Graph Neural Network based Speaker Modelling for Emotion Recognition in Conversation

Prakhar Saxena, Yin Jou Huang, Sadao Kurohashi
Kyoto University
{prakhar,huang,kuro}@nlp.ist.i.kyoto-u.ac.jp

Abstract

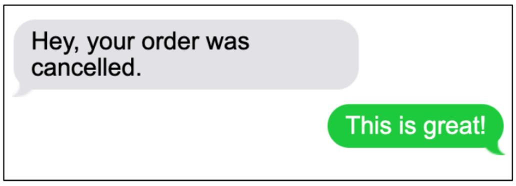
Each person has a unique personality which affects the way they feel and convey emotions. Hence, speaker modeling is important for the task of emotion recognition in conversation (ERC). In this paper, we propose a novel graph-based ERC model which considers both conversational context and speaker personality. Our model outperforms other graph-based models and achieves a performance comparable to the current state-of-the-art model.

1 Introduction

Emotion recognition in conversation (ERC) is a task within the sphere of emotion recognition. The goal of ERC is 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 building affective dialog systems [1] and opinion mining from social media chats [2].

The emotion of an utterance depends on many factors including surrounding context and speaker personality. The same utterance can express different emotions under different contexts. On the other hand, the speaker's personality and background may affect how we interpret the emotion of an utterance. For example, in Figure 1, the utterance "This is great!" can carry either negative sentiment (sarcastic person) or positive sentiment (not sarcastic). This difference can be attributed to the different personalities of speakers.

In speaker modeling, 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 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.



Hey, your order was cancelled.

This is great!

Figure 1 The emotion conveyed by the phrase "This is great" can either be negative (sarcasm) or positive (in the case that the person ordered the wrong item). This example is taken from [10].

In the past few years, Graph Neural Networks (GNNs) have been used increasingly in ERC. GNNs provide an intuitive way to model conversations [3] 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 [4], RGAT [5], EmoBERTa [6] and DAG-ERC [3] have modelled conversational contexts using various methods, however they do not model speaker state explicitly. Whereas ConGCN [7] and MMGCN [8] 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 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 [9] verified the effectiveness of our model regarding both context and speaker modeling.

2 Related Work

DialogGCN [4] 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.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

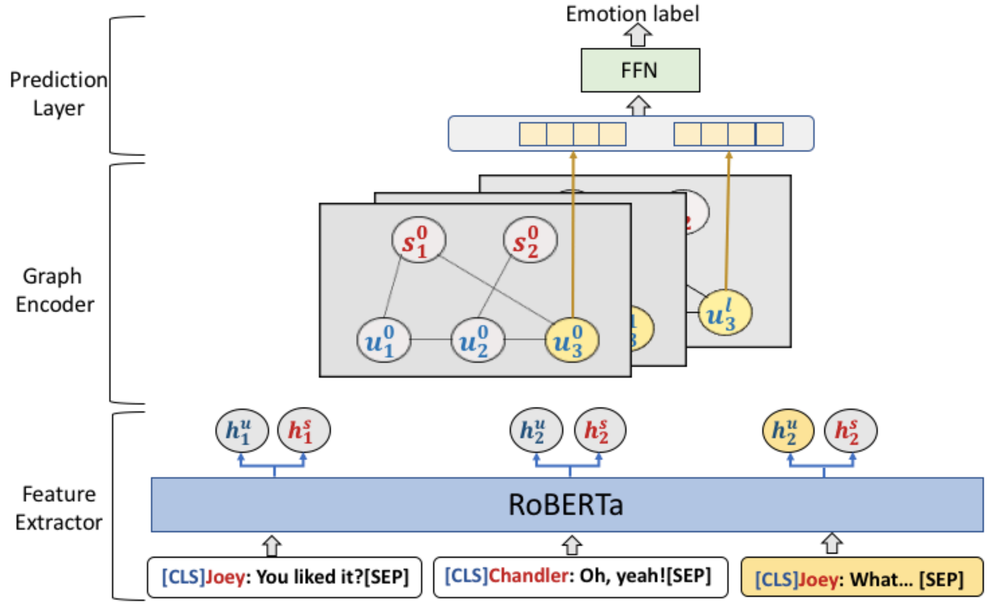


Figure 2 Model overview . The target utterance is denoted in yellow color.

Since graph-based neural networks do not take sequential information into account, RGAT [5] uses relational positional encodings to improve upon DialogGCN. DAG-ERC [3] builds a more intuitive graph structure by considering local and remote information, without using any future utterance.

EmoBERTa [6] models 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 [7] explicitly uses speaker nodes, which are initialized randomly. MMGCN [8] also incorporate randomly initialized speaking embeddings in their model.

3 Methodology

3.1 Problem Definition

In ERC, a conversation 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 utterance U_t .

3.2 Model Overview

Our model consists of three components: Feature extractor, Graph encoder, and Prediction layer. Figure 2 shows the overview of our proposed model.

3.3 Feature Extraction

We use pretrained RoBERTa [11] as our feature extractor. Inspired by EmoBERTa [6], 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 S_i[1)] as **speaker embedding** h_i^{sp}.

3.4 Graph Encoder

3.4.1 Graph Construction

For a target utterance U_t in the dialogue, we build a graph $G = (V, E)$ to model the information flow in a conversation, 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.

[1)] We consider the first token after [CLS] as the speaker embedding.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

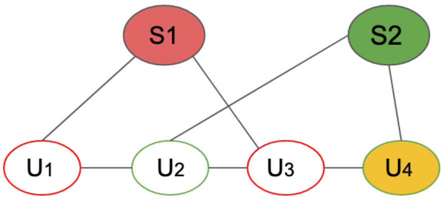


Figure 3 Graph Structure.

