Conversational Emotion Recognition Using Self-Attention Mechanisms and Graph Neural Networks

Zheng Lian\textsuperscript{1,3}, Jianhua Tao\textsuperscript{1,2,3}, Bin Liu\textsuperscript{1}, Jian Huang\textsuperscript{1,3}, Zhanlei Yang\textsuperscript{4}, Rongjun Li\textsuperscript{4}

\textsuperscript{1}National Laboratory of Pattern Recognition, CASIA, Beijing
\textsuperscript{2}CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing
\textsuperscript{3}School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing
\textsuperscript{4}Huawei Technologies Co., LTD., Beijing

\{zheng.lian, jhtao, liubin, jian.huang\}@nlpr.ia.ac.cn, \{yangzhanlei1, lirongjun3\}@huawei.com

Abstract

Different from the emotion estimation in individual utterances, context-sensitive and speaker-sensitive dependences are vitally pivotal for conversational emotion analysis. In this paper, we propose a graph-based neural network to model these dependences. Specifically, our approach represents each utterance and each speaker as a node. To bridge the context-sensitive dependence, each utterance node has edges between immediate utterances from the same conversation. Meanwhile, the directed edges between each utterance node and its speaker node bridge the speaker-sensitive dependence. To verify the effectiveness of our strategy, we conduct experiments on the MELD dataset. Experimental results demonstrate that our method shows an absolute improvement of 1\%\textendash{}2\% over state-of-the-art strategies.

Index Terms: deep learning, conversational emotion recognition, self-attention mechanism, graph neural networks.

1. Introduction

Conversational emotion recognition is an important research topic due to its potential applications in many tasks, such as dialogue generation [1, 2, 3], social media analysis [4, 5, 6] and intelligent systems [7, 8, 9]. The task of conversational emotion recognition requires understanding the way that humans express their emotions during conversations. Despite its importance, conversational emotion recognition is a complex task due to the following challenges: (1) Since frame-level features contain the temporal dynamics information, the first challenge is how to effectively extract utterance-level features from these frame-level features. (2) Since context-sensitive and speaker-sensitive dependences are vitally important for conversational emotion recognition [10, 11], the second challenge is how to effectively model these dependences in conversations.

The key challenge in emotion recognition is how to learn a good utterance-level representation that captures temporal dynamics from frame-level features. Previous works [12, 13] applied statistic functions (e.g., mean), mapping frame-level features into utterance-level features. However, these approaches roughly consider global information and ignore temporal dynamics of feature sequences. To address these shortcomings, researchers rely on sequence models that can capture temporal dynamics [14, 15], such as recurrent neural networks (RNNs) and its variations (long-short term memory (LSTM) [16] and gated recurrent unit (GRU) [17]). Recently, self-attention mechanism [18] has been verified to capture longer temporal dynamics than typical RNN-based models [18, 19]. It provides an opportunity for injecting the global context information into each input. Inspired by its success, we propose to use this mechanism for utterance-level feature extraction in this paper.

Besides utterance-level feature extraction process, modeling context-sensitive and speaker-sensitive dependences remains an active research topic for conversational emotion recognition [20, 21]. Recently, a graph neural networks (GNNs) based method [22] has been proposed, and achieved promising results on conversational emotion recognition. This method leverages context-sensitive and speaker-sensitive dependences by modeling the conversation using a directed graph. The nodes in the graph represent individual utterances. The edges between a pair of nodes represent the dependency between the speakers of those utterances, along with their relative positions in the conversation. On this basis, the entire conversational corpus can be symbolized as a large heterogeneous graph and the emotion detection task can be recast as a classification problem of the utterance nodes in the graph. However, if there are \(M\) distinct speakers in a conversation, there can be a maximum of \(2M^2\) distinct relation types in the graph [22]. Therefore, this graph structure causes each relation type cannot be fully learned when \(M\) is large, thus leading to performance degradation.

