Attentional Neural Network for Emotion Detection in Conversations with Speaker Influence Awareness

Jia Wei, Shi Feng, Daling Wang, Yifei Zhang and Xiangju Li

School of Computer Science and Engineering, Northeastern University, Shenyang, China
weijia_neu@163.com,
{fengshi,wangdaling,zhangyifei}@cse.neu.edu.cn,
lixiangju100@163.com

Abstract. Emotion detection in conversations has become a very important and challenging task. Most of previous studies do not distinguish different speakers in a dialogue and fail to characterize inter-speaker dependencies. In this paper, we propose **S**peaker **I**nfluence-aware **N**eural **N**etwork model (SINN) to predict the emotion of the last utterance in a conversation, which explicitly models the self and inter-speaker influences of historical utterances with GRUs and hierarchical attention matching network. Moreover, the empathy phenomenon is also considered by an emotion state tracking component in SINN. Finally, the target utterance representation is enhanced by speaker influence aware context modeling, where the attention mechanism is used to extract the most relevant features for emotion classification. Experiment results on DailyDialog dataset confirm that our model consistently outperforms the state-of-the-art methods.

Keywords: Emotion detection, Conversation, Speaker influence, Attention.

1. Introduction

Since the explosive growth of social media, massive conversations are produced through platforms (e.g., WeChat, Twitter and Weibo) on the Internet every day. Conversational emotion recognition plays a critical role in many applications such as cyber-crime investigation, human-robot interaction, customer service and so on. Thus, how to effectively detect emotions in conversations has attracted increasing attention from both academic and commercial communities.

A conversation consists of a sequence of utterances (2 at least) and each utterance is produced by a participant (the speaker). In this paper, we focus on the dyadic conversation between two speakers. It is generally known that the emotional dynamics in conversations are driven by two factors: self and inter-speaker emotional influence [1]. Self-influence reflects the speakers' own willingness to keep or change their emotions during dialogue. That means the emotion of the current utterance is closely related to the emotions of the speaker's past utterances. On the other hand, inter-speaker influence relates to emotional dynamics induced by the counterparts in the dialogue.

Despite the complex interactive emotional states of speakers in dialogue, most of the previous literature does not distinguish different speakers in a conversation and treat the context utterances only as a textual sequence. Recently, Hazarika et al. proposed CMN model to feed speakers' historical utterances into memory network [2], where each speaker is associated with a separate memory cell. Following this idea, Hazarika et al. further utilized GRU to model the influence between speakers [3]. Although these methods have achieved promising results, the inter-speaker influences are modeled by linear GRU utterance sequence or memory network, which could not fully capture the dependencies between the speakers during the dialogue.

To tackle these challenges, we propose a **S**peaker **I**nfluence-aware **N**eural **N**etwork model (dubbed as SINN) for emotion detection in conversations, which models the self and inter-speaker emotional influences explicitly and comprehensively. Specifically, SINN first adopts GRUs to deal with historical utterances of the target utterance based on each speaker. Furthermore, to incorporate inter-speaker influences, these histories are fed into two separate sections, which will extract speakers' interactive emotional features and track empathic states simultaneously. After that, the interactions between self as well as inter-speaker influence features with the target utterance are calculated by the attention mechanism to synthesize important contextual features. Eventually, the target utterance and the weighted contextual features are concatenated as a final representation which is used to predict the emotion category on the target.

To sum up, the main contributions of this paper are as follows:

- We propose a novel framework called **S**peaker **I**nfluence-aware **N**eural **N**etwork (SINN) to detect emotions in conversations. SINN leverages a hierarchical matching network to explicitly model self and inter-speaker influence and utilizes integrated components to comprehensively model the inter-speaker influence.
- We propose an attention mechanism to dynamically weight the speaker influence features, and learned an enhanced contextual representation.
- Extensive experimental results on benchmark dataset confirm that our SINN model outperforms state-of-the-art comparative methods for the emotion detection task.

