DialogueGCN: A Graph Convolutional Neural Network for Emotion Recognition in Conversation

Deepanway Ghosal[†], **Navonil Majumder**[‡], **Soujanya Poria**[†*], **Niyati Chhaya**[▽] **and Alexander Gelbukh**[‡]
[†] Singapore University of Technology and Design, Singapore
[‡]Instituto Politécnico Nacional, CIC, Mexico
[▽] Adobe Research, India

{1004721@mymail.,sporia@}sutd.edu.sg, navo@nlp.cic.ipn.mx, nchhaya@adobe.com, gelbukh@gelbukh.com

Abstract

Emotion recognition in conversation (ERC) has received much attention, lately, from researchers due to its potential widespread applications in diverse areas, such as health-care, education, and human resources. In this paper, we present Dialogue Graph Convolutional Network (DialogueGCN), a graph neural network based approach to ERC. We leverage self and inter-speaker dependency of the interlocutors to model conversational context for emotion recognition. Through the graph network, DialogueGCN addresses context propagation issues present in the current RNN-based methods. We empirically show that this method alleviates such issues, while outperforming the current state of the art on a number of benchmark emotion classification datasets.

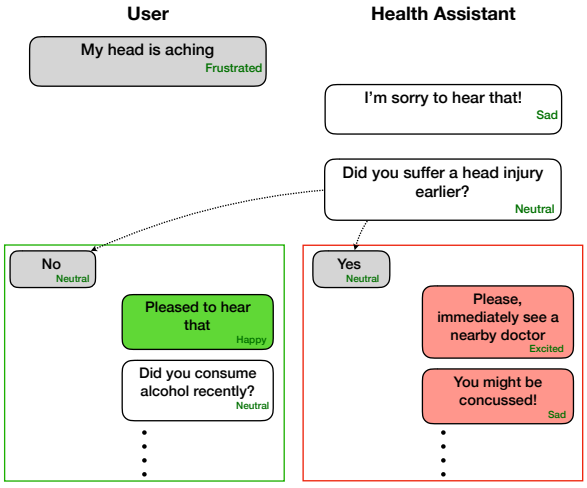


Figure 1: Illustration of an affective conversation where the emotion depends on the context. Health assistant understands affective state of the user in order to generate affective and empathetic responses.

1 Introduction

Emotion recognition has remained an active research topic for decades (K. D'Mello et al., 2006; Busso et al., 2008; Strapparava and Mihalcea, 2010). However, the recent proliferation of open conversational data on social media platforms, such as Facebook, Twitter, Youtube, and Reddit, has warranted serious attention (Poria et al., 2019b; Majumder et al., 2019; Huang et al., 2019) from researchers towards emotion recognition in conversation (ERC). ERC is also undeniably important in affective dialogue systems (as shown in Fig. 1) where bots understand users' emotions and sentiment to generate emotionally coherent and empathetic responses.

Recent works on ERC process the constituent utterances of a dialogue in sequence, with a recurrent neural network (RNN). Such a scheme is illustrated in Fig. 2 (Poria et al., 2019b), that relies on propagating contextual and sequential information to the utterances. Hence, we feed the conversation to a bidirectional gated recurrent unit (GRU) (Chung et al., 2014). However, like most of the current models, we also ignore intent modelling, topic, and personality due to lack of labelling on those aspects in the benchmark datasets. In theory, RNNs like long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and GRU should propagate long-term contextual information. However, in practice it is not always the case (Bradbury et al., 2017). This affects the efficacy of RNN-based models in various tasks, including ERC.

To mitigate this issue, some variants of the state-of-the-art method, DialogueRNN (Majumder et al., 2019), employ attention mechanism that pools information from entirety or part of the conversation per target utterance. However, this pooling mechanism does not consider speaker information of the utterances and the relative position of other utterances from the target utterance. Speaker information is necessary for mod-

* Corresponding author

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 154–164, Hong Kong, China, November 3–7, 2019.

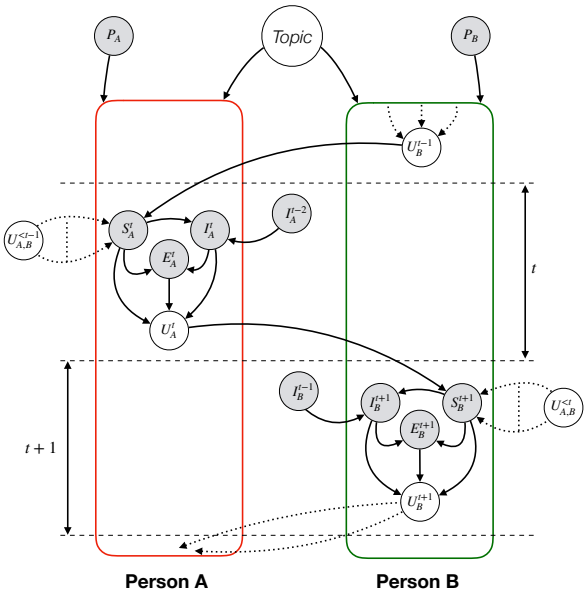


Figure 2: Interaction among different controlling variables during a dyadic conversation between persons A and B. Grey and white circles represent hidden and observed variables, respectively. P represents personality, U represents utterance, S represents interlocutor state, I represents interlocutor intent, E represents emotion and *Topic* represents topic of the conversation. This can easily be extended to multi-party conversations.