To address these difficulties, we propose to use a relation reduction process in the graph. Concretely, in addition to utterance nodes, we also use speaker nodes compared with [22]. To bridge the context-sensitive dependence, each utterance node has edges with the immediate utterance of the past, and the immediate utterance of the future. And we use two relation types to model both directions. To bridge the speaker-sensitive dependence, there are directed edges between each utterance node and its speaker node, and we use another relation type for these edges. Totally, we only need to model three kinds of relation types. We observe that our relation reduction process can improve the performance of conversational emotion recognition.

The main contributions of this paper include three aspects: 1) We apply the self-attention mechanism for utterance-level feature extraction, since this mechanism can capture longer temporal dynamics that typical RNN-based models [18, 19]; 2) We propose to use the relation reduction process in the graph, thus improving the performance of conversational emotion recognition; 3) Experimental results on the popular benchmark datasets MELD demonstrate that our method gains an absolute improvement of 1\%\textendash{}2\% over state-of-the-art strategies.

The remainder of this paper is organized as follows: In Section 2, we formalize the problem statement and describe our proposed method. Section 3 presents the experimental datasets, setup, results and analysis in detail. Finally, we give a conclusion of the proposed work in Section 4.
2. Proposed Method

2.1. Problem Definition

Suppose we have a conversation \( U = \{u_1, u_2, \ldots, u_N\} \), where \( N \) is the total number of utterances. And there are \( M \) speakers \( p_1, p_2, \ldots, p_M \) (\( M \geq 2 \)). Each utterance \( u_i \) is uttered by one speaker \( p_s(u_i) \), where the function \( s(\cdot) \) maps the index of the utterance into its corresponding speaker. The task is to predict the emotion label for each utterance in the conversation.

2.2. Utterance-level Feature Encoding via Self-attention

In this section, we propose to use self-attention mechanism [18] for utterance-level feature encoding. As shown in Figure 1, we assume the input sequence as \( x_{\alpha} \in \mathbb{R}^{T_{\alpha} \times d_{\alpha}} \), for modality \( \alpha \) (where modality \( \alpha \) can be either acoustic or lexical modality). \( T_{\alpha} \) and \( d_{\alpha} \) represent sequence length and feature dimensions, respectively. To learn the temporal contexts between the adjacent frames, we feed \( x_{\alpha} \) into a 1-dimensional convolutional layer (Conv1D). To take the order of sequence into account, we inject triangle positional embeddings [18] into each frame. The outputs of the last block as \( z_{\alpha} \in \mathbb{R}^{T_{\alpha} \times d_{\alpha}} \) and fusion representations \( g_{\alpha} \in \mathbb{R}^{1 \times d} \) are calculated as follows:

\[
\alpha_{\text{att}} = \text{softmax}(z_{\alpha}W_z) \tag{1}
\]

\[
g_{\alpha} = \alpha_{\text{att}}z_{\alpha} \tag{2}
\]

where \( W_z \in \mathbb{R}^{d \times 1} \) is the trainable parameter.

2.3. Speaker-level Context Encoding via GNNs

In this section, we propose to use GNNs for context-sensitive and speaker-sensitive modeling.

2.3.1. Graph Construction

A graph can be defined as \( G = \{V, E, W, R\} \). \( V \) denotes the set of nodes and \( E \) denotes the set of edges connecting these nodes. \( W \) and \( R \) represent weights and relation types of edges.

**Nodes:** As shown in Figure 1, the graph contains two kinds of nodes: utterance nodes and speaker nodes. We need to generate node representations \( h_i \) for each node. (1) Utterance nodes: As for unimodal settings, we generate representations by feeding acoustic features (or lexical features) into the utterance-level feature encoding module in Section 2.2. As for multimodal settings, to focus on important modalities, we generate representations via the attention-based fusion strategy in [24]. Specifically, we first compute the weights of different modalities via attention mechanisms. The weighted average results are utilized as the multimodal representations for utterance nodes. (2) Speaker nodes: To capture speaker characteristics, we extract representations for speaker nodes using the pre-trained speaker verification system, known as x-vector [25].

