Improved DialogueRNN: Dealing with Emotional Shift

Jinseo Lee
School of Computing, KAIST

Darae Lee
School of Computing, KAIST

Jonghee Jeon
School of Computing, KAIST

Joohee Kim
School of Computing, KAIST

Abstract

Emotion detection in conversations is gaining attentions because of its diverse applications such as artificial emotional intelligence, automated customer satisfaction center and so forth. In this regard, one innovative method had introduced from this field, which was tracking individual party's states as well as the context of a dialogue. However, it was not able to successfully detect the emotions changing rapidly. In this paper, we define those sudden changes of emotions as emotional shifts and introduce a new way to detect and handle it appropriately. In consequence, we can outperform the state of the art baseline model, DialogueRNN.

1 Introduction

Emotion detection in dialogues is gaining attentions owing to various applications including artificial emotional intelligence, smart customer satisfaction center, auto industry and so on. In this paper, we propose a new model improving existing methods based on recurrent neural networks(RNN) which can predict emotions from conversation datasets more accurately.

Among recent approaches, one model called *'DialogueRNN'* considering participants of conversations and keeping track of their states shows superior performance in this area. (Majumder et al., 2018) However, it still has limitations that it fails to predict successfully in some conditions, such as the circumstances in which emotions switch quickly. We define this phenomenon as an *"emotional shift"*, and investigate a few variables whether they have relationships to emotional shift to get insights before building our model. Therefore, we believe this approach can improve the performance of predicting since it can appropriately respond to the emotional shift.

The remaining parts of this paper consist of five sections: related work; methodology; experimental settings; results and discussion; conclusion.

2 Related Work

There are many different approaches to accurately detect emotions of man, especially in conversations. Alm, Roth and Sproat (2005) pioneered emotion recognition in dialogues, and this has been improved in various ways. Among lost of methods, we target DialogueRNN as a baseline model to enhance introduced by Majumder et al. (2018) adapting RNN and paying special attention to speakers' states as well as contexts.

To define the emotional shift and look for correct variable which can catch this well, the profound understand about conversations is required. In this context, See et al. (2019) found four attributes of conversations. Thus we decide to use these variables along with the concept of entropy. (1996)

3 Methodology

3.1 Problem Definition

The task aims to predict emotions of utterances in a conversation having M speakers.

Emotion in our dataset is represented in predefined labels such as (*happy, sad, neutral, angry, excited, frustrated* or continuous real number). These are transformed into the arousal-valence space by estimating each emotion's approximate placement in a circumplex model of Russell. (Russell, 1980) Here, the ranges for both dimensions are restricted to [-5,5].

3.2 Feature Extraction and DialogueRNN

DialogueRNN (Majumder et al., 2018) is our baseline model. Thus we adopt its feature extraction model and GRU-cell approach.

Feature Extraction More specifically, convolutional neural networks(CNN) (Kim, 2014) are used for that of text data. N-gram features are extracted first by three convolution filters having 50 feature-maps of size 3, 4, 5 then max-pooling and rectified linear unit (ReLU) activation are applied. And the results are concatenated, passed on to a 100-dimensional dense layer and now ready for training and testing. Meanwhile, about audio and video data, openSMILE (Eyben et al., 2010) and 3D-CNN (Hazarika et al., 2018) is used correspondingly though the text data is the main target of our model. Anyway, the utterance can be represented as u_t and $u_t \in \mathbb{R}^{D_M}$ where $D_M = 100$.

GRU cell DialogueRNN uses 3 GRU cells (Chung et al., 2014): *Party GRU, Global GRU and Emotion GRU*. These are used to apply previous utterances and emotions to influence the current decision of choosing an emotion. Each cell contains a hidden state, which can be thought of as a storage of previous information, and updates the value according to a new incoming input. The 3 GRU cells each memorizes a different aspect of the conversation, and the three are as the following.