elling inter-speaker dependency, which enables the model to understand how a speaker coerces emotional change in other speakers. Similarly, by extension, intra-speaker or self-dependency aids the model with the understanding of emotional inertia of individual speakers, where the speakers resist the change of their own emotion against external influence. On the other hand, consideration of relative position of target and context utterances decides how past utterances influence future utterances and vice versa. While past utterances influencing future utterances is natural, the converse may help the model fill in some relevant missing information, that is part of the speaker's background knowledge but explicitly appears in the conversation in the future. We leverage these two factors by modelling conversation using a directed graph. The nodes in the graph represent individual utterances. The edges between a pair of nodes/utterances represent the dependency between the speakers of those utterances, along with their relative positions in the conversation. By feeding this graph to a graph convolution network (GCN) (Defferrard et al., 2016), consisting of two consecutive convolution operations, we propagate contextual information among distant utterances. We surmise that these representations hold richer context relevant to emotion than DialogueRNN. This is empirically shown in Section 5.

The remainder of the paper is organized as follows — Section 2 briefly discusses the relevant and related works on ERC; Section 3 elaborates the method; Section 4 lays out the experiments; Section 5 shows and interprets the experimental results; and finally, Section 6 concludes the paper.

2 Related Work

Emotion recognition in conversation is a popular research area in natural language processing (Kratzwald et al., 2018; Colneriĉ and Demsar, 2018) because of its potential applications in a wide area of systems, including opinion mining, health-care, recommender systems, education, etc.

However, emotion recognition in conversation has attracted attention from researchers only in the past few years due to the increase in availability of open-sourced conversational datasets (Chen et al., 2018; Zhou et al., 2018; Poria et al., 2019a). A number of models has also been proposed for emotion recognition in multimodal data i.e. datasets with textual, acoustic and visual information. Some of the important works include (Poria et al., 2017; Chen et al., 2017; Zadeh et al., 2018a,b; Hazarika et al., 2018a,b), where mainly deep learning-based techniques have been employed for emotion (and sentiment) recognition in conversation, in only textual and multimodal settings. The current state-of-the-art model in emotion recognition in conversation is (Majumder et al., 2019), where authors introduced a party state and global state based recurrent model for modelling the emotional dynamics.

Graph neural networks have also been very popular recently and have been applied to semi-supervised learning, entity classification, link prediction, large scale knowledge base modelling, and a number of other problems (Kipf and Welling, 2016; Schlichtkrull et al., 2018; Bruna et al., 2013). Early work on graph neural networks include (Scarselli et al., 2008). Our graph model is closely related to the graph relational modelling work introduced in (Schlichtkrull et al., 2018).

3 Methodology

One of the most prominent strategies for emotion recognition in conversations is contextual mod-

elling. We identify two major types of context in ERC – sequential context and speaker-level context. Following Poria et al. (2017), we model these two types of context through the neighbouring utterances, per target utterance.

Computational modeling of context should also consider emotional dynamics of the interlocutors in a conversation. Emotional dynamics is typically subjected to two major factors in both dyadic and multiparty conversational systems — inter-speaker dependency and self-dependency. Inter-speaker dependency refers to the emotional influence that counterparts produce in a speaker. This dependency is closely related to the fact that speakers tend to mirror their counterparts to build rapport during the course of a dialogue (Navaretta et al., 2016). However, it must be taken into account, that not all participants are going to affect the speaker in identical way. Each participant generally affects each other participants in unique ways. In contrast, self-dependency, or emotional inertia, deals with the aspect of emotional influence that speakers have on themselves during conversations. Participants in a conversation are likely to stick to their own emotional state due to their emotional inertia, unless the counterparts invoke a change. Thus, there is always a major interplay between the inter-speaker dependency and self-dependency with respect to the emotional dynamics in the conversation.

We surmise that combining these two distinct yet related contextual information schemes (sequential encoding and speaker level encoding) would create enhanced context representation leading to better understanding of emotional dynamics in conversational systems.

3.1 Problem Definition

Let there be M speakers/parties $p_1, p_2, \ldots, p_M$ in a conversation. The task is to predict the emotion labels (*happy*, *sad*, *neutral*, *angry*, *excited*, *frustrated*, *disgust*, and *fear*) of the constituent utterances $u_1, u_2, \ldots, u_N$, where utterance u_i is uttered by speaker $p_{s(u_i)}$, while s being the mapping between utterance and index of its corresponding speaker. We also represent $u_i \in \mathbb{R}^{D_m}$ as the feature representation of the utterance, obtained using the feature extraction process described below.