**Edges with relations:** We use edges to model the context-sensitive and speaker-sensitive dependences in the conversation. (1) The context-sensitive dependence is represented by directed edges between two utterances nodes from the same conversation. (2) The speaker-sensitive dependence is represented by the directed edge between an utterance node and its speaker node. And we use another relation type for these edges.

As shown in Figure 1, we assume \( u_i \) and \( u_{i+1} \) are immediate utterances from the same conversation. \( p_s(u_i) \) and \( p_s(u_{i+1}) \) are their corresponding speakers, respectively. Our graph has edges between \( u_i \) and \( u_{i+1} \) in both directions with different relation types. To model both directions in the directed graph, we use two relation types. (2) The speaker-sensitive dependence is represented by the directed edge between an utterance node and its speaker node. And we use another relation type for these edges.

As shown in Figure 1, we assume \( u_i \) and \( u_{i+1} \) are immediate utterances from the same conversation. \( p_s(u_i) \) and \( p_s(u_{i+1}) \) are their corresponding speakers, respectively. Our graph has edges between \( u_i \) and \( u_{i+1} \) in both directions with different relation types. To model both directions in the directed graph, we use two relation types. (2) The speaker-sensitive dependence is represented by the directed edge between an utterance node and its speaker node. And we use another relation type for these edges.

As shown in Figure 1, we assume \( u_i \) and \( u_{i+1} \) are immediate utterances from the same conversation. \( p_s(u_i) \) and \( p_s(u_{i+1}) \) are their corresponding speakers, respectively. Our graph has edges between \( u_i \) and \( u_{i+1} \) in both directions with different relation types. To model both directions in the directed graph, we use two relation types. (2) The speaker-sensitive dependence is represented by the directed edge between an utterance node and its speaker node. And we use another relation type for these edges.

2348
weight determination strategies. (1) As for the context-sensitive dependence, we need to determine weights between utterance nodes. Different from previous works that predetermined weights using distant functions and rules [26], we attempt to learn optimal weights via attention mechanisms. Concretely, as for the utterance \( u_i \), it has edges with \( u_{i-1} \) and \( u_{i+1} \). \( h_{i-1} \in \mathbb{R}^{1 \times d} \), \( h_i \in \mathbb{R}^{1 \times d} \), and \( h_{i+1} \in \mathbb{R}^{1 \times d} \) represent node representations of \( u_{i-1} \), \( u_i \), and \( u_{i+1} \), respectively. To calculate weights for these edges, we linearly project the weights between utterance node pairs into a softmax classifier for emotion recognition. During training, we attempt to learn optimal weights via attention mechanisms. Concretely,\[ \alpha_{\text{weight}} \in \mathbb{R}^{1 \times 2}, \]where \( W_h \in \mathbb{R}^{d \times d} \) is the trainable parameter.

(2) As for the speaker-sensitive dependence, we need to determine the weight between the utterance node and its corresponding speaker node. Considering the fact that speaking frequency is unbalanced in the corpus, we use the inverse speaking frequency to release such imbalance [26]. Concretely, as for the utterance \( u_i \), the weight between \( u_i \) and its speaker \( p_i(u_i) \) is set to be \( 1/F \), where \( F \) represents the utterance number of the speaker \( p_i(u_i) \) in the whole corpus.

### 2.3.2. Graph Learning

To aggregate the local neighborhood information, we use the relation specific GNNs [27]. For a single-layer GNN, the new feature vector \( h_i^{(1)} \) is computed for the node \( v_i \in V \):

\[
h_i^{(1)} = ReLU \left( \sum_{r \in R} \sum_{j \in N_r^i} \frac{\alpha_{ij}}{|N_r^i|} W_r^{(1)} h_j^{(0)} \right)
\]

where \( \alpha_{ij} \) is the edge weight between node \( v_i \) and node \( v_j \). \( N_r^i \) represents the neighboring indexes of node \( v_i \) under relation \( r \in R \), and \( |N_r^i| \) is the number of \( N_r^i \). \( W_r^{(1)} \) is the trainable parameter for relation \( r \) and \( h_j^{(0)} \) is the original representation for node \( v_j \). As for a multi-layer GNN, the node features are updated by the following formula:

\[
h_i^{(l)} = ReLU \left( \sum_{r \in R} \sum_{j \in N_r^i} \frac{\alpha_{ij}}{|N_r^i|} W_r^{(l)} h_j^{(l-1)} \right)
\]

where \( l \) denotes the layer number. In our approach, we employ GNNs with \( L \) layers, where \( L \) is treated as a hyper-parameter.