2. Related Work

Most of the contextual sentiment analysis studies utilize some kinds of contextual information in the conversation. Huang et al. proposed a hierarchical LSTM model with two levels of LSTM networks to model the retweeting/replying process and capture the long-range dependencies between a tweet and its contextual tweets [4]. Ren et al. utilized two sub-modules to study features from conversation-based context, author-based context and topic-based context about a target tweet, respectively [5]. Andrea et al. employed a model named SVM^{hmm} using Markovian formulation of the SVM to predict the sentiment polarity of entire sequences of tweets [6].

A large section of researches tends to regard a tweet/microblog as a conversation with sequential characteristics. However, conversations in the real world contain quite different contextual information. Zhang et al. built a large-scale human-computer conversation data and adopted a single-level architecture by using Convolutional Neural Networks (CNNs) for sentiment classification [7]. Gupta et al. proposed a model consisting of two LSTM layers using two different word embedding matrices, Glove and SSWE, for detecting emotions in textual conversations [8]. Luo et al. proposed a self-attentive bidirectional long short-term memory network, which used self-attention to extract the dependence of all the utterances in the conversation [9].

However, the main shortage of these methods is that they do not treat the speakers in a conversation individually. Hazarika et al. utilized a Conversational Memory Network (CMN) to amend this shortcoming [2]. CMN considers utterance histories of each speaker to model emotional memories and uses memory network to capture inter-speaker dependencies. Then, Hazarika et al. proposed another improved model named as Interactive COnversational memory Network (ICON) [3]. Different from CMN, ICON adopts an interactive scheme that incorporates self and inter-speaker influences simultaneously and adopts a multiple hop scheme on them. Our model is inspired by ICON partially while quite different with ICON, where we adopt a more comprehensive approach to model the inter-speaker influences from two aspects, namely interactive dependency as well as empathy.

3. Proposed Model

Suppose there are n utterances in a dyadic two-person conversation, where the communication between two speakers P_A and P_B goes on alternately. Here, a conversation $\mathcal{C} = (u_A^1, u_B^2, u_A^3, u_B^4, \ldots, u_\lambda^n)$ is ordered temporally, where u_λ^n is the n^{th} utterance spoken by person P_λ, $\lambda \in \{A, B\}$. Our goal is to predict the emotion (Anger, Happiness, Sadness, Surprise and Neutral) of the last utterance in the conversation. The schematic overview of our proposed model SINN is shown in Figure 1.

As illustrated in Figure 1, our SINN network can be divided into three main parts: (1) self-influence modeling, (2) inter-speaker influence modeling, and (3) the interaction with the utterance to be predicted. The second part can be further broken down into two components: (a) interactive dependency matching and (b) empathy tracking.

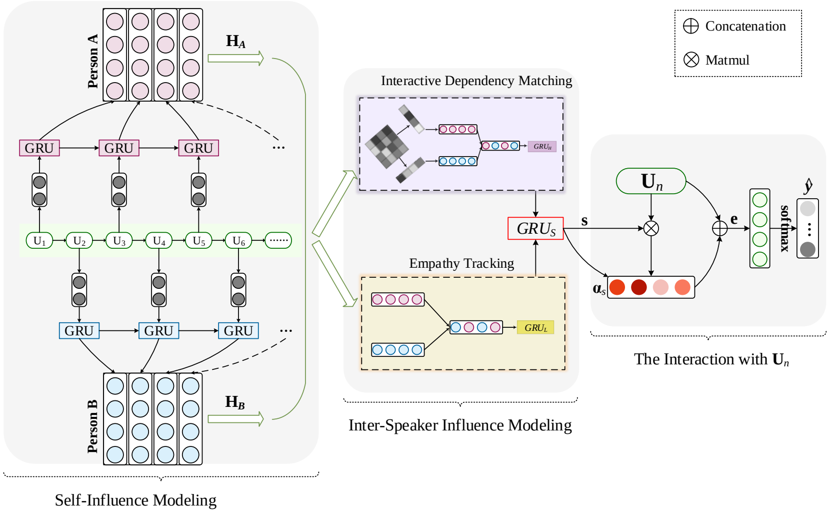

Fig. 1. The architecture of Speaker Influence-aware Neural Network (SINN)