3.2 Context Independent Utterance-Level Feature Extraction

A convolutional neural network (Kim, 2014) is used to extract textual features from the transcript of the utterances. We use a single convolutional layer followed by max-pooling and a fully connected layer to obtain the feature representations for the utterances. The input to this network is the 300 dimensional pretrained 840B GloVe vectors (Pennington et al., 2014). We use filters of size 3, 4 and 5 with 50 feature maps in each. The convoluted features are then max-pooled with a window size of 2 followed by the ReLU activation (Nair and Hinton, 2010). These are then concatenated and fed to a 100 dimensional fully connected layer, whose activations form the representation of the utterance. This network is trained at utterance level with the emotion labels.

3.3 Model

We now present our Dialogue Graph Convolutional Network (DialogueGCN[1]) framework for emotion recognition in conversational setups. DialogueGCN consists of three integral components — Sequential Context Encoder, Speaker-Level Context Encoder, and Emotion Classifier. An overall architecture of the proposed framework is illustrated in Fig. 3.

3.3.1 Sequential Context Encoder

Since, conversations are sequential by nature, contextual information flows along that sequence. We feed the conversation to a bidirectional gated recurrent unit (GRU) to capture this contextual information: $g_i = \overleftrightarrow{GRU_S}(g_{i(+,-)1}, u_i)$, for i = 1, 2,..., N, where u_i and g_i are context-independent and sequential context-aware utterance representations, respectively.

Since, the utterances are encoded irrespective of its speaker, this initial encoding scheme is speaker agnostic, as opposed to the state of the art, DialogueRNN (Majumder et al., 2019).

3.3.2 Speaker-Level Context Encoder

We propose the Speaker-Level Context Encoder module in the form of a graphical network to capture speaker dependent contextual information in a conversation. Effectively modelling speaker level context requires capturing the inter-dependency

[1]Implementation available at https://github.com/SenticNet/conv-emotion

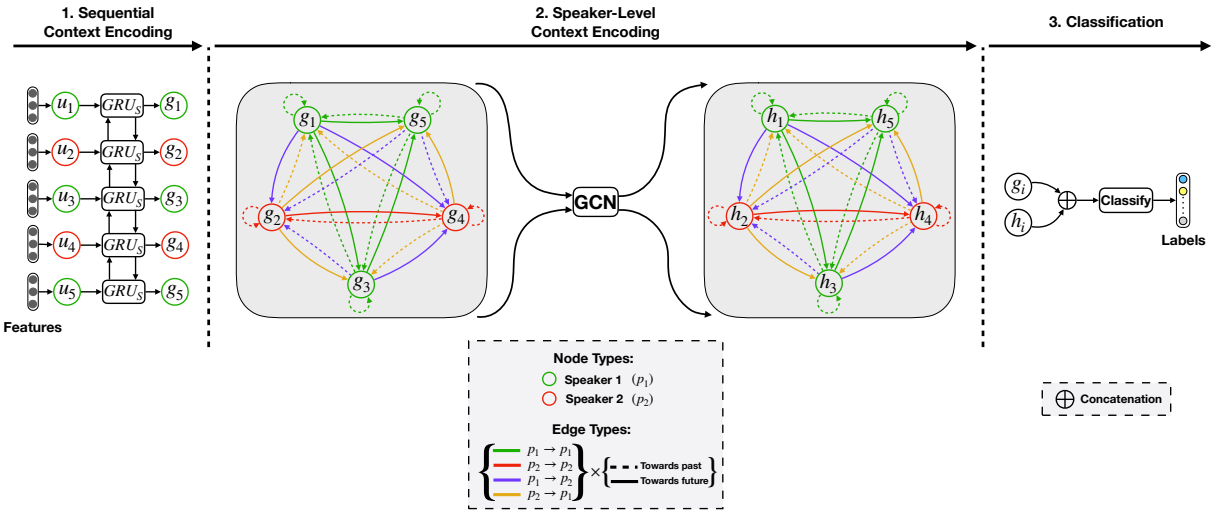


Figure 3: Overview of DialogueGCN, congruent to the illustration in Table 1.

and self-dependency among participants. We design a directed graph from the sequentially encoded utterances to capture this interaction between the participants. Furthermore, we propose a local neighbourhood based convolutional feature transformation process to create the enriched speaker-level contextually encoded features. The framework is detailed here.

First, we introduce the following notation: a conversation having N utterances is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathcal{W})$, with vertices/nodes $v_i \in \mathcal{V}$, labeled edges (relations) $r_{ij} \in \mathcal{E}$ where $r \in \mathcal{R}$ is the relation type of the edge between v_i and v_j and α_{ij} is the weight of the labeled edge r_{ij}, with $0 \leqslant \alpha_{ij} \leqslant 1$, where $\alpha_{ij} \in \mathcal{W}$ and $i, j \in [1, 2, ..., N]$.

Graph Construction: The graph is constructed from the utterances in the following way,

Vertices: Each utterance in the conversation is represented as a vertex $v_i \in \mathcal{V}$ in $\mathcal{G}$. Each vertex v_i is initialized with the corresponding sequentially encoded feature vector g_i, for all $i \in [1, 2, ..., N]$. We denote this vector as the vertex feature. Vertex features are subject to change downstream, when the neighbourhood based transformation process is applied to encode speaker-level context.