Global GRU Minutely, Global GRU catches the context of an utterance through the speaker state and utterances along with its preceding global states. That is, with the size of a global state vector D_g and the concatenation $\oplus$,

$$g_t = GRU_g(g_{t-1}, (u_t \oplus q_{s(u_t),t-1})), \quad (1)$$

where D_p is the size of a party state vector and the trainable parameters are $W^{\{r,z,c\}}_{*,\{h,x\}}$ and $b^{\{r,z,c\}}_{*}$, $W^{\{r,z,c\}}_{g,h} \in \mathbb{R}^{D_g \times D_g}$, $W^{\{r,z,c\}}_{g,x} \in \mathbb{R}^{D_g \times (D_m + D_p)}$, $b^{\{r,z,c\}}_{g} \in \mathbb{R}^{D_g}$, $q_{s(u_t),t-1} \in \mathbb{R}^{D_p}$ and $g_t, g_{t-1} \in \mathbb{R}^{D_g}$.

Party GRU On the other hand, Party GRU records the states of each participant during the conversation. Each state is updated according to its role: speaker or listener. If the participant is the speaker, we update Party GRU but otherwise just maintain the value. About updating procedure, the context c_t should be extracted from the utterance u_t first, as follows:

$$\alpha = softmax(u_t^T W_\alpha [g_1, g_2, ..., g_{t-1}]), \quad (2)$$

$$softmax(x) = [e^{x_1} / \sum_i e^{x_i}, e^{x_2} / \sum_i e^{x_i}, ...], \quad (3)$$

$$c_t = \alpha [g_1, g_2, ..., g_{t-1}]^T, \quad (4)$$

where $W_\alpha \in \mathbb{R}^{D_m \times D_g}$, $\alpha^T \in \mathbb{R}^{(t-1)}$ and $c_t \in \mathbb{R}^{D_g}$. Then updating state through Party GRU GRU_p,

$$q_{s(u_t),t} = GRU_p(q_{s(u_t),t-1}, (u_t \oplus c_t)), \quad (5)$$

where $W^{\{r,z,c\}}_{p,h} \in \mathbb{R}^{D_p \times D_p}$, $W^{\{r,z,c\}}_{p,x} \in \mathbb{R}^{D_p \times (D_m + D_g)}$, $b^{\{r,z,c\}}_{p} \in \mathbb{R}^{D_p}$ and $q_{s(u_t),t}, q_{s(u_t),t-1} \in \mathbb{R}^{D_p}$.

Emotion GRU Finally, the Emotion GRU keeps on track of what emotions it went through. And its specific calculation, with the size of an emotion representation vector D_e is as follows:

$$e_t = GRU_e(e_{t-1}, q_{s(u_t),t}), \quad (6)$$

where $e_{\{t,t-1\}} \in \mathbb{R}^{D_e}$, $W^{\{r,z,c\}}_{e,h} \in \mathbb{R}^{D_e \times D_e}$, $W^{\{r,z,c\}}_{e,x} \in \mathbb{R}^{D_e \times D_p}$ and $b^{\{r,z,c\}}_{e} \in \mathbb{R}^{D_e}$.

Emotion Label-based Classification Based on the calculated emotion representations, DialogueRNN predicts the emotion label $\hat{y}_t$ (6-classes in IEMOCAP, so $c = 6$) of the utterance u_t is calculated by using ReLU as its activation function, and adding an additional softmax layer. The detailed functions are as the following.

$$l_t = ReLU(W_l e_t + b_l), \quad (7)$$

$$\rho_t = softmax(W_{smax} l_t + b_{smax}), \quad (8)$$

$$\hat{y}_t = \underset{i}{argmax}(\rho_t[i]), \quad (9)$$

where $W_l \in \mathbb{R}^{D_l \times D_e}$, $b_l \in \mathbb{R}^{D_l}$, $W_{smax} \in \mathbb{R}^{c \times D_l}$, $b_{smax} \in \mathbb{R}^c$ and $\rho_t \in \mathbb{R}^c$.