3.1 Self-Influence Modeling

In this part, the first thing to do is to encode an utterance with distributed representation. For the n^{th} utterance in the conversation $\mathcal{C}$, pre-trained d-dimensional ELMo embeddings are adopted to represent each word of it. An utterance with m words is then represented as $\mathbf{U}_n = (\omega_1, \omega_2, \ldots, \omega_m)$, where ω_i is d-dimensional word embedding for the i^{th} word in the utterance, and we can get a $m \times d$ embedding matrix $\mathbf{W}$. Then we use CNNs and GRUs to extract features of matrix $\mathbf{W}$.

CNNs are effective in extracting representations of a sentence based on its constituting words. In this paper, we use a simple CNN with a single convolutional layer to deal with $\mathbf{W}$. The outputs are then fed into a max-pooling layer followed by a concatenation operation. In addition, we also employ GRU to extract sequential characteristics of an utterance. Each GRU cell computes a hidden state $h_t = GRU(h_{t\text{-}1}, x_t)$, where x_t is the current input and $h_{t\text{-}1}$ is the previous GRU state. We will explain the detail of GRU in the subsequent modules. The input of GRU here is individual words, and the hidden state of the last word is taken as the features of the entire utterance via GRU.

Eventually, the representation of an utterance $\mathbf{U}_n$ is a concatenation of the features from CNN and GRU, which enriches the representation of the utterance.

After the single utterance representation, we need to capture the self-influence on all historical utterances separately. The dialogue in $\mathcal{C}$ goes on alternately between two interlocutors. Here, for a $\mathcal{C} = (u_A^1, u_B^2, u_A^3, u_B^4, \ldots, u_\lambda^n)$, we split it into two series according to each speaker, getting $\mathcal{C}_A = (u_A^1, u_A^3, \ldots, u_A^i)$ and $\mathcal{C}_B = (u_B^2, u_B^4, \ldots, u_B^j)$ defined as new sequence $\mathcal{C}_\lambda = (u_{\lambda,1}, u_{\lambda,2}, \ldots, u_{\lambda,T})$, where $\lambda \in \{A, B\}$, $i < n$, $j < n$, $T \in \{i, j\}$. For each $\mathcal{C}_\lambda \in \{\mathcal{C}_A, \mathcal{C}_B\}$, we feed it into the GRU_λ to grasp the temporal history respectively. Specifically, at each timestep t, we get hidden state h_t as follows:

$$r_t = \text{sigmod}(\mathbf{W}^r h_{t\text{-}1} + \mathbf{V}^r x_t + \mathbf{b}^r) \tag{1}$$

$$z_t = \text{sigmod}(\mathbf{W}^z h_{t\text{-}1} + \mathbf{V}^z x_t + \mathbf{b}^z) \tag{2}$$

$$c_t = \tanh(\mathbf{W}^c (h_{t\text{-}1} \odot r_t) + \mathbf{V}^c x_t + \mathbf{b}^c) \tag{3}$$

$$h_t = z_t \odot h_{t\text{-}1} + (1 - z_t) \odot c_t \tag{4}$$

where $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{b}$ are parameter matrices and vector, and $\odot$ is dot product operation. x_t is the current input, which is the current utterance's representation $\mathbf{U}_t$ ($t \in [1, T]$) obtained from the approach mentioned above.

These hidden states of all timesteps can be concatenated together to form self-influence matrix $\mathbf{H}_\lambda = [h_{\lambda,1}, h_{\lambda,2}, \ldots, h_{\lambda,T}]$, $\mathbf{H}_\lambda \in \{\mathbf{H}_A, \mathbf{H}_B\}$. $\mathbf{H}_A$ or $\mathbf{H}_B$ represents the historical information of a speaker with his own previous utterances. After that, we encode two matrices $\mathbf{H}_A$ and $\mathbf{H}_B$ to further explore correlations between utterances.