Edges: Construction of the edges $\mathcal{E}$ depends on the context to be modeled. For instance, if we hypothesize that each utterance (vertex) is contextually dependent on all the other utterances in a conversation (when encoding speaker level information), then a fully connected graph would be constructed. That is each vertex is connected to all the other vertices (including itself) with an edge. However, this results in $O(N^2)$ number of edges, which is computationally very expensive for graphs with large number of vertices. A more practical solution is to construct the edges by keeping a past context window size of p and a future context window size of f. In this scenario, each utterance vertex v_i has an edge with the immediate p utterances of the past: $v_{i-1}, v_{i-2}, ..v_{i-p}$, f utterances of the future: $v_{i+1}, v_{i+2}, ..v_{i+f}$ and itself: v_i. For all our experiments in this paper, we consider a past context window size of 10 and future context window size of 10.

As the graph is directed, two vertices can have edges in both directions with different relations.

Edge Weights: The edge weights are set using a similarity based attention module. The attention function is computed in a way such that, for each vertex, the incoming set of edges has a sum total weight of 1. Considering a past context window size of p and a future context window size of f, the weights are calculated as follows,

$$\alpha_{ij} = \text{softmax}(g_i^T W_e [g_{i-p}, \ldots, g_{i+f}]), \quad \text{for } j = i-p, \ldots, i+f. \tag{1}$$

This ensures that, vertex v_i which has incoming edges with vertices $v_{i-p}, \ldots, v_{i+f}$ (as speaker-level context) receives a total weight contribution of 1.

Relations: The relation r of an edge r_{ij} is set depending upon two aspects:

Speaker dependency — The relation depends on both the speakers of the constituting vertices: $p_{s(u_i)}$ (speaker of v_i) and $p_{s(u_j)}$ (speaker of v_j).

Temporal dependency — The relation also de-

pends upon the relative position of occurrence of u_i and u_j in the conversation: whether u_i is uttered before u_j or after. If there are M distinct speakers in a conversation, there can be a maximum of M (speaker of u_i) $*M$ (speaker of u_j) $*2$ (u_i occurs before u_j or after) = $2M^2$ distinct relation types r in the graph $\mathcal{G}$.

Each speaker in a conversation is uniquely affected by each other speaker, hence we hypothesize that explicit declaration of such relational edges in the graph would help in capturing the inter-dependency and self-dependency among the speakers in play, which in succession would facilitate speaker-level context encoding.

As an illustration, let two parties p_1, p_2 participate in a dyadic conversation having 5 utterances, where u_1, u_3, u_5 are uttered by p_1 and u_2, u_4 are uttered by p_2. If we consider a fully connected graph, the edges and relations will be constructed as shown in Table 1.

Feature Transformation: We now describe the methodology to transform the sequentially encoded features using the graph network. The vertex feature vectors (g_i) are initially speaker independent and thereafter transformed into a speaker dependent feature vector using a two-step graph convolution process. Both of these transformations can be understood as special cases of a basic differentiable message passing method (Gilmer et al., 2017).

In the first step, a new feature vector $h_i^{(1)}$ is computed for vertex v_i by aggregating local neighbourhood information (in this case neighbour utterances specified by the past and future context window size) using the relation specific transformation inspired from (Schlichtkrull et al., 2018):

$$h_i^{(1)} = \sigma\Big(\sum_{r\in\mathcal{R}}\sum_{j\in N_i^r}\frac{\alpha_{ij}}{c_{i,r}}W_r^{(1)}g_j + \alpha_{ii}W_0^{(1)}g_i\Big), \quad \text{for } i = 1,2,\ldots,N, \tag{2}$$

where, α_{ij} and α_{ii} are the edge weights, N_i^r denotes the neighbouring indices of vertex i under relation $r \in \mathcal{R}$. $c_{i,r}$ is a problem specific normalization constant which either can be set in advance, such that, $c_{i,r} = |N_i^r|$, or can be automatically learned in a gradient based learning setup. σ is an activation function such as ReLU, $W_r^{(1)}$ and $W_0^{(1)}$ are learnable parameters of the transformation. In the second step, another local neighbourhood based transformation is applied over the output of the first step,

$$h_i^{(2)} = \sigma\Big(\sum_{j\in N_i^r} W^{(2)}h_j^{(1)} + W_0^{(2)}h_i^{(1)}\Big), \quad \text{for } i = 1,2,\ldots,N, \tag{3}$$

where, $W^{(2)}$ and $W_0^{(2)}$ are parameters of these transformation and σ is the activation function.

This stack of transformations, Eqs. (2) and (3), effectively accumulates normalized sum of the local neighbourhood (features of the neighbours) i.e. the neighbourhood speaker information for each utterance in the graph. The self connection ensures self dependent feature transformation.