Training For training, categorical cross-entropy and L2 regularization are used for measuring loss L,

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

where N is the size of dataset, $c(i)$ is the size of the i-th dialogue (i.e. the number of utterances in the

i-th dialogue), $\rho_{i,j}$ is the probability distribution of labels for j-th utterance in i-th dialogue, $y_{i,j}$ is prediction for j-th utterance in i-th dialogue, λ is the weight for L2 regularization and θ is the parameters, which can be expressed as

$$\theta = \{W_\alpha, W_{\rho,\{h,x\}}^{\{r,z,c\}}, b_\rho^{\{r,z,c\}}, W_{g,\{h,x\}}^{\{r,z,c\}}, b_g^{\{r,z,c\}}, W_{e,\{h,x\}}^{\{r,z,c\}}, b_e^{\{r,z,c\}}, W_l, b_l, W_{smax}, b_{smax}\}.$$

In addition, first-order gradient-based algorithm Adam (Kingma and Ba, 2014) is used as the optimizer. (It was known to better than stochastic gradient descent (SGD)).

3.3 iDialogueRNN(Our Model)

Our model, *iDialogueRNN* or improved DialogueRNN consists of two main parts: the emotional shift detector and the prediction enhancer module (PEM). If the detector catches the occurrence of an emotional shift, PEM keeps it in mind and predicts an emotion accordingly. This can be summarized by figure 1.

Emotional Shift Before examining our model, we have to define what emotional shift exactly is in advance. The definition itself is relatively simple: *Emotional shift is a rapid change of emotion which happens in a dialogue.* However, more important task is investigating which variable can actually detect this sudden change. From what see et al. found, (See et al., 2019) and from our preliminary research, we select four variables as candidates: specificity, similarity, question-asking and entropy.

Specificity Specificity of utterance u of dialogue d is the variable measuring how rare u is among d, based on word rareness Normalized IDF. (See et al., 2019) For a word w:

$$NIDF(w) = \frac{IDF(w) - min\_idf}{max\_idf - min\_idf}$$

$$Specificity(u) = \overline{NIDF(w)},$$

where $IDF(w) = \log(R/c_w)$, $\overline{NIDF(w)}$ is the average NIDF of all words constitute u and min_idf, max_idf are the minimum and maximum over all vocabulary in d, respectively.

Similarity Similarity between two utterances are calculated by a cosine similarity as follows:

$$Similarity(A, B) = cos\_sim(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{11}$$

where u and u' are utterances that seeks to find similarity.

Question-asking Question-asking between two utterances is simply the difference of the curiosity each implies in it. The curiosity of a utterance u is defined by the number of pre-selected words contained in u, where the pre-selected words are 'how', 'what', 'when', 'where', 'which', 'who', 'whom', 'whose' and 'why'.

Entropy Entropy is a measure of the amount of information contained in one utterance. This is defined as follows, when w_i mean words of an utterance u and $P(w_i)$ is the probability that w_i will appear in u:

$$Entropy(u) = \sum_i P(w_i) \log_2 P(w_i).$$

Emotional Shift Detector To rationally design the appropriate detection algorithm, we experiment given four variables and compare their capabilities for catching emotional shifts in actual datasets. We provide details of this experiment in Section 4.3. In short, we conclude the similarity is the best option. And thus we devise an emotional shift detection algorithm using this measure.

Unlike the original model, our new model-iDialogueRNN-calculates similarity at the very beginning of each training. (i.e. for each utterance) If the calculated value is smaller than predetermined threshold value, it judges that an emotional shift occurs so set $Detection_t$ as 1, which is the variable for binary classification for occurrence of the emotional shift.

$$Detection_t = \left\{ \begin{matrix} 1 \; if \; Similarity(u_{t-1}, u_t) < \tau_r \\ 0 \; otherwise \end{matrix} \right\}, \tag{12}$$

where a speaker is changed at time t and τ_r is a threshold value in response-centric, or

$$Detection_t = \left\{ \begin{matrix} 1 \; if \; Similarity(u_{t-1}, u_t) < \tau_S \\ 0 \; otherwise \end{matrix} \right\}, \tag{13}$$

where a speaker is same during $t-1$ and t, τ_S is a threshold value in speaker-centric value.