3.2 Inter-Speaker Influence Modeling

It is a remarkable fact that each speaker or even an utterance in $\mathcal{C}$ will affect the progress of a conversation. In this part, we will introduce a novel approach to distill these influential factors through two components, interactive dependency matching component and empathy tracking component synchronously.

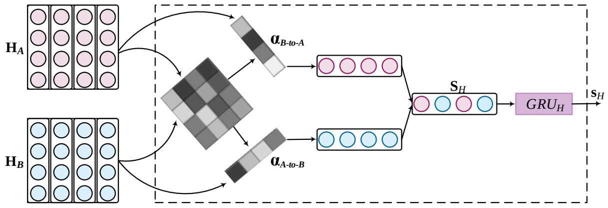
$\mathbf{H}_A$
$\boldsymbol{\alpha}_{B\text{-}to\text{-}A}$
$\mathbf{S}_H$
GRU_H
$\mathbf{s}_H$
$\mathbf{H}_B$
$\boldsymbol{\alpha}_{A\text{-}to\text{-}B}$

Fig. 2. Schematic overview of Interactive Dependency Matching

Interactive Dependency Matching: Since utterances constantly interfere with each other, we introduce an interactive mechanism to condense the hidden interplays between them. Figure 2 depicts the detail architecture of it. In order to compute features that are interdependent, we first calculate the confusion matrix $\mathbf{H} = \mathbf{H}_A \times \mathbf{H}_B^{\mathrm{T}}$. Given the confusion matrix $\mathbf{H}$, we apply it with attention mechanism [10] from two directions, which could be seen as a *B*-to-*A* attention and an *A*-to-*B* attention. Therefore, attention mechanism can help us to mine the significant interactive information between $\mathbf{H}_A$ and $\mathbf{H}_B$. Particularly, we need to calculate the attention scores of both sides involved, $\boldsymbol{\alpha}_{B\text{-to-}A}$ (the effect of person P_B on P_A) as well as $\boldsymbol{\alpha}_{A\text{-to-}B}$ (the effect of person P_A on P_B) which is inspired by [11]. Explicitly, the computations are as follows:

$$u_A = \tanh(\mathbf{W}_{w_1}\mathbf{H}^{\mathrm{T}} + \mathbf{b}_{w_1}) \tag{5}$$

$$\boldsymbol{\alpha}_{B\text{-}to\text{-}A} = \mathrm{softmax}(u_A^{\mathrm{T}}\mathbf{u}_{w_1}) \tag{6}$$

$$\mathbf{H}^{'}_A = \mathbf{H}_A\boldsymbol{\alpha}_{B\text{-}to\text{-}A} \tag{7}$$

where $\mathbf{W}_{w_1}, \mathbf{b}_{w_1}, \mathbf{u}_{w_1}$ are weight matrices and vector, and $\boldsymbol{\alpha}_{B\text{-to-}A} \in \mathbb{R}^{l_A}$ (l_A is the length of preceding utterances of P_A) is the attention weight vector implying the influence of person P_B's utterances on P_A. More precisely, each element in $\boldsymbol{\alpha}_{B\text{-to-}A}$ is the score that represents the importance of each utterance among P_A's previous utterances. More than that, due to the joining of $\mathbf{H}_B$, which represents the history of P_B, $\boldsymbol{\alpha}_{B\text{-to-}A}$ can also indicates the hidden trails of how P_B acts on P_A interactively. After this attention, we get a weighted matrix $\mathbf{H}^{'}_A$ of P_A's history based on the attention scores $\boldsymbol{\alpha}_{B\text{-to-}A}$.

