Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory

Hao Zhou[†], **Minlie Huang**[†], **Tianyang Zhang**[†], **Xiaoyan Zhu**[†], **Bing Liu**[‡]
[†]State Key Laboratory of Intelligent Technology and Systems,
National Laboratory for Information Science and Technology,
Dept. of Computer Science and Technology, Tsinghua University, Beijing, PR China
[‡]Dept. of Computer Science, University of Illinois at Chicago, Chicago, Illinois, USA
tuxchow@gmail.com, aihuang@tsinghua.edu.cn,
keavilzhangzty@gmail.com, zxy-dcs@tsinghua.edu.cn, liub@cs.uic.edu

arXiv:1704.01074v1 [cs.CL] 4 Apr 2017

Abstract

Emotional intelligence is one of the key factors to the success of dialogue systems or conversational agents. In this paper, we propose Emotional Chatting Machine (ECM) which generates responses that are appropriate not only at the content level (relevant and grammatical) but also at the emotion level (consistent emotional expression). To the best of our knowledge, this is the first work that addresses the emotion factor in large-scale conversation generation. ECM addresses the factor in three ways: modeling high-level abstraction of emotion expression by embedding emotion categories, changing of implicit internal emotion states, and using explicit emotion expressions with an external emotion vocabulary. Experiments show that our model can generate responses appropriate not only in content but also in emotion.

1 Introduction

In recent years, there has been a rising tendency in AI research to enhance Human-Computer Interaction by humanizing machines (Treur, 2007). However, to create a robot capable of acting and talking with a user at the human level requires the robot to understand human cognitive behaviors, while one of the most important human behaviors is expressing and understanding emotions and affects (Salovey and Mayer, 1990; Picard and Picard, 1997). As a vital part of human intelligence, emotional intelligence is defined as the ability to perceive, integrate, understand, and regulate emotions (Mayer and Salovey, 1997; Mayer et al., 2008). To create a machine capable to communicate with a user at the human level, it is necessary to equip the machine with emotional intelligence.

There have been prior works on endowing dialogue systems or conversational agents with emotional intelligence, through text, facial expression, and other modalities (Andre et al., 2004; Ptaszynski et al., 2009; Skowron, 2010; Skowron et al., 2011). These studies showed that addressing the issue of emotional intelligence can enhance users performance (Partala and Surakka, 2004) and users satisfaction (Prendinger et al., 2005), and lead to less breakdowns in dialogues (Martinovski and Traum, 2003). Additionally, dialogue behaviors can be adjusted to the users emotional state (Polzin and Waibel, 2000), and systems can respond to users' utterances both at the content- and affect-related levels (Skowron, 2010).

However, these studies, mostly inspired by psychology findings, are either rule-based or limited to small-scale data, and not easily extensible to large-scale data. Recently, thanks to the advance of deep learning, large-scale conversation generation approaches have been investigated (Ritter et al., 2011; Vinyals and Le, 2015; Shang et al., 2015; Serban et al., 2016). Training on large scale data collected from social media or movie subtitles, most of these works aim to improve the **content quality** issues of conversation generation systems (Gu et al., 2016; Li et al., 2015; Xing et al., 2016; Mou et al., 2016; Li et al., 2016a). However, to the best of our knowledge, the **emotion factor** has not been addressed in current large-scale conversation generation models. Table 1 shows some dialogue examples with/without emotions. We can see that our ECM model can be more emotionally empathic.

In this paper, we address the problem of generating emotional responses in conversational chatbots. We propose a model in the encoder-decoder framework of large-scale sequence-to-sequence

User: Worst day ever. I arrived late because of the traffic.
Chatbot (Basic Seq2seq): You were late. ECM (*Like*): I am always here to support you. ECM (*Happiness*): Keep smiling! Things will get better. ECM (*Sadness*): It's depressing. ECM (*Disgust*): Sometimes life just sucks. ECM (*Anger*): The traffic is too bad!

Table 1: Example conversations with/without Emotional Intelligence.

generation that can respond to users emotionally. The central idea is to model emotion expressions in conversation generation with the following additional new mechanisms: a high-level abstraction of emotion expressions is considered through embedding emotion categories and feeding into the decoder; an internal memory module is used to implicitly model the change of the internal emotion state of the decoder; and the expression of an emotion is explicitly modeled by choosing a generic (non-emotion) or emotion word during decoding.

To the best of our knowledge, this is the first work addressing the emotion factor in large-scale conversation generation. To summarize, our contributions are as follows:

- We propose an end-to-end framework to incorporate emotion influence in large-scale sequence-to-sequence conversation generation. Three novel mechanisms are proposed: emotion category embedding, an internal emotion memory, and an external memory.