Prediction Enhancer Module (PEM) After an emotional shift is detected, we try to modify DialogueRNN to predict responding to it. And to do this, we introduce 'weighted global state vectors'. This approach can be briefly explained by reducing the implications of previous global state

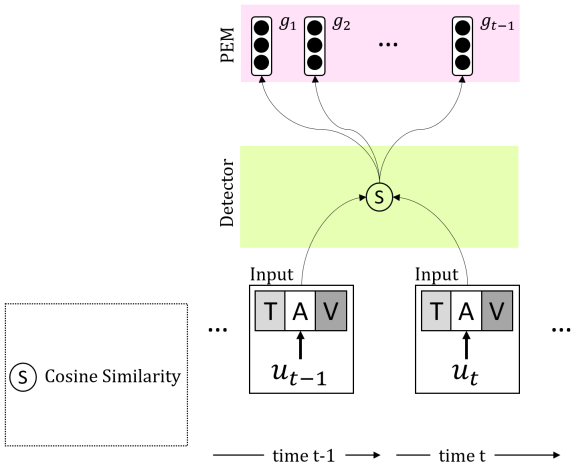


Figure 1: iDialogueRNN architecture.

vectors for predicting an emotion. In this sense, our model should update the global state vectors right after the detection, resulting in the weighted global state vectors. Updating procedure is defined as the scalar product of the global state vectors before the current global state vector g_t by the predetermined weight.

$$g'_1, ..., g'_{t-1} = g_1 \times \omega, ..., g_{t_1} \times \omega, \qquad (14)$$

where ω is the predetermined weight and g'_k is the updated value of g_k. ω can be ω_r and ω_s which are response-centric weight and speaker-centric weight, respectively.

4 Experimental Setting

4.1 Datasets

Following the baseline paper, (Majumder et al., 2018) we use two datasets for training: IEMOCAP (Busso et al., 2008); AVEC (Schuller et al., 2012). And we divide these datasets into two sub-sets for training (plus validating) and testing by the ratio of four to one while maintaining the state that no any speaker exists in both sub-sets. In addition, we analyze two more datasets for hypothesis testing: MELD (Poria et al., 2018) and DailyDialog (Li et al., 2017).

IEMOCAP IEMOCAP is a multimodal and multi-speaker video dataset. But there are only two speakers in one dialogue. It has total of 7,433 sentences and each utterance is labeled with following 6 labels; happy, sad, neutral, angry, excited, and frustrated.

AVEC AVEC is an audiovisual dataset containing videos which participants speak to one of four artificial agents. It has total of 5,816 sentences and each videos is labeled continuously in terms of time and dimensions originally: continuous real number of arousal, expectation, power and valence in different ranges, for every 0.2 seconds. However, to use the AVEC dataset for DialogueRNN and iDialogueRNN, we averaged out them over the unit of an utterance. unit.

MELD Multimodal Emotion Lines Dataset (MELD) is a dataset extended from Emotion Lines dataset to audio and visual modality as well as text. There are nine speakers, and there are several speakers in one dialogue. It has total of 13,708 sentences and each utterance is labeled with following 7 labels; anger, disgust, sadness, joy, neutral, surprise, and fear.

DailyDialog DailyDialog is a high-quality multi-turn dialog dataset, and there are only two speakers in one dialogue. It has total of 8,206 sentences, and each utterance is labeled with following 7 labels; anger, disgust, fear, joy, sadness, and surprise.