After feature transformation by GNNs, we concatenate the final layer node embeddings \( h_i^{(L)} \in \mathbb{R}^d \) and original node embeddings \( h_i^{(0)} \in \mathbb{R}^d \) for the node \( v_i \). These embeddings are fed into a softmax classifier for emotion recognition.

\[
h_i = [h_i^{(0)}, h_i^{(L)}]
\]

\[
P_i = softmax(h_i W_l)
\]

where \( W_l \in \mathbb{R}^{2d \times c} \) is the trainable parameter. Here, \( h_i \in \mathbb{R}^{2d} \) and \( c \) is the number of emotion labels. \( P_i \in \mathbb{R}^c \) is the predicted label for the node \( v_i \). We choose the cross-entropy loss function during training:

\[
L = \frac{1}{K} \sum_{i=1}^{K} \sum_{j=1}^{L_s} Y_i^{(j)} \log P_i^{(j)}
\]

where \( K \) is the number of conversations and \( L_s \) is the number of utterances in the \( s \)-th conversation. \( P_i^{(j)} \in \mathbb{R}^c \) and \( Y_i^{(j)} \in \mathbb{R}^c \) are the emotion-class probabilities and one-hot vector ground truth for the \( j \)-th utterance in the \( i \)-th conversation, respectively.

### 3. Experiments and Discussion

#### 3.1. Corpus Description

We perform experiments on the popular benchmark dataset, the Multi-modal EmotionLines Dataset (MELD) [28]. MELD is a multi-party dataset where three or more speakers are involved in a conversation. All the conversations are split into small utterances, which are annotated using the following categories: anger, joy, sadness, neutral, disgust, fear and surprise. Totally, it contains 1433 conversations and 13708 utterances of various dialogue scenarios. To compare our method with state-of-the-art methods, we utilize the train/val/test splits in [22, 28]. The data distribution of the MELD dataset is listed in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#dialogues</th>
<th>#utterances</th>
</tr>
</thead>
<tbody>
<tr>
<td>MELD</td>
<td>1039</td>
<td>2160</td>
</tr>
</tbody>
</table>

#### 3.2. Data Representation

Frame-level acoustic features are extracted from raw waveforms using the openSMILE [29] speech toolkit with 25 ms frame window size and 10 ms frame intervals. Specifically, we use the extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) introduced by Eyben et al. [30]. Totally, 88-dimensional frame-level acoustic features are extracted; Word-level lexical features are extracted from the transcripts of spoken words. Specifically, we get 300-dimensional vector representation of words using the public available Word2Vec [31] model.

### 3.3. Experimental Setup

In the Utterance-level Feature Encoding process, Conv1D layers map acoustic and lexical features into the fixed dimension of size \( d = 30 \), followed with 5 multi-head attention blocks (with 30 dimensional states and 5 attention heads). To optimize the parameters, we use the Adam optimization, starting with an initial learning rate of 0.001. We train our model for 100 epochs with a batch size of 32. To alleviate over-fitting problems, we also use the dropout [32] with the rate 0.4. In our experiments, each configuration is tested 20 times with varied weight initializations. Experimental results are evaluated using the weighted average accuracy.