Emotion Classifier: The contextually encoded feature vectors g_i (from sequential encoder) and $h_i^{(2)}$ (from speaker-level encoder) are concatenated and a similarity-based attention mechanism is applied to obtain the final utterance representation:

$$h_i = [g_i, h_i^{(2)}], \tag{4}$$

$$\beta_i = \text{softmax}(h_i^T W_\beta[h_1, h_2 \ldots, h_N]), \tag{5}$$

$$\tilde{h}_i = \beta_i[h_1, h_2, \ldots, h_N]^T. \tag{6}$$

Finally, the utterance is classified using a fully-connected network:

$$l_i = \text{ReLU}(W_l\tilde{h}_i + b_l), \tag{7}$$

$$\mathcal{P}_i = \text{softmax}(W_{smax}l_i + b_{smax}), \tag{8}$$

$$\hat{y}_i = \underset{k}{\text{argmax}}(\mathcal{P}_i[k]). \tag{9}$$

Relation	$p_s(u_i), p_s(u_j)$	$i < j$	(i, j)
1	p_1, p_1	Yes	(1,3), (1,5), (3,5)
2	p_1, p_1	No	(1,1), (3,1), (3,3) (5,1), (5,3), (5,5)
3	p_2, p_2	Yes	(2,4)
4	p_2, p_2	No	(2,2), (4,2), (4, 4)
5	p_1, p_2	Yes	(1,2), (1,4), (3,4)
6	p_1, p_2	No	(3,2), (5,2), (5,4)
7	p_2, p_1	Yes	(2,3), (2,5), (4,5)
8	p_2, p_1	No	(2,1), (4,1), (4,3)

Table 1: $p_s(u_i)$ and $p_s(u_j)$ denotes the speaker of utterances u_i and u_j. 2 distinct speakers in the conversation implies $2 * M^2 = 2 * 2^2 = 8$ distinct relation types. The rightmost column denotes the indices of the vertices of the constituting edge which has the relation type indicated by the leftmost column.

Training Setup: We use categorical cross-entropy along with L2-regularization as the measure of loss (L) during training:

$$L = -\frac{1}{\sum_{s=1}^{N} c(s)} \sum_{i=1}^{N} \sum_{j=1}^{c(i)} \log \mathcal{P}_{i,j}[y_{i,j}] + \lambda \|\theta\|_2 , \tag{10}$$

where N is the number of samples/dialogues, $c(i)$ is the number of utterances in sample i, $\mathcal{P}_{i,j}$ is the probability distribution of emotion labels for utterance j of dialogue i, $y_{i,j}$ is the expected class label of utterance j of dialogue i, λ is the L2-regularizer weight, and θ is the set of all trainable parameters.

We used stochastic gradient descent based Adam (Kingma and Ba, 2014) optimizer to train our network. Hyperparameters were optimized using grid search.

4 Experimental Setting

4.1 Datasets Used

We evaluate our DialogueGCN model on three benchmark datasets — IEMOCAP (Busso et al., 2008), AVEC (Schuller et al., 2012), and MELD (Poria et al., 2019a). All these three datasets are multimodal datasets containing textual, visual and acoustic information for every utterance of each conversation. However, in this work we focus on conversational emotion recognition only from the textual information. Multimodal emotion recognition is outside the scope of this paper, and is left as future work.

IEMOCAP (Busso et al., 2008) dataset contains videos of two-way conversations of ten unique speakers, where only the first eight speakers from session one to four belong to the trainset. Each video contains a single dyadic dialogue, segmented into utterances. The utterances are annotated with one of six emotion labels, which are happy, sad, neutral, angry, excited, and frustrated.

AVEC (Schuller et al., 2012) dataset is a modification of SEMAINE database (McKeown et al., 2012) containing interactions between humans and artificially intelligent agents. Each utterance of a dialogue is annotated with four real valued affective attributes: valence ($[-1, 1]$), arousal ($[-1, 1]$), expectancy ($[-1, 1]$), and power ($[0, \infty)$). The annotations are available every 0.2 seconds in the original database. However, in order to adapt the annotations to our need of utterance-level annotation, we averaged the attributes over the span of an utterance.

MELD (Poria et al., 2019a) is a multimodal emotion/sentiment classification dataset which has been created by the extending the EmotionLines dataset (Chen et al., 2018). Contrary to IEMOCAP and AVEC, MELD is a multiparty dialog dataset. MELD contains textual, acoustic and visual information for more than 1400 dialogues and 13000 utterances from the Friends TV series. Each utterance in every dialog is annotated as one of the seven emotion classes: anger, disgust, sadness, joy, surprise, fear or neutral.

Dataset	# dialogues			# utterances		
	train	val	test	train	val	test
IEMOCAP	120		31	5810		1623
AVEC	63		32	4368		1430
MELD	1039	114	280	9989	1109	2610

Table 2: Training, validation and test data distribution in the datasets. No predefined train/val split is provided in IEMOCAP and AVEC, hence we use 10% of the training dialogues as validation split.

4.2 Baselines and State of the Art

For a comprehensive evaluation of DialogueGCN, we compare our model with the following baseline methods:

CNN (Kim, 2014) This is the baseline convolutional neural network based model which is identical to our utterance level feature extractor network (Section 3.2). This model is context independent as it doesn't use information from contextual utterances.