4.2 Dataset Analysis

In order to understand the sequence of emotion changes and emotion detection accuracy, we analyzed emotion labels for each dataset.

In all dataset, we chose two flows: speaker-centric and response-centric. In the speak-centric approach, only the utterances of the same speaker are listed in chronological order in the conversation. In the response-centric approach, speaker-changing utterances are used. The response-centric approach

can measure the degree of change according to the response between speakers.

Emotional distribution To understand the distribution of emotion labels, we counted the corresponding number of utterances for each label.

Emotional Change To understand the flow of emotions in the utterances in the dialogue, we created a transition matrix based on the previous and current values of the emotion label. We also measured the rate at which emotion changes occur in the overall utterances.

DialogueRNN performance check To find out which emotion shift detection DialogueRNN is vulnerable to, we measured the emotion label where false prediction occurs. We've ran the model and splitted each dataset to the correctly classified ones and the ones that weren't. And used it to check the precision and recall of every emotional shift prediction.

4.3 Hypothesis Testing

To investigate which variable is the most similar to the emotional switching, we conduct the experiment. In hypothesis testing, the same two flows as data analysis were used: speaker-centric and response-centric. In the speak-centric approach, we measured the change in the utterance and the corresponding change in emotion label. In the response-centric approach, we measured the change in utterance progressed by several speakers within the conversation. The response-centric approach can measures the degree of change according to the response between speakers.

1. First, extracting sentences, labels and speakers from the dataset.

2. Then creating two modified datasets using extracted information: one in the speaker-centric order; the other one in the response-centric order. Imagine a situation with two participants F, M and a dialogue δ consisting of utterances u_F, u_M spoken by F and M respectively as follows: $(u_F, u_M, u_M, u_F, u_M)$. In this circumstance, the former should be $(u_F \rightarrow u_M, u_M \rightarrow u_F, u_F \rightarrow u_M)$ whereas the latter should be $u_M \rightarrow u_M$.

3. At the same time, vectorizing emotion labels according to a circumplex model of Russell. (Russell, 1980) Specifically, in IEMOCAP dataset, "happy" is converted to $[2, -2]$ for instance.

4. Then calculating distances between two mutually-paired emotion label vectors.

5. Based on the above specifications of each variable, calculating the value differences in two consecutive utterances. To give an example of similarity for dialogue δ, we should calculate the difference of two sequential u_Ms.

6. Finally, obtaining the cosine similarity between emotional vectors' differences(distances calculated from 4.) and variables' differences.

Vectorizing labels can be skipped for AVEC dataset, because its labels are presented as vectors

4.4 Baseline Result

For the fair comparison, we test both models with the same environments five times and use the average as the final result. We provide relevant data in Table 3's DialogueRNN row.

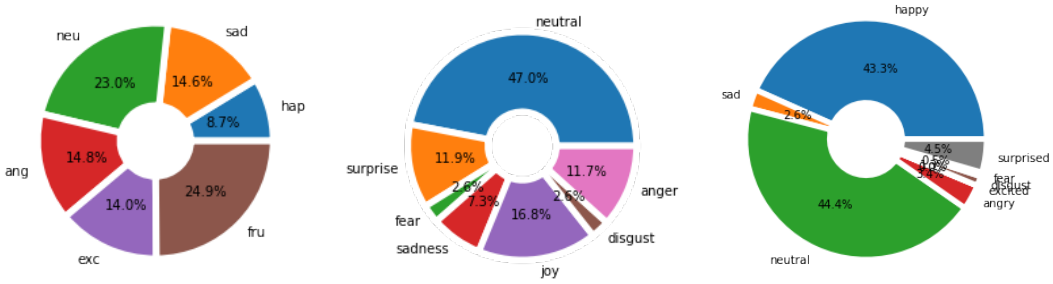
5 Results and Discussion

5.1 Dataset Analysis

Analysis is done in IEMOCAP, MELD, and DailyDialog. We excluded AVEC dataset because of the continuity in emotion labeling.