- We show that ECM can generate responses with higher naturalness and emotion intelligence than the traditional seq2seq model. We believe that future work such as empathetic computer agent and emotion interaction model can be carried out based on ECM.

2 Related Work

2.1 Emotional Intelligence

Several attempts have been made to endow dialogue systems or conversational agents with emotional intelligence (Andre et al., 2004; Ptaszynski et al., 2009; Skowron, 2010; Skowron et al., 2011). In interactions between humans and artificial agents, the capability to detect signs of human emotions and to suitably react to them can enrich communication. For example, display of empathic emotional expressions enhanced users performance (Partala and Surakka, 2004), led to an increase in users satisfaction (Prendinger et al., 2005). Experiments in (Prendinger and Ishizuka, 2005) showed that an empathetic computer agent can contribute to a more positive perception of the interaction. In (Martinovski and Traum, 2003), the authors showed that many breakdowns could be avoided if the machine was able to recognize the emotional state of the user and responded to it sensitively. The work in (Polzin and Waibel, 2000) presented how dialogue behaviors can be adjusted to the users emotional state. Skowron (2010) proposed conversational systems, called affect listeners, that can respond to users' utterances both at the content- and affect-related level.

These works, mainly inspired by psychological findings, are either rule-based, or limited to small data, making them difficult to apply to large-scale dialogue or conversation generation.

2.2 Large-scale Sequence-to-sequence Based Conversation Generation

Thanks to the recent advances of deep learning, data-driven approaches are applied to open-domain conversation systems (Ritter et al., 2011; Shang et al., 2015; Serban et al., 2015). In (Ritter et al., 2011), statistical machine translation (SMT) was used to generate conversational responses from social media data. Due to the success of sequence-to-sequence generation models in machine translation (Sutskever et al., 2014; Bahdanau et al., 2014), these models were soon applied to conversation generation (Vinyals and Le, 2015), including the neural responding machine (Shang et al., 2015), the hierarchical recurrent encoder-decoder neural network (Serban et al., 2015), and many others (Sordoni et al., 2015).

To avoid generating meaningless and universal responses such as "I dont know", Li et al. (2015) proposed the maximum mutual information (MMI) principle as an alternative to maximum likelihood estimation. Beam search was also extensively used to generate meaningful and diversified responses (Li et al., 2015, 2016b; Shao et al., 2017). Other attempts to improve content quality include incorporating into the decoder additional topic words (Xing et al., 2016; Mou et al., 2016), topic categories (Xiong et al., 2016), persona information (Li et al., 2016a), or other retrieved responses (Song et al., 2016). Another path to avoid meaningless response is to apply deep reinforcement learning which can model the long-term delayed reward in chatbot's dialogues (Li et al., 2016c; Ranzato et al., 2015). Dealing with unknown words can improve the generation quality too. Duplicating low-frequency words from post to response was proposed in (Gu et al., 2016).

These works have been done to improve the content quality generation, however, no work has addressed the emotion factor. Our work aims to generate responses both relevant in content and coherent in emotion.

2.3 Memory-based Networks

Memory Network (Weston et al., 2014) and Neural Turing Machine (NTM) (Graves et al., 2014) augment traditional RNNs with memory structures to improve the ability of modeling long-range sequences. Inspired by these models, many other memory networks have been proposed for tasks such as machine translation (Meng et al., 2015), question answering (Miller et al., 2016) and dialogue state tracking (Perez and Liu, 2016). Our model adopts a dynamic memory to model the change of an internal emotion state, and a static memory to store a dictionary of emotion words.

3 Background: Sequence-to-sequence Generation

Before presenting the proposed model, we first introduce a general encoder-decoder framework based on sequence-to-sequence (seq2seq) learning (Sutskever et al., 2014), which will be adopted for our work. The encoder and decoder of the seq2seq model (Sutskever et al., 2014) are implemented with GRU (Cho et al., 2014; Chung et al., 2014).

The encoder converts the post sequence $\boldsymbol{X} = (x_1, x_2, \cdots, x_n)$ to hidden representations $\boldsymbol{h} = (h_1, h_2, \cdots, h_n)$, which is briefly defined as:

$$h_t = \mathbf{GRU}(h_{t-1}, x_t) \tag{1}$$

Details of GRU can be found in (Cho et al., 2014).

The decoder takes a context vector c_t and the embedding of a previously decoded word $e(y_{t-1})$, and updates its state s_t using another GRU:

$$s_t = \mathbf{GRU}(s_{t-1}, [c_t; e(y_{t-1})]) \tag{2}$$

where $[c_t; e(y_{t-1})]$ is concatenation of the two vectors, serving as input to the GRU network. The context vector c_t is designed to dynamically attend on key information of the input post during decoding (Bahdanau et al., 2014). Formally, c_t depends on the previous state of the decoder s_{t-1}:

$$c_t = \sum_{k=1}^{n} \alpha_{tk} h_k \tag{3}$$

$$\alpha_{tk} = \frac{exp(e_{tk})}{\sum_{j=1}^{n} exp(e_{tj})} \tag{4}$$

$$e_{tk} = v_a^\top tanh(W_a s_{t-1} + U_a h_k) \tag{5}$$

where α_{tk} can be viewed as the similarity score between each hidden state h_k of the post and the decoder's state s_{t-1}.