### 3.4. Impact of Multi-layer GNNs

To illustrate the impact of different numbers of GNN layers, we conduct experiments to compare the performance among unimodal and bimodal results. Experimental results are listed in Table 2. As for the textual modality, experimental results show that the performance of our proposed method first rises and then decreases, as the number of GNN layers increases. It shows that our method gains the best performance when using a two-layer GNN for the textual modality. Differently, as for the acoustic modality and multi-modality, we find that the performance decreases when the number of GNN layers increases. These results are the same with previous works [33, 34]. These works also show the limitations of stacking multiple GNN layers, which leads to highly complex back-propagation and the common vanishing gradient problem. Therefore, more than three layers of GNN seems not a good choice.
3.5. Importance of Individual Components

In this section, we evaluate the contribution of each component. Two comparison systems are implemented to compare with our proposed method. Table 3 provides the results on this analysis.

(1) System 1 (S1): It is our proposed method.
(2) System 2 (S2): It comes from (S1), but removing the utterance-level feature encoding process (in Figure 1). Specifically, to extract utterance-level features, we utilize mean values of frame-level (or word-level) features in the utterance.
(3) System 3 (S3): It comes from (S1), but removing the relation reduction process. Specifically, we use the graph structure in [22] with \(2M^2\) distinct relation types.

Firstly, to verify the effectiveness of utterance-level feature encoding (in Figure 1), we compare the performance of S1 and S2. Experimental results in Table 3 show that S1 is superior to S2 with a large margin. Compared with S2, our method learns long-term temporal dependence via self-attention mechanism. This structure is able to improve recognition performance.

Secondly, to verify the importance of relation reduction process, we compare the performance of S1 and S3. As shown in Table 3, we find our method is superior to S3 in all cases. The MELD dataset contains multi-party conversations and the average conversation length is 10 utterances. We find that many conversations have more than \(M = 5\) participants, which means that many speakers only utter a small number of utterances in a conversation. Without the relation reduction process, we need to model at least \(2M^2 = 50\) distinct relation types in a conversation, causing that each relation type cannot be fully learned [22]. Through our relation reduction process, we only need to model three relation types, which alleviates the challenges for speaker-sensitive modeling. Therefore, our relation reduction process improves the performance of emotion recognition.

3.6. Comparison to State-of-the-art Approaches

To verify the effectiveness of the proposed method, we further compare our method with other currently advanced approaches. Experimental results of different methods are listed in Table 4.

Compared with our proposed method, these approaches [5, 10, 22] also utilized acoustic and lexical features for conversational emotion recognition. Poria et al. [10] captured the context from surroundings via the bi-directional LSTM layer. However, this method suffered from incapability of capturing the speaker-sensitive dependence. To model this dependence, Majumder et al. [5] employed three GRUs to track individual speaker states, emotion states and global contexts during conversations. Ghosal et al. [22] modeled the context-sensitive and speaker-sensitive dependence via graph neural networks.

Experimental results in Table 4 demonstrate the effectiveness of our method. Compared with previous graph-based approaches [22], our graph-base method shows an absolute improvement of 0.5%, 2.4% and 2.2% for acoustic results, lexical results and bimodal results, respectively. These results verify the effectiveness of our relation reduction process. Meanwhile, our method shows an absolute improvement of 0.5%, 1.7% and 1.3% over state-of-the-art strategies for acoustic results, lexical results and bimodal results, respectively. These results serve as strong evidence that our proposed method can yield a promising performance for conversational emotion recognition.

4. Conclusions

In this paper, we propose a multimodal multi-party framework for conversational emotion recognition. Our method utilizes graph neural networks to model context-sensitive and speaker-sensitive dependencies in the conversation. Ablation studies verify the effectiveness of our proposed relation reduction process and utterance-level feature encoding process. Experimental results on the MELD dataset demonstrate the effectiveness of our proposed framework. As for lexical and bimodal results, our method shows absolute 1.3%~1.7% performance improvement over the state-of-the-art strategies.

5. Acknowledgements

This work is supported by the National Key Research & Development Plan of China (No.2017YFB1002804), the National Natural Science Foundation of China (NSFC) (No.61831022, No.61771472, No.61773379, No.61901473) and the Key Program of the Natural Science Foundation of Tianjin (Grant No. 18JCZDJC36300).
6. References