Emotional distribution Fig. 2(a), Fig. 2(b), Fig.2(c) shows the distribution of emotion label in each dataset. In IEMOCAP, neutral label takes 23%, positive emotion labels (i.e. happy, excited) take 22.7%, and negative emotion labels (i.e. sad, angry, frustrated) take 54.3% in total. We can observe that negative emotion labels account for a larger percentage of the total. In MELD, it is obvious that neutral emotion label takes about half of the total, and positive emotion labels (i.e. joy, surprised) take 28.7%, and negative emotion labels (i.e. anger, disgust, sadness, fear) take 24.1% in total. In DailyDialog, 'happy' and 'neutral' takes 43.3% and 41.4% in total, which is over 80% of the total data. In conclusion, we can observe that emotion labels are imbalanced in each dataset.

Emotional change To analyze how the emotion label changes, we extract pair of emotion labels of the previous utterance and the current utterance. Then we calculate the frequency of pairs



(a) (b) (c)

Figure 2: Emotional distribution of (a) IEMOCAP (b) MELD (c) DailyDialog

and show it as the transition matrix of emotion label change in speaker-centric flow and response-centric flow. The dark color in diagonals in the transition matrix indicates no emotional shift. In IEMOCAP (Fig3(a),Fig3(b)) speaker-centric flow shows higher frequency in the diagonal relative to response-centric flow, which means there are not much emotional shift. In MELD(Fig4(a),Fig4(b)) and DailyDialog (Fig5(a),Fig5(b)), the imbalance in data label shows imbalance in transition matrix. The emotional shift ratio is calculated in the Table1. IEMOCAP shows significant difference between speaker-centric flow and response-centric flow, about 28% while other dataset shows difference below 5%. This indicates that in IEMOCAP, the emotion flow of the speaker is more active and this is influenced by other speakers.

	speaker-centric flow	response-centric flow
IEMOCAP	0.28	0.56
MELD	0.55	0.60
DailyDialog	0.37	0.33

Table 1: Emotional shift ratio

DialogueRNN performance check The result is shown at Fig. 5. We found that the models usually performed well when emotional shift didn't happen. Thus, our goal of handling cases with emotional shift could be approved. Additionally, we found that there were some shifts such as 'angry' to 'happy' such that never existed in the datasets.

5.2 Hypothesis Testing

We provide results in Table 2. We can check that similarity is the best for following the emotional shift in both approaches for every dataset. Based on this result, we decide to use similarity in both speaker-centric and response-centric as the detector.

5.3 Prediction Performance

We provide performances of our model and the baseline model in terms of F1, mean absolute error and Pearson correlation coefficient in Table 3. Our model outperforms dialogueRNN in IEMOCAP although the performance decreases happen in happy, sad and angry labels. However it failed to show significant betterment in AVEC and this is due to poor predetermined parameters caused by compared to IEMOCAP.

From this, we can infer that PEM in our model actually helps to increase accuracy, but also it affects negatively in some cases through the unnecessary diminution in contexts' effect.

5.4 Ablation Study

To deeply investigate the impact of detection by both similarities-one in response-centric and the other in speaker-centric-and effectiveness, we conduct an ablation study with IEMOCAP dataset. We provide the result in Table 4.

We can find that using response-centric similarity only results in better performance than combining two approaches. This is related to drawbacks of our model: *cutting global state vectors' influence has side effects that disturbing prediction using context information.*

5.5 Future Research

We analyze our model and identify the crucial limitation of current model. That is, our model sometimes affect performance badly due to insufficient context referencing caused by unnecessary weighing. To overcome this problem, we suggest two

		Specificity	Similarity	Question-asking	Entropy
Speaker-centric	IEMOCAP	0.7733	**0.8421**	0.4980	0.7808
	AVEC	0.5488	**0.5844**	0	0.5064
	MELD	**0.5237**	0.4889	0.4242	0.4996
	DailyDialog	0.5552	**0.9371**	0.5282	0.9249
Response-centric	IEMOCAP	0.7707	**0.8234**	0.5822	0.7181
	AVEC	0.5086	**0.5744**	0	0.4421
	MELD	0.5363	**0.5625**	0.4965	0.5376
	DailyDialog	0.5633	**0.9290**	0.6545	0.8955

Table 2: Hypothesis testing result; bold font denotes the best result.