Once the state vector s_t is obtained, the decoder generates a token by sampling from the output probability distribution o_t computed from the decoder's state s_t, as follows:

$$y_t \sim o_t = P(y_t \mid y_1, y_2, \cdots, y_{t-1}, c_t) \tag{6}$$

$$= softmax(W_o s_t) \tag{7}$$

4 Emotional Chatting Machine

4.1 Task Definition and Overview

The problem we deal with in this paper is formulated as follows: Given a post $\boldsymbol{X} = (x_1, x_2, \cdots, x_n)$, and a user-specified emotion category e of the response to be generated, the goal is to generate a response $\boldsymbol{Y} = (y_1, y_2, \cdots, y_m)$ that is coherent with the emotion category e. Essentially, the model estimates the probability: $P(\boldsymbol{Y}|\boldsymbol{X}, e) = \prod_{t=1}^{m} P(y_t|y_{<t}, \boldsymbol{X}, e)$. The emotion categories are $\{$*Anger, Disgust, Happiness, Like, Sadness, Other*$\}$, adopted from a Chinese emotion classification challenge task.[1] In order to train our model, $\boldsymbol{X}/\boldsymbol{Y}$ will be automatically annotated with emotion categories by an emotion classifier.

Built upon the sequence-to-sequence (seq2seq) generation framework discussed in the previous section. Emotional Chatting Machine (ECM) generates an emotion expression using three mechanisms: First, since the emotion category is a high-level abstraction of an emotion expression, ECM embeds the emotion category and takes the emotion category embedding as input to the decoder; Second, we assume that during decoding, there is an internal emotion state, and in order to model

[1]The taxonomy comes from http://tcci.ccf.org.cn/conference/2014/dldoc/evatask1.pdf

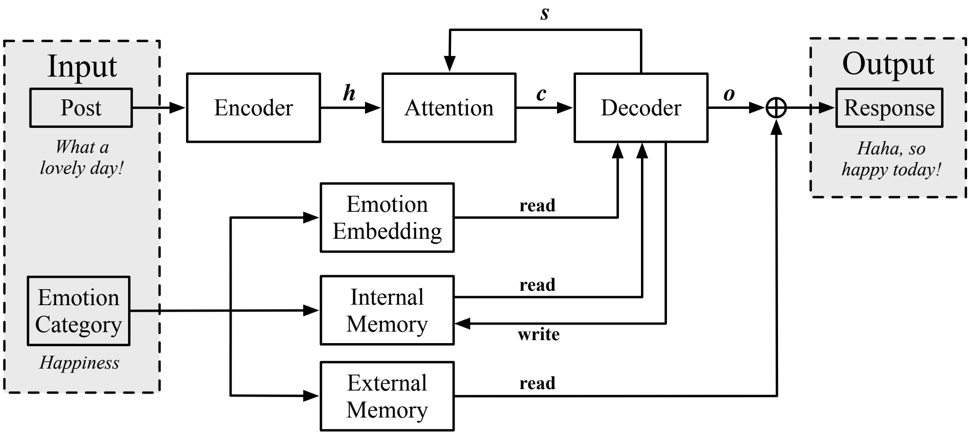

Figure 1: The overview of ECM. $\boldsymbol{h}$ is the hidden representation of an input post, $\boldsymbol{c}$ is a context vector generated by attention conditioned on $\boldsymbol{h}$ and the state of the decoder $\boldsymbol{s}$, $\boldsymbol{o}$ is the decoding probability distribution. Emotion embedding is a vector representation of the input category e, internal memory is a matrix storing the internal emotion states, and external memory is an emotion dictionary for decoding.

the implicit change of the state, ECM adopts an internal memory module; Last, explicit expression of an emotion is modeled through explicit selection of a common or emotion word by an external memory module.

An overview of ECM is given in Figure 1. The decoder has several inputs: the context vector $\boldsymbol{c}$, and, the emotion embedding vector and the internal memory both indexed by the emotion category. After combined with an external memory module, the decoder obtains a probability distribution $\boldsymbol{o}$ (see Eq. 17) for response generation.