We can get $\mathbf{H}^{'}_B$ by using the following formulas with different parameters:

$$u_B = \tanh(\mathbf{W}_{w_2}\mathbf{H} + \mathbf{b}_{w_2}) \tag{8}$$

$$\boldsymbol{\alpha}_{A\text{-}to\text{-}B} = \mathrm{softmax}(u_B^{\mathrm{T}}\mathbf{u}_{w_2}) \tag{9}$$

$$\mathbf{H}^{'}_B = \mathbf{H}_B\boldsymbol{\alpha}_{A\text{-}to\text{-}B} \tag{10}$$

Then, we use Eq.11 to integrate $\mathbf{H}^{'}_A$ and $\mathbf{H}^{'}_B$ into a complete interactive distribution of all previous utterances. Intuitively, we recover the original sequences of $\mathcal{C}$ ignoring speakers. $\mathbf{S}_H$ temporally denotes the interdependent abstract of each utterance and evaluates its importance at the same time. However, for extracting features more effectively, we adopt GRU_H to refine $\mathbf{S}_H$ and the output is viewed as a portion of our inter-speaker influence, which is expressed by $\mathbf{s}_H$.

$$\mathbf{S}_H = [\ \mathbf{H}'_{A,1}, \mathbf{H}'_{B,1}, \mathbf{H}'_{A,2}, \mathbf{H}'_{B,2}, \ldots\ldots, \mathbf{H}'_{\lambda,\text{n-1}}\ ] \tag{11}$$

Empathy Tracking: In this component, we model the emotional tracking of those historical utterances. The main purpose of this module is to ensure that we can maintain the empathic trend of $\mathcal{C}$, which will play a great role in inferring the final emotion. Since the emotion is extremely straightforward, we don't need to achieve it with such complicated process as component introduced above. For the sake of simplicity, $\mathbf{H}_A$ and $\mathbf{H}_B$ are first aggregated by Eq.12 along the temporal dimension, which incorporates with respective emotional labels at the same time.

$$\mathbf{S}_L = [\ \mathbf{H}_{A,1}L_{A,1}, \mathbf{H}_{B,1}L_{B,1}, \mathbf{H}_{A,2}L_{A,2}, \mathbf{H}_{B,2}L_{B,2}, \ldots\ldots, \mathbf{H}_{\lambda,\text{n-1}}L_{\lambda,\text{n-1}}\ ] \tag{12}$$

Similarly, we adopt another GRU_L to refine $\mathbf{S}_L$ to $\mathbf{s}_L$ denoting empathic features as another portion of our inter-speaker influence.

From the above two components, we can get a comprehensive historical features of $\mathbf{U}_n$. Eventually, we combine both the $\boldsymbol{s}_H$ and $\boldsymbol{s}_L$ through a GRU_S to merge them forming the inter-speaker influence features for further progress.

$$\mathbf{s} = GRU_S\ (\mathbf{s}_H \oplus \mathbf{s}_L) \tag{13}$$

3.3 The Interaction with $\boldsymbol{U_n}$

After accumulating the speaker influences of entire history, this step calculates the attentional weight of $\mathbf{s}$ with respect to target utterance $\mathbf{U}_n$. In Eq.5, 6, 8, 9, attention scores are got by considering the inner relevance of one input only. While quite different with aforementioned attentions, here we adopt an interactive mechanism to yield attention vector. In order to capture the attentive dependence of $\mathbf{s}$ relevant to $\mathbf{U}_n$, we perform a mutual calculation between them, which can be expressed as follows:

$$\boldsymbol{\alpha}_s = \text{softmax}\ (\mathbf{s}^{\mathrm{T}}\mathbf{U}_n) \tag{14}$$

$$\mathbf{e} = (\boldsymbol{\alpha}_s \odot \mathbf{s}) \oplus \mathbf{U}_n \tag{15}$$

From Eq.14, we get the attention scores $\boldsymbol{\alpha}_s$ based on the $\mathbf{U}_n$, which assigns higher attention to the information relevant to $\mathbf{U}_n$. We update the $\mathbf{s}$ according to $\boldsymbol{\alpha}_s$ and concatenate it with $\mathbf{U}_n$ to be our final emotional representation $\mathbf{e}$. The $\mathbf{e}$ contains the information about the $\mathbf{U}_n$ along with its context from entire previous utterances. To generate the final prediction of $\mathbf{U}_n$, $\mathbf{e}$ is fed into a fully-connected layer followed by a softmax layer to predict the target emotion.