Methods	IEMOCAP							AVEC			
	Happy	Sad	Neutral	Angry	Excited	Frustrated	**Average(w)**	Valence	Arousal	Expectancy	Power
DialogueRNN	36.94	77.72	54.93	63.88	64.90	58.91	60.48	0.1781/0.3183	0.1846/0.3491	0.1816/0.3023	25.73/-0.0384
iDialogueRNN	36.08	75.57	55.6	62.59	68.73	59.04	60.84	0.1798/0.1575	0.1849/0.3814	0.1821/0.3003	20.47/-0.038
Difference	-0.86	-2.15	+0.67	-1.29	+3.83	+0.13	+0.36	+0.17/-0.1608	+0.0003/0.035	+0.0005/-0.002	-5.26/+0.0004

Table 3: Performances of models; all numbers below the IEMOCAP row are the average F1 over five times or difference; all numbers below AVEC row are the average Mean Absolute Error and Pearson correlation coefficient over five times in sequence or differences; Average(w) is the weighted average of F1 for each trial; each difference is calculated by $iDialogueRNN'svalue - DialogueRNN'svalue$.

response-centric	speaker-centric	F1 score
+	-	61.43
-	+	60.74
+	+	60.84

Table 4: Ablated iDialogueRNN's performance in IEMOCAP; F1 is the weighted average score.

further modifications for future research: detection with multiple variables; PEM with a CRF layer.

Multi-variable Detector The first way to correct this is improving accuracy of detection, resulting in minimized side effects. In our current model, we use the similarity only to detect the emotional shift. Although it is true that similarity is the best variable following emotional shifts, still there is room for an enhancement: combining it with other variables.

It is possible that combining a number of variables, to increase the accuracy of detection but we should find moderate weights between them. For that, we suggest linear regression. To give an example, imagine we want to combine the entropy with the similarity and the specificity. Then detecting variable is defined as $d = w_0 Entropy + w_1 Similarity + w_2 Specificity$, and find the best sets of weights through experiments. (i.e., going through the linear regression process)

PEM with a CRF layer The second way is to make a prediction enhanced module of extra conditional random field(CRF) layer above the emotion representation cell (GRU_e) in DialogueRNN model. This module will be used additionally when the emotional shift detector detects the shift. CRF layer is a probabilistic graphical model that captures non-independent features of data. The model calculates the conditional probability of the input label sequence and outputs the label which maximizes its probability. By adding this additional layer, the model can take the sequence dependency of emotions into account when predicting emotion layer.

6 Conclusion

We have analyzed a dialogueRNN's limitation-an emotional shift-and designed a new model which can correct it. This new model is called iDialogueRNN and consists of two sub-modules, detector and prediction enhancer. The detector detects occurrences of emotional shifts then enhancer reflects it through weighted global state vectors. Our model surpasses the baseline model in one dataset but not in the other. Our model can be improved by multi-variable detector and PEM with CRF layer, which is our future research plan.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *Proceed-*

ings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pages 579–586, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Adam Berger, Stephen A Della Pietra, and Vincent J Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. 2008. Iemocap: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. opensmile – the munich versatile and fast open-source audio feature extractor. pages 1459–1462.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2122–2132, New Orleans, Louisiana. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2018. Dialoguernn: An attentive rnn for emotion detection in conversations.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations.

James Russell. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178.

