Static and Dynamic Speaker Modeling based on Graph Neural Network for Emotion Recognition in Conversation

Anonymous ACL submission

Abstract

Each person has a unique personality which affects how they feel and convey emotions. Hence, speaker modeling is important for the task of emotion recognition in conversation (ERC). In this paper, we propose a novel graph-based ERC model which considers both conversational context and speaker personality. We model the internal state of the speaker (personality) as *Static* and *Dynamic* speaker state, where the *Dynamic* speaker state is modeled with a graph neural network based encoder. Experiments on benchmark dataset shows the effectiveness of our model. Our model outperforms baseline and other graph-based methods. Analysis of results also show the importance of explicit speaker modeling.

1 Introduction

Emotion recognition in conversation (ERC) is a task within the sphere of emotion recognition. ERC aims to predict the emotion of each utterance in a conversation. With the recent advances of dialogue research, ERC has gained popularity due to its potential to support downstream applications such as affective dialog systems (Majumder et al., 2020) and opinion mining from social media chats (Chatterjee et al., 2019).

The emotion of an utterance depends on many factors including surrounding context and speaker personality. Previous studies show that the same utterance can express different emotions under different contexts (Poria et al., 2019b). On the other hand, the speaker's personality and background should be considered when we interpret the emotion of an utterance. For example, in Figure 1, the utterance "This is great!" can carry the emotion of *anger* (sarcastic person) or *joy* (not sarcastic). This difference can be attributed to the different personalities of the speakers.

In speaker modeling, we aim to model the internal state of the speaker. Moreover, we distinguish between the *Static* and *Dynamic* states of a speaker. The *Static* speaker state refers to the average state of a person that remains unchanged over a long period of time. On the other hand, the *Dynamic* speaker state refers to the deviation from the *Static* state in presence of external stimuli. External stimuli can dictate and change the speaker's internal state, which in turn affects the emotion displayed by an individual, hence modeling the *Dynamic* state of a speaker is important for ERC.

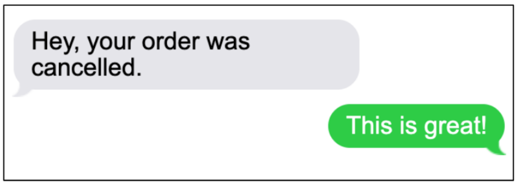

Figure 1: The emotion conveyed by the phrase "This is great" can either be *anger* (sarcasm) or *joy* (in the case that the person ordered the wrong item). This example is taken from (Poria et al., 2019b).

In the past few years, Graph Neural Networks (GNNs) have been used increasingly for ERC. GNNs provide an intuitive way to model conversations (Shen et al., 2021) given the inherent structural flexibility of the graph. The graph structure can be used to capture the dependency between utterances and speakers.

Recent works such as DialogGCN (Ghosal et al., 2019), RGAT (Ishiwatari et al., 2020), EmoBERTa (Kim and Vossen, 2021) and DAG-ERC (Shen et al., 2021) have modelled conversational contexts using various methods, however they do not model speaker state explicitly. Whereas ConGCN (Zhang et al., 2019) and MMGCN (Hu et al., 2021) models the speaker state explicitly, however, they use random embedding for initialization and model just the *Static* aspect.

In this study, we propose a novel graph-based ERC model which considers both *Static* and *Dynamic* aspects of speaker state. We utilize a graph

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, pages 247 - 253
July 10-15, 2022 ©2022 Association for Computational Linguistics