4.2 Emotion Category Embedding

Since an emotion category provides a high-level abstraction of an expression of the emotion, the most intuitive approach to model emotion in response generation is to take as additional input the emotion category of a response to be generated. Each emotion category is represented by a real-valued, low dimensional vector. For each emotion category e, we randomly initialize the vector of the emotion category v_e, and then learn the emotion category representations through training. The emotion category embedding v_e, along with word embedding $e(y_{t-1})$, and the context vector c_t, are fed into the decoder to update the decoder's state s_t:

$$s_t = \mathbf{GRU}(s_{t-1}, [c_t; e(y_{t-1}); v_e]) \qquad (8)$$

Based on s_t, the decoding probability distribution can be updated accordingly as Eq. 7 to generate the next token y_t.

4.3 Internal Memory

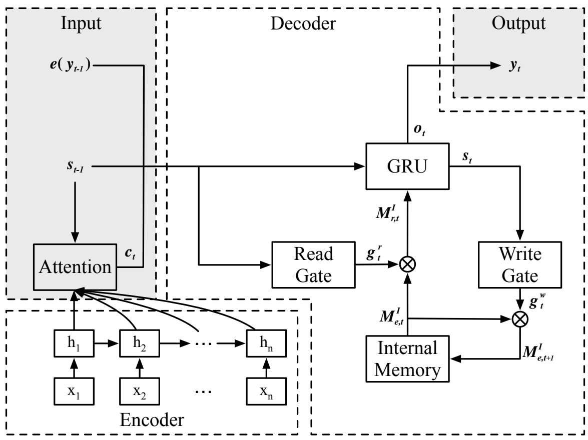

Figure 2: Data flow of the decoder with an internal memory. The internal memory $M^I_{e,t}$ is read with the read gate g^r_t by an amount $M^I_{r,t}$ to update the decoder's state, and the memory is updated to $M^I_{e,t+1}$ with the write gate g^w_t.

The method presented in the preceding section is rather static: the emotion category embedding will not change during the token generation process. We assume that there is an internal emotion state during the decoding process, and the state changes as the decoding process proceeds. Hence, we design an internal memory module to approach the emotion dynamics during decoding. We simulate the process as follows: there is an internal emotion state for each category before the decoding process starts; at each step of the decoding process, the emotion state decays by a certain amount; once the decoding process stops, the emotion state should decay to zero.

The detailed process of the internal memory module is illustrated in Figure 2. At each step t, the read gate g_t^r takes as input the word embedding of the previously decoded word $e(y_{t-1})$, the previous state of the decoder s_{t-1}, and the current context vector c_t. The write gate g_t^w computes a write vector from the decoder's state vector s_t. The read and write vector are then used to read from and write into the internal memory. Hence, the emotion state is erased by a certain amount (by g_t^w) at each step. At the last step, the internal emotion state will decay to zero. This process is formally described as below:

$$g_t^r = \sigma(W_g^r[e(y_{t-1}); s_{t-1}; c_t]) \quad (9)$$

$$g_t^w = \sigma(W_g^w s_t) \quad (10)$$

$$M_{r,t}^I = g_t^r \odot M_{e,t}^I \quad (11)$$

$$M_{e,t+1}^I = g_t^w \odot M_{e,t}^I \quad (12)$$

where r/w denotes read/write respectively, and I means Internal. With the input of the previous target word $e(y_{t-1})$, the previous state of the decoder s_{t-1}, the context vector c_t, and the emotion state update $M_{r,t}^I$, GRU outputs the current state s_t,

$$s_t = \mathbf{GRU}(s_{t-1}, [c_t; e(y_{t-1}); M_{r,t}^I]) \quad (13)$$

With the state, the word generation distribution o_t can be obtained with Eq. 7, and the next word y_t can be sampled. After generating the next word, $M_{e,t+1}^I$ is written back to the internal memory.

It is noteworthy that the write vector works as a DELETE operation to model the expression of an emotion in generation of a response, which is quite different from the ADD and DELETE operation used in previous memory networks (Meng et al., 2015; Miller et al., 2016; Perez and Liu, 2016).

4.4 External Memory

In the internal memory module, the correlation between the change of the internal emotion state and choosing of a word is **implicit and not directly observable**. As most emotional responses contain emotion words such as *lovely* and *awesome* which indicate strong emotions compared to generic (non-emotion) words such as *person* and *day*, we propose an external memory model to model emotion expressions **explicitly** by assigning different generation probabilities to emotion words and generic words. Thus the model can choose to generate words from a generic vocabulary or an emotion vocabulary (Xu et al., 2008).

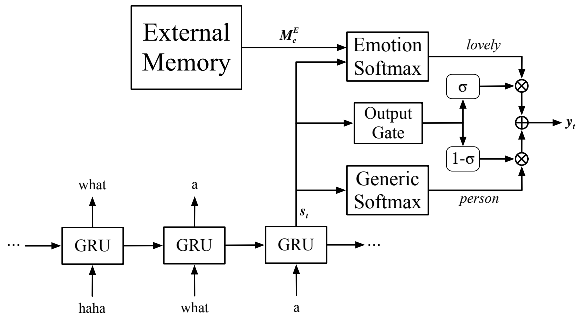

Figure 3: Data flow of the decoder with an external memory. The final decoding probability is weighted between the emotion softmax and the generic softmax, where the weight is computed by the output gate.