The model is trained by minimizing the cross-entropy along with a L_2 regularization term. We also adopt dropout and early stopping to ease overfitting.

4. Experiments

4.1 Dataset

We conduct experiments on the DailyDialog dataset [12], which is a high-quality multi-turn dialog dataset reflecting our daily communication way. As far as we know, DailyDialog dataset is rarely used in the field of conversation sentiment analysis. On the original dataset, each utterance in a dialogue is annotated with one of seven emotion labels, which are Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral. Moreover, we find that Disgust and Fear emotions account for only a small proportion, with merely 353 (0.34%) and 174 (0.17%) utterances. In order to relieve the severe imbalance of data, we remove the dialogue that contains Disgust or Fear. Moreover, we split a dialogue with *n* utterances into *n*-1 sub dialogues that each sub dialogue includes at least two utterances, namely one historical utterance. After that, we get a modified dataset with 5 emotion labels, the distribution is shown in Table 1.

From Table 1 we can see that Neutral and Happiness appear more frequently, which is truly in accordance with our daily life. Other details can be counted that the speaker turns are roughly 8, and the average words per utterance is about 15.

Table 1. The statistics of the modified DailyDialog dataset

Emotion	Train	Dev	Test	Proportion
Neutral	61028	6140	5248	72416 (82.7%)
Anger	645	58	92	795 (0.9%)
Happiness	10113	642	914	11669 (13.3%)
Sadness	861	65	93	1019 (1.2%)
Surprise	1458	96	100	1654 (1.9%)
Total	74105	7001	6447	87553 (100%)

4.2 Experimental Setup

To initialize the word embedding matrix, we use the pre-trained 1024-dimension ELMo embedding of the output of second LSTM layer in ELMo model. All weight parameters are initialized using the default Tensorflow initializer and we use Adam optimization algorithm to train them with learning rate of 0.001. The number of convolutional filters is set 128 and the filter sizes are set as 2, 3 and 4. The number of GRU cells is 128 for all GRU modules except GRU_S, which contains 256 GRU cells. The weight of L_2 regularization term λ is set 0.001. Dropout rate of 0.5 is set to obtain better performance. Batch size is 128 finally.

We evaluate our experiments in terms of accuracy, and F1-score of the 5 emotion labels individually. Macro-averaged accuracy (Acc for short) and F1-score (F1 for short) are also reported on the whole data. Because the dataset has unbalanced classes as shown in Table 1, weighted averaged accuracy and F1-score are displayed for better contrast, as did in CMN [2] as well as ICON [3].

4.3 Baselines

In our experiments, we compare our proposed SINN network with the following baseline methods with the same word embeddings for fair comparison:

- **Hierarchical GRU-GRU (HGG** for short**):** This baseline contains two-level GRU networks. The first level is a word-level GRU, which can generate a representation of a single utterance. And the second level is an utterance-level GRU, which can model all the utterances in conversation temporally.
- **Hierarchical CNN-GRU (HCG** for short**):** Similar with HGG, HCG is also a two-level network, while we replace the first level GRU with CNN to model the word-level representation of an utterance.
- **CMN**[2]**:** This model uses GRUs to extract both speakers' utterances as historical memories. Then the current utterance is sent to two memory networks as a query with historical memories and employs attention mechanism on them. This step is performed *R* hops on these memories. In the original experiment, CMN gets its best performance when the number of hops is 3. Thus for a better comparison, we also set hops as 3 to apply CMN model to our dataset.
- **ICON**[3]**:** ICON is built based on CMN by the same authors. It also utilizes separate memory networks for both speakers' historical utterances. The difference with the CMN is that ICON incorporates self and inter-speaker influences in a dialogue with fewer trainable parameters. The hops are also set 3 on the memories.