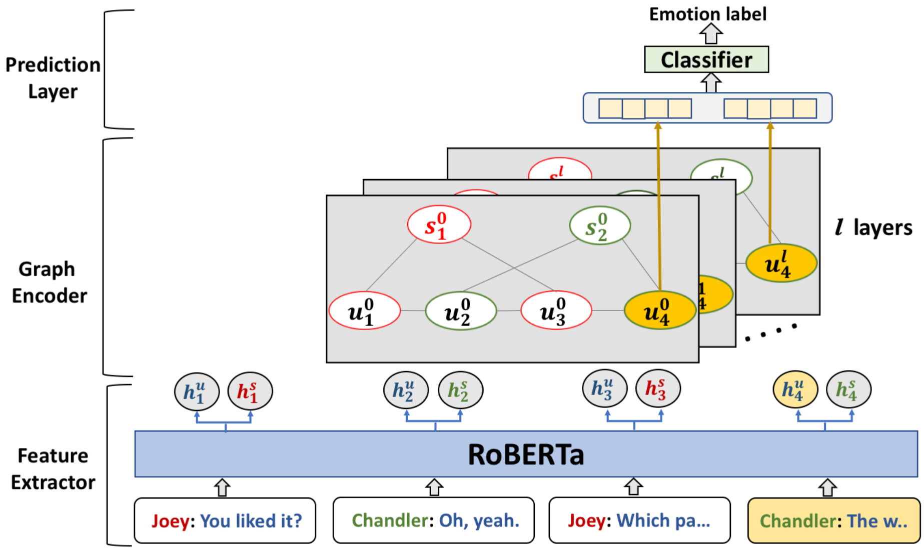

Figure 2: Model overview. The target utterance is denoted in yellow color.

which includes past utterance nodes and explicit speaker nodes to model the interactions between utterances and speakers in the dialogue. Experimental results on the benchmark MELD dataset (Poria et al., 2019a) verified the effectiveness of our model regarding both context and speaker modeling.

2 Related Work

DialogGCN (Ghosal et al., 2019) was the first paper to use GNN to model dialogues. Given an input dialogue, a complete graph within a fixed context (past and future) window is built. Since graph-based neural networks do not take sequential information into account, RGAT (Ishiwatari et al., 2020) uses relational positional encodings to improve upon DialogGCN. DAG-ERC (Shen et al., 2021) built a more intuitive graph structure by considering local and remote information, without using any future utterance.
EmoBERTa (Kim and Vossen, 2021) modeled the speaker state and context by prepending the speaker names to utterances and inserting separation tokens between the utterances in a dialogue, and feeding it to RoBERTa. ConGCN (Zhang et al., 2019) explicitly used speaker nodes, which were initialized randomly. MMGCN (Hu et al., 2021) also incorporated randomly initialized speaking embeddings in their model.

3 Methodology

Our model consists of three components: Feature extractor, Graph encoder, and Prediction layer. Figure 2 shows an overview of our proposed model. We will give a detailed explanation of our model in this section.

3.1 Problem Definition

In ERC, a dialogue is defined as a sequence of utterances $\{U_1, U_2, ..., U_N\}$, where N is the number of utterances. Each utterance U_i is spoken by a speaker S_i and has an emotion label Y_i. The goal of ERC is to predict the emotion label Y_t for a given U_t and S_t .

3.2 Feature Extractor

We use pretrained RoBERTa (Liu et al., 2019) as our feature extractor. Inspired by EmoBERTa (Kim and Vossen, 2021), we feed the following sequence to RoBERTa for each utterance U_i with speaker S_i (as shown in Figure 2):

$$[CLS]S_i : U_i[SEP] \tag{1}$$

For each utterance U_i, we take the output vector of RoBERTa corresponding to the [CLS] token as the **utterance embedding** h_i^u. In addition, we extract the RoBERTa output vector corresponding to the speaker token[1] S_i as the **speaker embedding** h_i^s. This component is responsible for the *Static* speaker state modeling and h_i^s represents the *Static* speaker state.

3.3 Graph Encoder

In this section, we introduce the construction of a dialogue graph and the details of the graph encoder.

[1]In the case when speaker name is a multi-token entity, we consider the first token for the speaker embedding.

3.3.1 Graph Construction

For a target utterance U_t in the dialogue, we build a graph $G = (V, E)$ to model the surrounding context and speaker information, where V denotes the set of nodes and E is the set of edges.

The graph G contains two types of nodes:

- *Utterance node:* We consider the target utterance U_t and up to w utterances preceding U_t as past utterances.
- *Speaker node:* We consider the unique speakers of the target and past utterances.

The set of nodes can be represented as:

$$V = \{U_i\}_{i=t-w}^{i=t} \cup \text{Uniq}(\{S_i\}_{i=t-w}^{i=t}) \tag{2}$$

where the function Uniq() returns all the unique elements in a set.

Our graph contains two types of edges:

- *Utterance-Utterance Edge:* We connect each utterance to its previous utterance. These model the effect of past utterance on the present utterance. These are given by $E_{uu} = \{(U_{i-1}, U_i)\}_{i=t-w+1}^{i=t}$
- *Utterance-Speaker Edge:* We connect each utterance U_i to its corresponding speaker S_j. The set of utterance-speaker edges are denoted as $E_{us} = \{(U_i, S_j)\}_{i=t-w}^{i=t}$. These edges model the effect of speakers on the utterances.