The decoder with an external memory is illustrated in Figure 3. Given the current state of the decoder s_t, the emotion softmax $P_e(y_t = w_e)$ and the generic softmax $P_g(y_t = w_g)$ are computed over the emotion vocabulary which is read from the external memory and generic vocabulary, respectively. The output gate g_t^o controls the weights of generating an emotion or a generic word. Finally, the next word y_t is sampled from the next word probability, and the concatenation of the two weighted probabilities. The process can be formulated as follows:

$$g_t^o = \sigma(v_u^\top s_t) \quad (14)$$

$$P_g(y_t = w_g) = softmax(W_g^o s_t) \quad (15)$$

$$P_e(y_t = w_e) = softmax(W_e^o s_t) \quad (16)$$

$$\boldsymbol{o}_t = P(y_t) = \begin{bmatrix} (1 - g_t^o)P_g(y_t = w_g) \\ g_t^o P_e(y_t = w_e) \end{bmatrix} \quad (17)$$

where $g_t^o \in [0, 1]$ is a scalar to balance the choice between an emotion word w_e and a generic word w_g, P_g/P_e is the distribution over generic/emotion words respectively, and $P(y_t)$ is the final word decoding distribution. Note that the two vocabularies have no intersection, and the final distribution $P(y_t)$ is a concatenation of two distributions.

4.5 Loss Function

The loss function is the cross entropy error between the predicted token distribution o_t and the gold distribution p_t in the training corpus. Additionally, we apply two regularization terms: one on the internal memory, enforcing that the internal emotion state should decay to zero at the end of decoding, and the other on the external memory, constraining the selection of an emotional or generic word. The loss on one sample $< \boldsymbol{X}, \boldsymbol{Y} >$ ($\boldsymbol{X} = x_1, x_2, ..., x_n$, $\boldsymbol{Y} = y_1, y_2, ..., y_m$) is defined as:

$$L(\theta) = -\sum_{t=1}^{m} p_t log(o_t) - \sum_{t=1}^{m} q_t log(g_t^o) + \parallel M_{e,m}^I \parallel \tag{18}$$

where $M_{e,m}^I$ is the internal emotion state at the last step m, g_t^o is the probability of choosing an emotion word or a generic word, and $q_t \in \{0, 1\}$ is the true choice of an emotion word or a generic word in $\boldsymbol{Y}$. The second term is used to supervise the probability of selecting an emotion or generic word. And the third term is used to ensure that the internal emotion state has been expressed completely once the generation is completed.

5 Datasets and Preparation

5.1 Data Collection

We collected several Weibo datasets:

- **The STC dataset**: a conversation dataset from (Shang et al., 2015), where each post has multiple responses. It has 219,905 posts and 4,308,211 responses. The ratio of post to response is about 1:20. This dataset will be automatically annotated by an emotion classifier and then used to train our model. We also collected 9,698,728 one-to-one post-response pairs from Weibo for pretraining.
- **The NLPCC dataset**: the Emotion Analysis Dataset of NLPCC 2013[2]&2014[3], a benchmark dataset for emotion classification, consisting of 23,105 sentences collected from Weibo. The dataset is manually annotated with the emotion categories *Anger*, *Disgust*, *Fear*, *Happiness*, *Like*, *Sadness* and *Surprise*, and is used to train the emotion classifier.

5.2 Emotion Classifier

To annotate the large-scale STC Dataset with emotion categories, we compare several models for emotion classification, including a dictionary based classifier (denoted by Dict in Table 2), RNN (Mikolov et al., 2010), LSTM (Hochreiter and Schmidhuber, 1997), and Bidirectional LSTM (Bi-LSTM) (Graves et al., 2005). After removing the infrequent classes including *Fear* and *Surprise* of which the proportion are 1.5% and 4.4% respectively, we partition the NLPCC dataset into the training, validation, and test set with the ratio of 8:1:1. Finally, the accuracy of 6-class (*Anger*, *Disgust*, *Happiness*, *Like*, *Sadness* and *Other*) classification is shown in Table 2.

Method	Accuracy
Dict	0.432
RNN	0.564
LSTM	0.594
Bi-LSTM	0.623

Table 2: The accuracy of emotion classifiers on the NLPCC dataset.

Results show that all neural classifiers outperform the dictionary-based classifier, and the Bi-LSTM classifier obtains the best performance of 0.623. Although this accuracy may introduce errors in the classification of emotion categories, we found that it is sufficient for our models to generate good emotional responses in practice. As future work, we will consider how the classification errors influence response generation.

