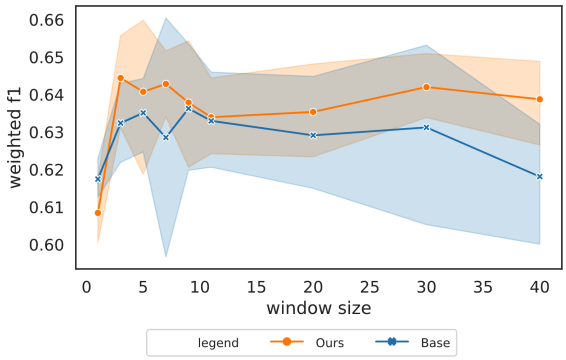

Figure 4: Effect of different window sizes on the weighted average F1 score of our method (Ours) and the baseline model (Base) on the IEMOCAP dataset. We plotted the scores by using a marker with a confidence interval of 95%, which was estimated using a bootstrap.

tion encoding, i.e., a fixed function and a learned representation.

The baseline model using BERT and RGAT without position encodings (♯0) had a recognition performance of 64.36%. We added various position encodings to the baseline model and selected fixed functions or learned representations as the position representation (from ♯1 to ♯6). The model using the relational position encodings with learned representations had a recognition performance of 65.22%, the best score and outperforming the base model by around 1 point. Our relational position encodings were more effective than the other position encodings.

We also found that the fixed functions in various positions resulted in a score lower than that of the baseline model. We can conclude that it is required to learn a position representation.

5.4 Effect of Varying the Window Size

We conducted another experiment to evaluate the key aspects of our framework. We carried out an experiment by increasing the past and future window sizes [(1,1), (3,3), (5,5), (7,7), (9,9), (11,11), (20,20), (30,30), and (40,40)] on the IEMOCAP dataset and compared the results with those of the baseline model using BERT and RGAT without positional information. The experimental results are illustrated in Figure 4.

As an illustration, it is clear that both models perform better with a window size around $3, 5, 7$. On the other hand, long utterance information may obstruct efficient recognition (see the results for a window size around $30, 40$). Although it is required to select a small window size, too small a size results in poor performance, no better than choosing a size of 1.

Furthermore, the proposed position encoding method is robust to a varying window size. As the window size increased, the baseline model's F1 score decreased, while our model maintained its performance even with a large window. One possible reason is that, as our position encodings clearly distinguish between immediate and far utterances, it can reduce the influence of these distant utterances.

6 Conclusion

We proposed relational position encodings for RGAT to recognize human emotions in textual conversation. We incorporated the relational position encodings in the RGAT structure to capture both speaker dependency and the sequential order of utterances. On four ERC datasets, our model improved recognition performance over those of the baselines and existing state-of-the-art methods. Additional experimental studies demonstrated that the relational position encoding approach outperformed the other position encodings and showed that it is robust to changes in window size.

In future studies, we plan to increase the number of dimensions of the relational position encodings, since a scalar value may not be able to express positional information adequately.

Acknowledgements

We would like to thank Dr. Ichiro Yamada, Dr. Rei Endo, and Hideya Mino for the valuable discussions. We also thank the anonymous reviewers for their helpful comments.

References

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2016. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. *arXiv preprint arXiv:1908.11540*.

Chuan Guo, Juan Cao, Xueyao Zhang, Kai Shu, and Huan Liu. 2019. DEAN: Learning dual emotion for fake news detection on social media. *arXiv preprint arXiv:1903.01728*.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018a. ICON: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018b. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2122–2132.

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

John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. 2019. Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems*, pages 15794–15805.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Jieun Lee and Ilyoo B Hong. 2016. Predicting positive user responses to social media advertising: The roles of emotional appeal, informativeness, and creativity. *International Journal of Information Management*, 36(3):360–373.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Ilya Loshchilov and Frank Hutter. 2016. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Linkai Luo and Yue Wang. 2019. EmotionX-HSU: Adopting pre-trained bert for emotion classification. *arXiv preprint arXiv:1907.09669*.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. DialogueRNN: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6818–6825.

Rosalind W Picard. 2010. Affective computing: From laughter to ieee. *IEEE Transactions on Affective Computing*, 1(1):11–17.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. 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)*, pages 873–883.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. MELD: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019. Self-attention with structural position representations. *arXiv preprint arXiv:1909.00383*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Kisu Yang, Dongyub Lee, Taesun Whang, Seolhwa Lee, and Heuiseok Lim. 2019. EmotionX-KU: Bert-max based contextual emotion classifier. *arXiv preprint arXiv:1906.11565*.

Sayyed M Zahiri and Jinho D Choi. 2018. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. *arXiv preprint arXiv:1909.10681*.