Memnet (Sukhbaatar et al., 2015) This is an end-to-end memory network baseline (Hazarika et al., 2018b). Every utterance is fed to the network and the memories, which correspond to the previous utterances, is continuously updated in a multi-hop fashion. Finally the output from the memory network is used for emotion classification.

c-LSTM (Poria et al., 2017) Context-aware utterance representations are generated by capturing the contextual content from the surrounding utterances using a Bi-directional LSTM (Hochreiter and Schmidhuber, 1997) network. The context-aware utterance representations are then used for emotion classification. The contextual-LSTM

model is speaker independent as it doesn't model any speaker level dependency.

c-LSTM+Att (Poria et al., 2017) In this variant of c-LSTM, an attention module is applied to the output of c-LSTM at each timestamp by following Eqs. (5) and (6). Generally this provides better context to create a more informative final utterance representation.

CMN (Hazarika et al., 2018b) CMN models utterance context from dialogue history using two distinct GRUs for two speakers. Finally, utterance representation is obtained by feeding the current utterance as query to two distinct memory networks for both speakers. However, this model can only model conversations with two speakers.

ICON (Hazarika et al., 2018b) ICON which is an extension of CMN, connects outputs of individual speaker GRUs in CMN using another GRU for explicit inter-speaker modeling. This GRU is considered as a memory to track the overall conversational flow. Similar to CMN, ICON can not be extended to apply on multiparty datasets e.g., MELD.

DialogueRNN (Majumder et al., 2019) This is the state-of-the-art method for ERC. It is a recurrent network that uses two GRUs to track individual speaker states and global context during the conversation. Further, another GRU is employed to track emotional state through the conversation. DialogueRNN claims to model inter-speaker relation and it can be applied on multiparty datasets.

5 Results and Discussions

5.1 Comparison with State of the Art and Baseline

We compare the performance of our proposed DialogueGCN framework with the state-of-the-art DialogueRNN and baseline methods in Tables 3 and 4. We report all results with average of 5 runs. Our DialogueGCN model outperforms the SOTA and all the baseline models, on all the datasets, while also being statistically significant under the paired t-test ($p < 0.05$).

IEMOCAP and AVEC: On the IEMOCAP dataset, DialogueGCN achieves new state-of-the-art average F1-score of 64.18% and accuracy of 65.25%, which is around 2% better than DialogueRNN, and at least 5% better than all the other baseline models. Similarly, on AVEC dataset, DialogueGCN outperforms the state-of-the-art on all the four emotion dimensions: valence, arousal, expectancy, and power.

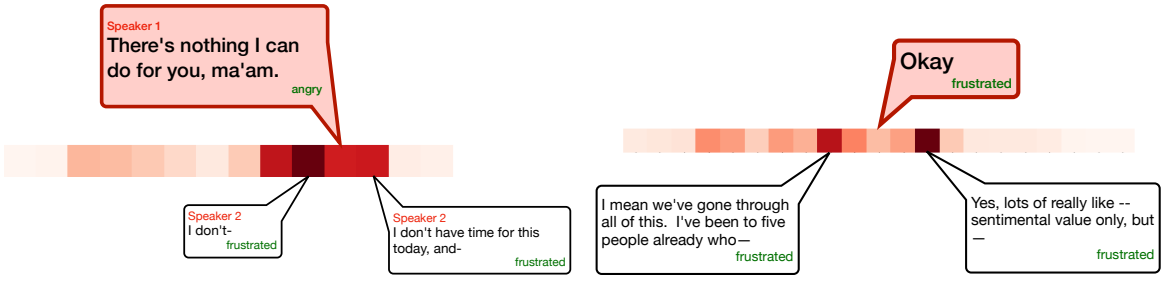
To explain this gap in performance, it is important to understand the nature of these models. DialogueGCN and DialogueRNN both try to model speaker-level context (albeit differently), whereas, none of the other models encode speaker-level context (they only encode sequential context). This is a key limitation in the baseline models, as speaker-level context is indeed very important in conversational emotion recognition.

As for the difference of performance between DialogueRNN and DialogueGCN, we believe that this is due to the different nature of speaker-level context encoding. DialogueRNN employs a gated recurrent unit (GRU) network to model individual speaker states. Both IEMOCAP and AVEC dataset has many conversations with over 70 utterances (the average conversation length is 50 utterances in IEMOCAP and 72 in AVEC). As recurrent encoders have long-term information propagation issues, speaker-level encoding can be problematic for long sequences like those found in these two datasets. In contrast, DialogueGCN tries to overcome this issue by using neighbourhood based convolution to model speaker-level context.