5.3 Weibo Emotion Dataset

Training	**posts**	217,905	
	responses	*Anger*	234,635
		Disgust	689,295
		Happiness	306,364
		Like	1,226,954
		Sadness	537,028
		Other	1,365,371
Validation	**posts**	1,000	
Test	**posts**	1,000	

Table 3: Statistics of the Weibo Emotion Dataset.

We apply the Bi-LSTM emotion classifier to annotate the STC Dataset with emotion categories. To build the validation and test sets, we randomly sample 1000 posts respectively, and these samples don't contain low frequency words and have more than 20 responses.

After automated annotation, we call this dataset the *Weibo Emotion Dataset* which is used to train our ECM models. The statistics of the Weibo Emotion Dataset are shown in Table 3.

[2]http://tcci.ccf.org.cn/conference/2013/
[3]http://tcci.ccf.org.cn/conference/2014/

6 Experiments

6.1 Implementation Details

We use Tensorflow(Abadi et al., 2016) to implement the proposed model. The encoder and decoder have 2-layer GRU structures with 256 hidden cells for each layer and use different sets of parameters respectively. The word embedding size is set to 100. The vocabulary size is limited to 40,000. The embedding size of emotion category is set to 100. The internal memory is a trainable matrix of size 6×256 and the external memory is a list of 40,000 words containing generic words and emotion words (but emotion words have different markers). Parameters are initialized by sampling from the uniform distribution $(-sqrt(3/n), sqrt(3/n))$, where n represents the dimension of parameters. To generate diverse responses, we adopt beam search in the decoding process of which the beam size is set to 20. And then rerank responses by the generation probability after removing those containing *UNK*s, unknown words.

We use the stochastic gradient descent (SGD) algorithm with mini-batch. Batch size and learning rate are set to 128 and 0.5, respectively. To accelerate the training process, we train a seq2seq model on the Weibo pretraining dataset with pre-trained word embeddings. And we then train our model on the Weibo Emotion Dataset with parameters initialized by the parameters of the pre-trained seq2seq model. We run 20 epoches, and the training stage of each model took about a week on a Titan X GPU machine.

6.2 Quantitative Evaluation

Method	Perplexity	Coherence
Seq2Seq	68.0	-
Emb	62.5	0.724
IMem	61.8	0.731
EMem	65.9	0.773

Table 4: Quantitative evaluation in terms of perplexity and emotion coherence.

As argued in (Liu et al., 2016), BLEU is not suitable for measuring conversation generation. Instead, we adopt perplexity to evaluate the model at the content level (whether the content is relevant and grammatical). To evaluate the model at the emotion level, we adopt the coherence between the expected emotion category (as input to the model) and the predicted emotion category of a generated response by the emotion classifier as mentioned in Section 5.2.

We compare the seq2seq model (Sutskever et al., 2014) as the baseline with our ECM models. The three options including emotion category embedding (Emb), internal memory (IMem), and external memory (EMem) are **incrementally** added to the basic model. In other words, EMem in the tables means all three options have been included. We choose the model with the best perplexity on the validation set to evaluate the two metrics of perplexity and coherence in the test set. To evaluate the performance of ECM models at the emotion level, we generate emotional responses using beam search and evaluate the emotion coherence of the top 5 responses for each model on the 1,000 test set.

The results are shown in Table 4. As it can be seen, the perplexity of our models is much lower than that of the seq2seq model; the emotion coherence is much higher correspondingly. Note that emotion coherence measures the agreement between the pre-specified emotion category and the predicted category by the emotion classifier, which may be influenced by the errors of the classifier. However, as shown in the table, better coherence is observed when more model options are incrementally added.

Although the EMem model obtains the best performance in emotion coherence, the perplexity of this model is worse than the other two settings after incorporating the external memory. This may be due to the fact that the loss function of the EMem model is not only supervised on perplexity, but also on the selection of generic or emotion words (see Eq.18).

6.3 Human Evaluation

Method	Naturalness	Emotion
Seq2Seq	2.678	-
Emb	2.655	1.947
IMem	2.719	1.966
EMem	2.700	2.016

Table 5: Human evaluation for the quality of responses on two metrics (maximal score is 3.0).