4.4 Results and Discussion

The experimental results are shown in Table 2. As expected, our proposed model SINN, with novel approach to grasp speaker influence features, outperforms other baseline models obviously.

From Table 2, we can find that as a multi-level network, HGG performs relatively poorly compared with HCG. The reason may be due to the fact that CNNs is more efficient in extracting the features of a sentence than GRUs. That supports the way that we adopt CNN to extract the features of an utterance in our model. However, we still can not ignore the sequential characteristics of an utterance, so we use GRU to deal with it too. Both HGG and HCG perform worse than other baselines, the main reason can be that a simple two-layer architecture fails to excavate the deep dependencies between speakers which is extraordinary important in conversations.

ICON is the state-of-the-art model in [3], while on DailyDialog dataset CMN gets much advantage over ICON but is still not as good as our model. Both the ICON and CMN consider the interactions between speakers in conversation, and ICON incorporates self and inter-speaker influences in a conversation with fewer trainable parameters which may be the reason why ICON is inferior to CMN. That is to say, ICON is not guaranteed to work well in all situations.

Our final SINN model outperforms all the baseline models significantly by merging the self-influence with the inter-speak influence jointly to improve the representations of historical utterances and interacting with the target utterance by attention mechanism. We can see that the improvement is more than 20% on the macro averaged accuracy and F1-score, which confirms our initial assumption that utilizing the self and inter-speaker emotional influences is helpful for emotion prediction.

For each category in Table 2, we notice that SINN outperforms all the compared models except for Anger emotion on Acc and F1, and Sadness emotion on Acc. This situation may be caused by the fact that the number of training data of these two categories is not enough due to data imbalance, so that predicting emotion of Anger or Sadness is harder than the other emotions. However, in terms of weighted averaged accuracy and F1-score, our SINN acquires great improvement (more than 20%) compared with all other baselines, which can still support our view in the weighted condition. As CMN and ICON did in their experiments, we also use weighted averaged accuracy and F1-score to demonstrate the performance of our model.

Table 2. Comparison with the baseline models. Acc means accuracy, F1 means F1-score.

Model	Neutral		Anger		Happiness		Sadness		Surprise		Macro Avg		Weighted Avg	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
HGG	0.887	0.900	0.383	0.259	0.570	0.535	0.198	0.184	0.269	0.279	0.461	0.431	0.816	0.819
HCG	0.882	0.903	0.343	0.289	0.584	0.521	0.203	0.173	0.467	0.350	0.496	0.447	0.816	0.821
CMN	0.883	0.908	0.518	**0.392**	0.628	0.525	**0.349**	0.282	0.398	0.423	0.555	0.506	0.826	0.830
ICON	0.879	0.902	**0.533**	0.350	0.578	0.509	0.276	0.249	0.420	0.394	0.537	0.481	0.816	0.821
SINN-$_{\text{IDM}}$	0.882	0.913	**0.649**	**0.372**	0.662	0.540	**0.356**	0.313	**0.506**	**0.453**	**0.611**	0.518	0.834	0.836
SINN-$_{\text{ET}}$	0.899	0.915	0.295	0.315	**0.728**	0.562	0.289	0.327	0.481	0.425	0.536	0.509	0.842	0.840
SINN	**0.899**	**0.919**	0.490	0.350	**0.691**	**0.611**	0.327	**0.345**	**0.470**	**0.426**	**0.575**	**0.530**	**0.849**	**0.851**

4.5 Ablation Experiments

In this section, we implement several model variants for ablation experiments to verify how our model operates in various parts. The results are also shown in Table 2.

- **SINN-$_{\text{IDM}}$:** Due to the fact that baselines above don't consider the previous emotion labels of the target utterance, here we eliminate the empathy tracking component of our model for a better comparison.
- **SINN-$_{\text{ET}}$:** It is SINN without interactive dependency matching component.