MELD: The MELD dataset consists of multiparty conversations and we found that emotion recognition in MELD is considerably harder to model than IEMOCAP and AVEC - which only consists of dyadic conversations. Utterances in MELD are much shorter and rarely contain emotion specific expressions, which means emotion modelling is highly context dependent. Moreover, the average conversation length is 10 utterances, with many conversations having more than 5 participants, which means majority of the participants only utter a small number of utterances per conversation. This makes inter-dependency and self-dependency modeling difficult. Because of these reasons, we found that the difference in results between the baseline models and DialogueGCN is not as contrasting as it is in the case of IEMOCAP and AVEC. Memnet, CMN, and ICON are not suitable for this dataset as they exclusively work in dyadic conversations. Our DialogueGCN model achieves new state-of-the-art F1 score of 58.10% outperforming DialogueRNN by more than 1%.

Methods	IEMOCAP													
	Happy		Sad		Neutral		Angry		Excited		Frustrated		**Average(w)**	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
CNN	27.77	29.86	57.14	53.83	34.33	40.14	61.17	52.44	46.15	50.09	62.99	55.75	48.92	48.18
Memnet	25.72	33.53	55.53	61.77	58.12	52.84	59.32	55.39	51.50	58.30	67.20	59.00	55.72	55.10
bc-LSTM	29.17	34.43	57.14	60.87	54.17	51.81	57.06	56.73	51.17	57.95	67.19	58.92	55.21	54.95
bc-LSTM+Att	30.56	35.63	56.73	62.90	57.55	53.00	59.41	59.24	52.84	58.85	65.88	59.41	56.32	56.19
CMN	25.00	30.38	55.92	62.41	52.86	52.39	61.76	59.83	55.52	60.25	71.13	60.69	56.56	56.13
ICON	22.22	29.91	58.78	64.57	62.76	57.38	64.71	63.04	58.86	63.42	67.19	60.81	59.09	58.54
DialogueRNN	25.69	33.18	75.10	78.80	58.59	59.21	64.71	**65.28**	80.27	**71.86**	61.15	58.91	63.40	62.75
DialogueGCN	40.62	**42.75**	89.14	**84.54**	61.92	**63.54**	67.53	64.19	65.46	63.08	64.18	**66.99**	65.25	**64.18**

Table 3: Comparison with the baseline methods on IEMOCAP dataset; Acc. = Accuracy; bold font denotes the best performances. Average(w) = Weighted average.



(a) (b)

Figure 4: Visualization of edge-weights in Eq. (1) — (a) Target utterance attends to other speaker's utterance for correct context; (b) Short utterance attends to appropriate contextual utterances to be classified correctly.

We surmise that this improvement is a result of the speaker dependent relation modelling of the edges in our graph network which inherently improves the context understanding over DialogueRNN.

5.2 Effect of Context Window

We report results for DialogueGCN model in Tables 3 and 4 with a past and future context window size of (10, 10) to construct the edges. We also carried out experiments with decreasing context window sizes of (8, 8), (4, 4), (0, 0) and found that performance steadily decreased with F1 scores of 62.48%, 59.41% and 55.80% on IEMOCAP. DialogueGCN with context window size of (0, 0) is equivalent to a model with only sequential encoder (as it only has self edges), and performance is expectedly much worse. We couldn't perform extensive experiments with larger windows because of computational constraints, but we expect performance to improve with larger context sizes.

5.3 Ablation Study

We perform ablation study for different level of context encoders, namely sequential encoder and speaker-level encoder, in Table 5. We remove them one at a time and found that the speaker-level encoder is slightly more important in overall performance. This is due to speaker-level encoder mitigating long distance dependency issue of sequential encoder and DialogueRNN. Removing both of them results in a very poor F1 score of 36.7 %, which demonstrates the importance of contextual modelling in conversational emotion recognition.

Further, we study the effect of edge relation modelling. As mentioned in Section 3.3.2, there are total $2M^2$ distinct edge relations for a conversation with M distinct speakers. First we removed only the temporal dependency (resulting in M^2 distinct edge relations), and then only the speaker dependency (resulting in 2 distinct edge relations) and then both (resulting in a single edge relation all throughout the graph). The results of these tests in Table 6 show that having these different relational edges is indeed very important for modelling emotional dynamics. These results support our hypothesis that each speaker in a conversation is uniquely affected by the others, and hence, modelling interlocutors-dependency is rudimentary. Fig. 4a illustrates one such instance where target utterance attends to other speaker's utterance for context. This phenomenon is com-

Methods	AVEC				MELD
	Valence	Arousal	Expectancy	Power	
CNN	0.545	0.542	0.605	8.71	55.02
Memnet	0.202	0.211	0.216	8.97	-
bc-LSTM	0.194	0.212	0.201	8.90	56.44
bc-LSTM+Att	0.189	0.213	0.190	8.67	56.70
CMN	0.192	0.213	0.195	8.74	-
ICON	0.180	0.190	0.180	8.45	-
DialogueRNN	0.168	0.165	0.175	7.90	57.03
DialogueGCN	**0.157**	**0.161**	**0.168**	**7.68**	**58.10**

Table 4: Comparison with the baseline methods on AVEC and MELD dataset; MAE and F1 metrics are user for AVEC and MELD, respectively.