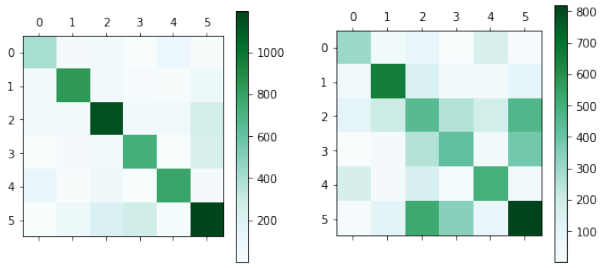
Björn Schuller, Michel Valstar, Roddy Cowie, and Maja Pantic. 2012. Avec 2012: The continuous audio/visual emotion challenge - an introduction. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, page 361–362, New York, NY, USA. Association for Computing Machinery.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments.

A Appendix

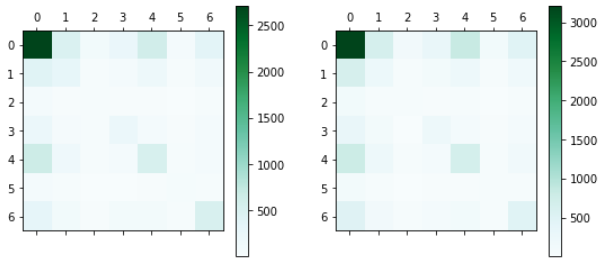
From	To	Precision	Recall	From	To	Precision	Recall
Happy	Happy	0.973	0.856	Angry	Happy	0.0	0.0
	Sad	0.5	0.8		Sad	0.2	1.0
	Neutral	0.05	0.125		Neutral	0.471	0.444
	Angry	None	None		Angry	0.943	0.209
	Excited	0.273	0.545		Excited	0.4	1.0
	Frustrated	None	0.0		Frustrated	0.373	0.704
Sad	Happy	0.02	0.75	Excited	Happy	0.012	0.13
	Sad	0.938	0.478		Sad	0.0	None
	Neutral	0.222	0.308		Neutral	0.286	0.25
	Angry	0.0	0.0		Angry	None	0.0
	Excited	0.143	1.0		Excited	0.851	0.406
	Frustrated	0.4	0.333		Frustrated	0.25	0.667
Neutral	Happy	0.021	0.571	Frustrated	Happy	0.005	0.5
	Sad	0.5	0.8		Sad	0.2	0.9
	Neutral	0.851	0.397		Neutral	0.261	0.545
	Angry	0.556	0.526		Angry	0.29	0.667
	Excited	0.559	0.613		Excited	0.429	0.75
	Frustrated	0.202	0.419		Frustrated	0.892	0.368

Table 5: DialogueRNN Performance Check



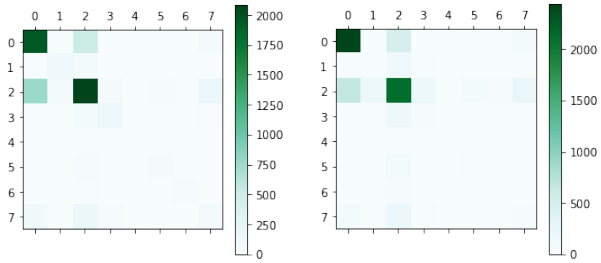
(a) (b)

Figure 3: Emotional transition matrix of IEMOCAP
(a) in speaker-centric flow (b) in response-centric flow
{ 0:happy, 1:sad, 2:neutral, 3:angry, 4:excited, 5:frustrated }



(a) (b)

Figure 4: Emotional transition matrix of MELD
(a) in speaker-centric flow (b) in response-centric flow
{ 0:neutral, 1:surprise, 2:fear, 3:sadness, 4:joy, 5:disgust, 6:anger }



(a) (b)

Figure 5: Emotional transition matrix of DailyDialog
(a) in speaker-centric flow (b) in response-centric flow
{ 0:happy, 1:sad, 2:neutral, 3:angry, 4:excited, 5:disgust, 6:fear, 7:surprise }