The set of edges can be given by:

$$E = E_{uu} \cup E_{us}, \tag{3}$$

Figure 2 (Graph Encoder part) illustrates an example of the constructed graph with a target utterance U_4 (colored in yellow) and 3 past utterances. U_1 and U_3 are spoken by a unique speaker S_1, while U_2 and U_4 are spoken by another unique speaker S_2. (Note that the subscripts of the speakers reflects the indices after Uniq().)

3.3.2 Node Initialization

We initialize the Utterance and Speaker nodes as follows:

- *Utterance node* : $u_i^0 = h_i^u \quad \forall i \in [t-w, t]$
- *Speaker node* : $s_j^0 = avg(h_i^s) \quad \forall i$ spoken by S_j.

Since there is only one speaker node for each unique speaker, we use the averaged speaker embeddings to initialize the Speaker node.

3.3.3 GNN-Based Graph Encoding Layers

After constructing and initializing the graph, we feed it to the GNN-based encoding layers, which update node representations considering the graph structure. This component is responsible for the *Dynamic* speaker state modeling.

We use l-layered GNN to get the updated node representations based on the graph structure of G. For k^{th} layer, all the nodes (Speaker and Utterance nodes) are updated considering each of their direct neighbours:

$$(\{u_i^k\}, \{s_j^k\}) = GNN^k(\{u_i^{k-1}\}, \{s_j^{k-1}\}) \tag{4}$$

After being updated by l layers, the *Static* speaker state, s_j^0, is updated to s_j^l, which represents the *Dynamic* speaker state. Similarly, the initial utterance embedding u_i^0 is updated to final utterance embedding u_i^l.

3.4 Emotion Classification

Finally, we concatenate the initial and the final utterance embeddings of target utterance and feed it through a feed-forward network to classify emotions.

$$P_t = \text{softmax}(FFN(u_t^0 || u_t^l)), \tag{5}$$

$$Y_t^* = \text{argmax}(P_t), \tag{6}$$

Here, $||$ denotes the concatenation operation, FFN is the feed-forward neural network layer, and P_t is the probability distribution for the predicted emotion.

3.5 Training Objective

We use the standard cross-entropy along with L2-regularization as the loss ($\mathcal{L}$):

$$\mathcal{L} = -\sum_{x=1}^{M} \sum_{t=1}^{N_x} \log P_{x,t}[Y_{x,t}] + \lambda ||\theta||_2, \tag{7}$$

Here, M is the total number of training dialogues, N_x is the number of utterances in the x^{th} dialogue, $P_{x,t}$ and $Y_{x,t}$ are the predicted probability distribution of emotion labels and the truth label respectively for utterance t of the dialogue x. λ is the L2-regularization weight, and θ is the set of all trainable parameters.

	Train	Dev	Test
# Utterance	9,989	1,109	2,610
# Dialogue	1,039	114	280

Table 1: Statistics for the MELD dataset.

4 Experiments and Results

Experiments on the benchmark dataset shows the effectiveness of our model. Details of experiments and analysis are given in this section.

4.1 Dataset

We evaluate our model on the benchmark Multimodal EmotionLines Dataset (MELD) dataset (Poria et al., 2019a). MELD is a multi-modal dataset collected from the TV show Friends. There are 7 emotion labels: neutral, happiness, surprise, sadness, anger, disgust, and fear. Since this is an imbalanced dataset, weighted-F1 is used as the evaluation metric. More than 85% of the utterances in MELD are spoken by 6 main speakers, this high utterance per speaker is useful for modeling the speaker state. The statistics of MELD are shown in Table 1.

4.2 Experimental Settings

The feature extractor used is the pre-trained RoBERTa-large (Liu et al., 2019). The size of all the hidden features is 1024. We experiment with Graph Convolutional Network(GCN) (Kipf and Welling, 2017) and Graph Attention Network(GAT) (Veličković et al., 2018) as the GNN-based graph encoding layers. For the GCN based model, the past context is set to be 3 utterances and the number of GNN layers was set to be 2. For the GAT based model, the past context is set to be 5 utterances and the number of GNN layers was set to be 3. GAT model also has three attention heads in addition to the above settings.

The models are trained for 10 epochs, batch size is set to be 8, and the learning rate is set to 1e-6. The model with the highest weighted-F1 on the validation set is selected for evaluation. Due to the stochastic nature of the model, we report the averaged score of 3 random runs on the test set.