Sequential Encoder	Speaker-Level Encoder	F1
✓	✓	64.18
✓	✗	55.30
✗	✓	56.71
✗	✗	36.75

Table 5: Ablation results w.r.t the contextual encoder modules on IEMOCAP dataset.

Speaker Dependency Edges	Temporal Dependency Edges	F1
✓	✓	64.18
✓	✗	62.52
✗	✓	61.03
✗	✗	60.11

Table 6: Ablation results w.r.t the edge relations in speaker-level encoder module on IEMOCAP dataset.

monly observable for DialogueGCN, as compared to DialogueRNN.

5.4 Performance on Short Utterances

Emotion of short utterances, like "okay", "yeah", depends on the context it appears in. For example, without context "okay" is assumed 'neutral'. However, in Fig. 4b, DialogueGCN correctly classifies "okay" as 'frustration', which is apparent from the context. We observed that, overall, DialogueGCN correctly classifies short utterances, where DialogueRNN fails.

5.5 Error Analysis

We analyzed our predicted emotion labels and found that misclassifications are often among similar emotion classes. In the confusion matrix, we observed that our model misclassifies several samples of 'frustrated' as 'angry' and 'neutral'. This is due to subtle difference between frustration and anger. Further, we also observed similar misclassification of 'excited' samples as 'happy' and 'neutral'. All the datasets that we use in our experiment are multimodal. A few utterances e.g., 'ok. yes' carrying *non-neutral* emotions were misclassified as we do not utilize audio and visual modality in our experiments. In such utterances, we found audio and visual (in this particular example, high pitched audio and frowning expression) modality providing key information to detect underlying emotions (*frustrated* in the above utterance) which DialogueGCN failed to understand by just looking at the textual context.

6 Conclusion

In this work, we present Dialogue Graph Convolutional Network (DialogueGCN), that models inter and self-party dependency to improve context understanding for utterance-level emotion detection in conversations. On three benchmark ERC datasets, DialogueGCN outperforms the strong baselines and existing state of the art, by a significant margin. Future works will focus on incorporating multimodal information into DialogueGCN, speaker-level emotion shift detection, and conceptual grounding of conversational emotion reasoning. We also plan to use DialogueGCN in dialogue systems to generate affective responses.

References

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2017. Quasi-Recurrent Neural Networks. In *International Conference on Learning Representations (ICLR 2017)*.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

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):335–359.

Minghai Chen, Sen Wang, Paul Pu Liang, Tadas Baltrušaitis, Amir Zadeh, and Louis-Philippe Morency. 2017. Multimodal sentiment analysis with word-level fusion and reinforcement learning. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 163–171. ACM.

Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Lun-Wei Ku, et al. 2018. Emotionlines: An emotion corpus of multi-party conversations. *arXiv preprint arXiv:1802.08379*.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, abs/1412.3555.

Niko Colneriĉ and Janez Demsar. 2018. Emotion recognition on twitter: comparative study and training a unison model. *IEEE Transactions on Affective Computing*.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org.

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, New Orleans, Louisiana. Association for Computational Linguistics.

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

Chenyang Huang, Amine Trabelsi, and Osmar R Zaïane. 2019. Ana at semeval-2019 task 3: Contextual emotion detection in conversations through hierarchical lstms and bert. *arXiv preprint arXiv:1904.00132*.

Sidney K. D'Mello, Scotty Craig, Jeremiah Sullins, and Arthur Graesser. 2006. Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue. *I. J. Artificial Intelligence in Education*, 16:3–28.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP 2014*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

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

Bernhard Kratzwald, Suzana Ilic, Mathias Kraus, Stefan Feuerriegel, and Helmut Prendinger. 2018. Decision support with text-based emotion recognition: Deep learning for affective computing. *arXiv preprint arXiv:1803.06397*.

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.

G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder. 2012. The SEMAINE Database: Annotated Multimodal Records of Emotionally Colored Conversations between a Person and a Limited Agent. *IEEE Transactions on Affective Computing*, 3(1):5–17.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Costanza Navarretta, K Choukri, T Declerck, S Goggi, M Grobelnik, and B Maegaard. 2016. Mirroring facial expressions and emotions in dyadic conversations. In *LREC*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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, Vancouver, Canada. Association for Computational Linguistics.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019a. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019b. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

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.

Björn Schuller, Michel Valster, Florian Eyben, Roddy Cowie, and Maja Pantic. 2012. AVEC 2012: The Continuous Audio/Visual Emotion Challenge. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, pages 449–456, New York, NY, USA. ACM.

Carlo Strapparava and Rada Mihalcea. 2010. Annotating and identifying emotions in text. In *Intelligent Information Access*, pages 21–38. Springer.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end Memory Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2440–2448, Cambridge, MA, USA. MIT Press.

Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018a. Memory Fusion Network for Multi-view Sequential Learning. In *AAAI Conference on Artificial Intelligence*, pages 5634–5641.

Amir Zadeh, Paul Pu Liang, Soujanya Poria, Prateek Vij, Erik Cambria, and Louis-Philippe Morency. 2018b. Multi-attention recurrent network for human communication comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5642–5649.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.