In order to better understand the quality of generated responses from the content and emotion perspectives, we recruit 5 judges for human eval-

Post	Model		Response(Original)	Response(Translated)
比情人节更庸俗的事就是，对情人节的接待规格待遇提出要求。 What is more vulgar than Valentine's day is to make specific demands on the reception of the Valentine's day.	Seq2Seq		这才是真正的情人节。	This is a real Valentine's day.
	ECM	*Like*	情人节快乐！	Happy Valentine's Day!
		Happiness	哈哈，这也太浪漫了吧！	Aha, this is too romantic!
		Sadness	我也想要这样的情人节。	I also want this kind of Valentine's day, sigh.
		Disgust	这就是所谓的情人节！	This is the so-called Valentine's day!
		Anger	这是赤裸裸的炫耀啊！	This is shameless show-off!
知道真相的我眼泪笑下来。 Knowing the truth, I laughed myself to tears.	Seq2Seq		知道真相的我眼泪掉下来。	After knowing the truth, I burst into tears.
	ECM	*Like*	太可爱了，太可爱了！	Too lovely, it's too lovely.
		Happiness	笑得我眼泪都出来了。	I can't help laugh myself to tears.
		Sadness	知道真相的我眼泪掉下来。	After knowing the truth, I burst into tears.
		Disgust	这是真的还是假的？	Is this true?
		Anger	什么意思？没看懂。	What do you mean? I don't understand.

Figure 4: Sample responses generated by the seq2seq model and ECM (original Chinese and English translation). The corresponding posts did not appear in the training set.

Pref.%	Seq2Seq	Emb	IMem	EMem
Seq2Seq	-	2.6	1.6	0.8
Emb	97.4	-	34.2	32.6
IMem	98.4	65.8	-	38.2
EMem	99.2	67.4	61.8	-

Table 6: Pairwise preference of the four systems.

uation experiments. Given the post and an emotion category, the seq2seq model and ECM models are used to generate one response, and the top response generated by beam search for each model was presented to human judges. Judges were asked to score each response in terms of naturalness and emotion (rating scale is 1,2,3), and also to state a preference between any two of the four systems. The naturalness is defined as whether the response could plausibly have been produced by a human and the emotion is defined as whether the emotion expression of a response is coherent and appropriate to the given emotion category. We test the same 500 posts extracted from the test set, for each setting of our models, we generate 2533 emotional responses, and additional 500 responses for the seq2seq model, which amounts to 8099 responses in total.

The results of human evaluation for the quality of response are shown in Table 5. As can be seen, EMem outperforms the other methods in both metrics. Besides, ECM models are much more preferred than the seq2seq model and EMem is most preferred by judges compared to other methods as shown in Table 6.

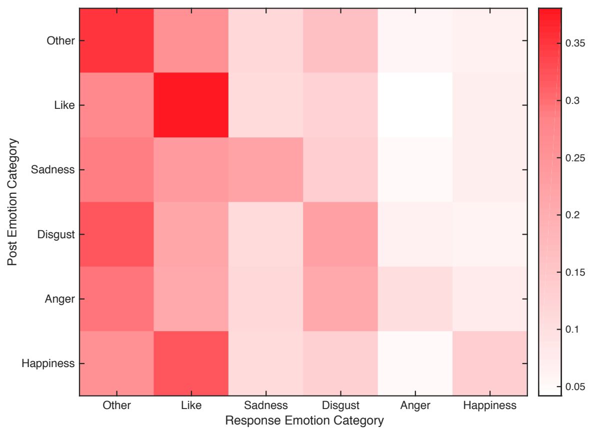

Figure 5: Visualization of emotion interaction.

6.4 Analysis of Emotion Interaction

We present some sample responses in Figure 4. We can see that ECM can generate responses that are appropriate both in content and in emotion if the pre-specified emotion category and the emotion of the post belong to one of frequent emotion interaction patterns. Emotion interaction pattern (EIP) is defined as $< e_p, e_r >$, the pair of emotion categories of a post and its response. The value of an EIP is the conditional probability $P(e_r|e_p) = P(e_r, e_p)/P(e_p)$. For low-valued EIPs such as $< \mathit{Happiness}, \mathit{Disgust} >$ and $< \mathit{Happiness}, \mathit{Anger} >$ as shown in last two lines of Figure 4, responses are less coherent to the emotion category due to the lack of training data and/or the errors caused by the emotion classifier./Users/tuxchow/Downloads/ZhouHao-ACL2017 (arxiv)/acl2017.bib

Figure 5 visualizes the emotion interaction patterns between post and response in the training data whose label is made by our emotion classifier. The color darkness indicates the strength of an EIP. Each cell indicates an EIP, for instance, the cell of row 6 and column 2 indicates $< \mathit{Happiness}, \mathit{Like} >$. Frequent EIPs show that there exists a major responding emotion given the post emotion category. For instance, when a post expresses *Happiness*, the emotion of responses is typically *Like*. The diagonal patterns indicate emotional empathy which is a common type of emotion interaction. Note that class *other* has much more data than other classes (see Table 3), and thus EIP is biased to this class (the first column of Figure 5), which may be biased by the data and the errors of the classifier. Also, note that the analysis highly depends on the results of an automated classifier but not manually labeled data.