4.3 Evaluation

Compared Methods and Results: We compare our proposed model with baselines and previous works. The results are reported in Table 2.

First, we establish two baselines: *RoBERTa (no context)* and *RoBERTa (w/ modified input)*. In the *RoBERTa (no context)* utterance alone is used as input to the pre-trained RoBERTa model. In the *RoBERTa (w/ modified input)* we use a modified input as given by Equation 1. Our proposed method outperforms both RoBERTa baselines by F1 scores of 2.4 and 1.8, respectively. This shows the advantage of using the graph encoding mechanism.

Next, we compare our model with other GNN-based models: *DAG-ERC*, *DialogGCN* and *RGAT*. For fair comparison, we use the models which use RoBERTa-large as the feature extractor[2]. Our model outperforms all these models, proving the advantage of using explicit speaker nodes to model conversations.

Finally, we compare our results with the *EmoBERTa* model[3]. Our model with GCN encoder performs slightly worse than EmoBERTa. However, our model with GAT encoder outperforms EmoBERTa. Hence, we can state that the performance of our model and EmoBERTa is comparable. Note that EmoBERTa uses both past and future utterances as context, whereas we only use the past utterances as context, which is more natural as conversations proceed with time and future utterances cannot be used for real-time applications. Under the condition that only the past utterances are allowed, both our proposed models outperform *EmoBERTa (wo/ future context)*.

GCN vs. GAT: In our experiments, models which utilize GAT as graph encoders outperformed the GCN ones. The edge weights for all edges in our GCN models were set to be 1. On the other hand, the edge weights for GAT models were learned and optimized during the training of our model due to the explicit attention heads of the GAT based models.

We speculate that since the utterance-utterance edge and speaker-utterance edge are different in nature so their edge weight should be different, hence GAT outperformed GCN and has the ability to better represent the relations between nodes.

Since, GAT based model performs superior to GCN based one, we use GAT based models for further analysis.

[2] The authors of DAG-ERC re-implement DialogGCN and RGAT using RoBERTa-large as feature extractor, we include the scores reported by the DAG-ERC paper.

[3] EmoBERTa was the SOTA model while this research was conducted, the new SOTA model is EmotionFlow. (https://github.com/fpcsong/emotionflow/blob/master/EmotionFlow.pdf)

Model	Weighted-F1
RoBERTa (no context)	0.635
RoBERTa (w/ modified input)	0.641
DAG-ERC	0.636
RGAT (+RoBERTa)	0.628
DialogueGCN (+RoBERTa)	0.630
EmoBERTa	0.656
EmoBERTa (wo/ future context)	0.646
Proposed (GCN)	0.652
Proposed (GAT)	**0.659**

Table 2: Experimental results on MELD.

Method	Weighted-F1
Proposed (Static + Dynamic)	**0.658**
Proposed (wo/ speaker) (Static)	0.646
Proposed (random init. speaker)	0.638

Table 3: Impact of speaker modeling.

4.4 Analysis

In this section, we conduct various analysis of our proposed model.

4.4.1 Impact of Speaker Modeling

To investigated the impact of the speaker modeling on the performance, we evaluated our model by removing speaker nodes, *Proposed (wo/ speaker)*, and by randomly initializing speaker nodes, *Proposed (random init. speaker)*. The results are shown in Table 3. These results are with three past context and two GAT layer model.

Removing speaker nodes reduces the weighted-F1 score by 1.2. The significant decrease indicates the importance of speaker modeling to the ERC task. Whereas, randomly initializing speaker nodes results in a performance drop of 2.0 points. Moreover, the score with random speaker initialization is lower than the score of the model without any speaker nodes. We hypothesize that the random embeddings create noise and hinder the performance.

4.4.2 Impact of Context Window Size and the Number of GAT layers

To analyze the impact of context window size, we varied the past context window size from 1 to 5. The results are reported for two and three GAT layers in Figure 3. The model performs worst when we use only one past context, which illustrates the necessity to model sufficient context. Moreover, we also find out that the optimal number of past context varied for different number of GNN layers (3 context for 2 layers and 5 context for 3 layers).

Next, we investigated the effect of changing the number of layers on the performance. One layer of graph encoder updates a node considering all the one-hop neighbours. The scores for the number of layers from two to five for a past context of size five is given in the Figure 4. The score is highest for three layers. Our graph structure allows information to be aggregated from the last context utterance in few hops due to utterances being connected by speaker nodes, so the performance does not change greatly by changing the number of layers.