As shown in Table 2, we can observe that both SINN-$_{\text{IDM}}$ and SINN-$_{\text{ET}}$ outperform baseline models on average, indicating that either SINN-$_{\text{IDM}}$ or SINN-$_{\text{ET}}$ can provide important inter-speaker clues to enhance the representations of historical utterances. And SINN-$_{\text{IDM}}$ outperforms SINN-$_{\text{ET}}$ on several categories, which are Anger, Sadness and Surprise with less samples, and some even better than final SINN. This situation is caused by the data imbalance since any negligible difference may arouse great margin on these categories. However, both SINN-$_{\text{IDM}}$ and SINN-$_{\text{ET}}$'s performance are still lower than SINN in terms of weighted averaged accuracy and F1-score, which means that the integrated entirety owns more ability than separate parts and each part plays an indispensable role on the whole SINN model.

5. Conclusion

In this paper, we propose a novel SINN modeling the self and inter-speaker influences to identify the emotions in the conversations. Our proposed SINN can extract the deep inter-speaker influences from two effective components and merge them with the target utterance in an intricate way. Moreover, we adopt multiple attention mechanism to help our model to pick up important information for predicting the final emotion. We demonstrated the effectiveness of our model on the high-quality conversational data DailyDialog and the results show that our model is superior to the state-of-the-art methods largely. This work can also be extended to multi-participant conversation which is left to our future work.

Acknowledgements. The work was supported by the National Key R&D Program of China under grant 2018YFB1004700, and National Natural Science Foundation of China (61872074, 61772122).

References

1. Morris, M., Keltner, D.: How emotions work: The social functions of emotional expression in negotiations. In: Research in organizational behavior 22, pp.1–50 (2000).
2. Hazarika, D., Poria, S., Zadeh, A., Cambria, E., Morency, L., Zimmermann, R.: Conversational Memory Network for Emotion Recognition in Dyadic Dialogue Videos. In: NAACL-HLT, pp. 2122-2132 (2018).
3. Hazarika, D., Poria, S., Mihalcea, R., et al..: ICON: Interactive Conversational Memory Network for Multimodal Emotion Detection. In: EMNLP, pp. 2594-2604 (2018).
4. Huang, M., Cao, Y., Dong, C.: Modeling Rich Contexts for Sentiment Classification with LSTM. arXiv preprint arXiv:1605.01478 (2016).
5. Ren, Y., Zhang, Y., Zhang, M., Ji, D.: Context-Sensitive Twitter Sentiment Classification Using Neural Network. In: AAAI, pp. 215-221 (2016).
6. Vanzo, A., Croce, D., Basili, R.: A context-based model for Sentiment Analysis in Twitter. In: COLING, pp.2345-2354 (2014).
7. Zhang, L., Chen, C.: Sentiment Classification with Convolutional Neural Networks: An Experimental Study on a Large-Scale Chinese Conversation Corpus. In: CIS, pp. 165-169 (2016).
8. Gupta, U., Chatterjee, A., et al.: A Sentiment-and-Semantics-Based Approach for Emotion Detection in Textual Conversations. arXiv preprint arXiv:1707.06996 (2017).
9. Luo, L., Yang, H., Chin, Y. L. F.: EmotionX-DLC: Self-Attentive BiLSTM for Detecting Sequential Emotions in Dialogues. In: SocialNLP@ACL, pp. 32-36 (2018).
10. Yang, Z., Yang, D., Dyer, C., et al.: Hierarchical Attention Networks for Document Classification. In: HLT-NAACL, pp. 1480-1489 (2016).
11. Shen, C., Sun, C., Wang, J., et al.: Sentiment Classification towards Question-Answering with Hierarchical Matching Network. In: EMNLP, pp. 3654-3663 (2018).
12. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. arXiv preprint arXiv: 1710.03957 (2017).