7 Conclusion

In this paper, we proposed the emotional chatting machine (ECM) to model the emotion influence in large-scale conversation generation. Three mechanisms have been proposed to model the emotion factor, including emotion category embedding, internal emotion memory, and external memory. Automatic and manual evaluation shows that ECM can generate responses appropriate not only in content but also in emotion. Preference tests show that our proposals are more preferred than the seq2seq model. As future work, we will explore the emotion interaction patterns based on ECM.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* .

Elisabeth Andre, Matthias Rehm, Wolfgang Minker, and Dirk Bühler. 2004. Endowing spoken language dialogue systems with emotional intelligence. In *Tutorial and Research Workshop on Affective Dialogue Systems*. Springer, pages 178–187.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* .

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*. Springer, pages 799–804.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401* .

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055* .

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155* .

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. A simple, fast diverse decoding algorithm for neural generation. *CoRR* abs/1611.08562.

Jiwei Li, Will Monroe, Alan Ritter, and Dan Jurafsky. 2016c. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541* .

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. pages 2122–2132.

Bilyana Martinovski and David Traum. 2003. Breakdown in human-machine interaction: the error is the clue. In *Proceedings of the ISCA tutorial and research workshop on Error handling in dialogue systems*. pages 11–16.

John D Mayer, Richard D Roberts, and Sigal G Barsade. 2008. Human abilities: Emotional intelligence. *Annu. Rev. Psychol.* 59:507–536.

John D Mayer and Peter Salovey. 1997. What is emotional intelligence? *Emotional Development and Emo- tional Intelligence* pages 3–31.

Fandong Meng, Zhengdong Lu, Zhaopeng Tu, Hang Li, and Qun Liu. 2015. A deep memory-based architecture for sequence-to-sequence learning. *arXiv preprint arXiv:1506.06442* .

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. volume 2, page 3.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126* .

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *26th International Conference on Computational Linguistics, Proceedings of the Conference*. pages 3349–3358.

Timo Partala and Veikko Surakka. 2004. The effects of affective interventions in human–computer interaction. *Interacting with computers* 16(2):295–309.

Julien Perez and Fei Liu. 2016. Dialog state tracking, a machine reading approach using memory network. *arXiv preprint arXiv:1606.04052* .

Rosalind W Picard and Roalind Picard. 1997. *Affective computing*, volume 252. MIT press Cambridge.

Thomas S Polzin and Alexander Waibel. 2000. Emotion-sensitive human-computer interfaces. In *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*.

Helmut Prendinger and Mitsuru Ishizuka. 2005. The empathic companion: A character-based interface that addresses users'affective states. *Applied Artificial Intelligence* 19(3-4):267–285.

Helmut Prendinger, Junichiro Mori, and Mitsuru Ishizuka. 2005. Using human physiology to evaluate subtle expressivity of a virtual quizmaster in a mathematical game. *International journal of human-computer studies* 62(2):231–245.

Michal Ptaszynski, Pawel Dybala, Wenhan Shi, Rafal Rzepka, and Kenji Araki. 2009. Towards context aware emotional intelligence in machines: Computing contextual appropriateness of affective states. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*. pages 1469–1474.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR* abs/1511.06732.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John Mcintyre Conference Centre, Edinburgh, Uk, A Meeting of Sigdat, A Special Interest Group of the ACL*. pages 583–593.

Peter Salovey and John D Mayer. 1990. Emotional intelligence. *Imagination, cognition and personality* 9(3):185–211.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808* .

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. *Computer Science* .

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *Computer Science* .

Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating long and diverse responses with neural conversation models. *CoRR* abs/1701.03185.

Marcin Skowron. 2010. Affect listeners: Acquisition of affective states by means of conversational systems. In *Development of Multimodal Interfaces: Active Listening and Synchrony*, Springer, pages 169–181.

Marcin Skowron, Stefan Rank, Mathias Theunis, and Julian Sienkiewicz. 2011. The good, the bad and the neutral: affective profile in dialog system-user communication. In *International Conference on Affective Computing and Intelligent Interaction*. Springer, pages 337–346.

Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149* .

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. pages 196–205.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Jan Treur. 2007. On human aspects in ambient intelligence. In *European Conference on Ambient Intelligence*. Springer, pages 262–267.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *Computer Science* .

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* .

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2016. Topic aware neural response generation. *arXiv preprint arXiv:1606.08340* .

Kun Xiong, Anqi Cui, Zefeng Zhang, and Ming Li. 2016. Neural contextual conversation learning with labeled question-answering pairs. *CoRR* abs/1607.05809.

Linhong Xu, Hongfei Lin, Yu Pan, Hui Ren, and Jianmei Chen. 2008. Affective lexicon ontology. *Journal of information* 27(2):180–185.