4.4.3 Case Study

We performed a qualitative analysis for our model. We used the model with five past contexts and three GAT layers. We manually inspected ten test samples that were predicted correctly and ten instances that were predicted incorrectly.

We found that utterances with speakers other than the six main speakers have a higher chance of being predicted incorrectly (six out of ten incorrectly predicted test samples contained at least one speaker other than the main speakers). We speculate that this can be attributed to the fact that we only modeled the main six speakers, and for the case of other speakers, we did not construct any speaker nodes. In the first sample given in Table 4 it is noted that a non-main speaker (Steve) accounts for a considerable part of the dialogue and our system predicts the emotion incorrectly.

However, in the cases in which the main speakers make up the majority of the past context, the emotion of utterances of other speakers can be predicted correctly. The second sample in Table 4 shows this, where the emotion label for the dialogue of a non-main speaker (Fireman #1) is predicted correctly. The reason might be that the speaker nodes of the main speakers assist the model in predicting the emotion label.

5 Conclusion

We proposed a novel graph-based method to model speaker states explicitly for the task of ERC. Experiments showed that our model outperforms baselines and other graph-based models. We analyse the impact of speaker modeling and show that both *Static* speaker state and *Dynamic* speaker state modeling are important for the accurate prediction of emotions in ERC. In addition, we investigate the

Dialogue	Predicted	Gold
Steve: Oh, okay, I get it.		
Ross : No wait, look. Look! I'm sorry, it's just I've never even		
Steve: Howard's the,		
Ross: Yes but too me he's just, man.		
Steve : Okay, fine, whatever. Welcome to the building.	**neutral**	**anger**
Phoebe: Oh!		
Rachel : My God!		
Joey: Hey buddy, do you think I can borrow your uniform this Thursday?		
Fireman #1: Excuse me?	**surprise**	**surprise**

Table 4: Case study. The target utterance is shown in italics.

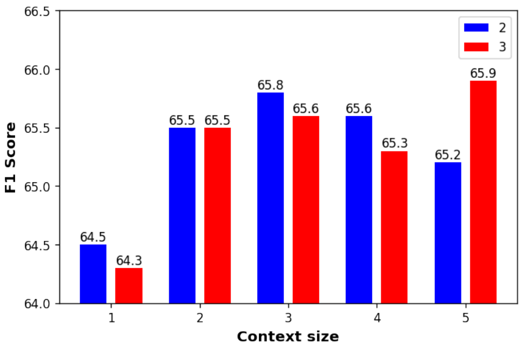

Figure 3: Impact of past context size with two and three GAT layers.

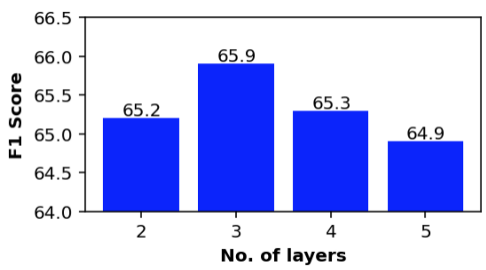

Figure 4: Impact of number of GAT layers. Context window is of size 5.

effect of changing the number of GNN layers and the past context on the performance of our model.

References

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 154–164, Hong Kong, China. Association for Computational Linguistics.

Jingwen Hu, Yuchen Liu, Jinming Zhao, and Qin Jin. 2021. MMGCN: Multimodal fusion via deep graph convolution network for emotion recognition in conversation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5666–5675, Online. Association for Computational Linguistics.

Taichi Ishiwatari, Yuki Yasuda, Taro Miyazaki, and Jun Goto. 2020. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7360–7370, Online. Association for Computational Linguistics.

Taewoon Kim and Piek Vossen. 2021. Emoberta: Speaker-aware emotion recognition in conversation with roberta.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Navonil Majumder, Pengfei Hong, Shanshan Peng, Jiankun Lu, Deepanway Ghosal, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. MIME: MIMicking emotions for empathetic response generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8968–8979, Online. 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.

Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 2021. Directed acyclic graph network for conversational emotion recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1551–1560, Online. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks.

Dong Zhang, Liangqing Wu, Changlong Sun, Shoushan Li, Qiaoming Zhu, and Guodong Zhou. 2019. Modeling both context- and speaker-sensitive dependence for emotion detection in multi-speaker conversations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5415–5421. International Joint Conferences on Artificial Intelligence Organization.