The set of nodes can be represented as:

$$V = \{U_j\}_{j=t-w}^{j=t} \cup \mathrm{Uniq}(\{S_j\}_{j=t-w}^{j=t}) \tag{2}$$

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

Our graph contains two types of edges, given by:

- **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 $E_{uu} = \{(U_{j-1}, U_j)\}_{j=t-w+1}^{j=t}$
- **Utterance-Speaker Edge:** We connect each utterance U_j to its corresponding speaker S_k. The set of utterance-speaker edges are denoted as $E_{us} = \{(U_j, S_k)\}_{j=t-w}^{j=t}$. 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 3 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 speaker S_1, while U_2 and U_4 are spoken by S_2.

3.4.2 Node Initialization

We initialize the utterance and speaker nodes as follows:

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

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

3.4.3 GCN-based graph encoding layers

After constructing and initializing the graph, we feed it to the Graph Convolutional Network (GCN) [12] based encoding layers, which update node representation considering the graph structure.

For each GCN layer, the layer-wise propagation rule is:

$$H^{l+1} = \sigma(A^* H^l W^l) \tag{4}$$

where:

- H^l : Matrix for layer l, with all the node embeddings row wise, of size $N \times D$. (N: number of nodes, D: embedding size)
- $A^* = A + I$, A is adjacency matrix and I is identity matrix.
- W^l, are the weights for l-th layer.

We use GCN to get the updated representation of the nodes in G. After being processed by L layers of GCN, the final utterance and speaker node representations are denoted as: u_i^l and s_j^l. s_j^l models the dynamic speaker state under the current dialog context.

3.5 Emotion Classification

Finally, we concatenate the initial and the final node embedding (embedding after the L-th GCN layer) of target utterance and feed it through a feed-forward network to classify emotions.

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

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

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

3.6 Training Objective

We use the standard cross-entropy along with L2-regularization as the measure of loss (ℓ):

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

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

4 Experiment

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

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

Table 1 Statistics for the MELD dataset.

4.1 Dataset

We evaluate our model on the benchmark Multimodal EmotionLines Dataset (MELD) dataset [9]. MELD is a multi-modal dataset collected from the TV show Friends. There are 7 emotion labels including neutral, happiness, surprise, sadness, anger, disgust, and fear. Since this is an imbalanced dataset, weighted F1 is used as the evaluation metric. The statistics of MELD are shown in Table 1.

4.2 Experimental Settings

The feature extractor used is RoBERTa-large[11] The model is trained for 10 epochs, batch-size is set to be 8, and the learning rate is set at 1e-6. The model with the highest weighted F1 on the validation set is selected for evaluation. The past context is set to be 3 utterances and the number of GCN layers is set to be 2. The size of hidden features is 1024. Also, we report the average score of 3 random runs on the test set.

5 Results and Analysis

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

First, we compare our model to the baseline RoBERTa models. In *RoBERTa (no context)*, the utterance alone is used as input to the pretrained RoBERTa model. In *RoBERTa (w/ modified input)*, we uses inputs as given by Equation 1. Our proposed method performs significantly better than both RoBERTa baselines. This shows the advantage of 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 as the feature extractor. The authors of DAG-ERC re-implement DialogGCN and RGAT using RoBERTa as feature extractor, we use the scores reported by the DAG-ERC paper. Our model outperforms all these models, proving the advantage of using explicit speaker nodes to model conversations.

Finally, we compare our results with the state-of-the-art model, *EmoBERTa*. Our model achieves comparable performance with EmoBERTa. However, EmoBERTa uses both past and future utterances as context, whereas we only use the past utterances as context.Under the condition that the past utterances are allowed, the proposed model outperforms *EmoBERTa (w/o future context)*.

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 (w/o future context)	0.646
Proposed	0.652

Table 2 Comparison with other models.

Method	Weighted F1
Proposed	**0.652**
Proposed (w/o speaker nodes)	0.646
Proposed (speaker node w/ random .init)	0.644

Table 3 Impact of speaker modeling.

Impact of speaker modeling To investigate the impact of the speaker modeling on the performance, we evaluate our model by removing speaker nodes and by randomly initializing speaker nodes. The results are shown in Table 3. Removing speaker nodes reduces the performance significantly, which confirms our hypothesis that, modelling speaker states is important. Whereas randomly initializing speaker nodes results in a performance which is comparable to using utterance nodes only.

6 Conclusion

We proposed a novel idea of modeling speaker states explicitly using a graph for emotion recognition in conversation (ERC). Experiments showed that our model achieves comparable results with the state-of-the-art model and outperforms other graph-based models. In addition, empirical results illustrated the effectiveness of explicit speaker modeling.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

References

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

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

[3] Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 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)**, pp. 1551–1560, Online, August 2021. Association for Computational Linguistics.

[4] Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 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)**, pp. 154–164, Hong Kong, China, November 2019. Association for Computational Linguistics.

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

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

[7] Dong Zhang, Liangqing Wu, Changlong Sun, Shoushan Li, Qiaoming Zhu, and Guodong Zhou. 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**, pp. 5415–5421. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[8] Jingwen Hu, Yuchen Liu, Jinming Zhao, and Qin Jin. Mmgcn: Multimodal fusion via deep graph convolution network for emotion recognition in conversation, 2021.

[9] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. Meld: A multimodal multi-party dataset for emotion recognition in conversations, 2019.

[10] Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. Emotion recognition in conversation: Research challenges, datasets, and recent advances, 2019.

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

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

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).