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 100084, PR China
[‡]Dept. of Computer Science, University of Illinois at Chicago, Chicago, Illinois, USA

tuxchow@gmail.com, keavilzhangzty@gmail.com, zxy-

Abstract

Perception and expression of emotion are key factors to the success of dialogue systems or conversational agents. However, this problem has not been studied in large-scale conversation generation so far. In this paper, we propose Emotional Chatting Machine (ECM) that can generate appropriate responses not only in content (relevant and grammatical) but also in emotion (emotionally consistent). 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 using three new mechanisms that respectively (1) models the high-level abstraction of emotion expressions by embedding emotion categories, (2) captures the change of implicit internal emotion states, and (3) uses explicit emotion expressions with an external emotion vocabulary. Experiments show that the proposed model can generate responses appropriate not only in content but also in emotion.

Introduction

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). It has been a long-term goal of artificial intelligence to enable a machine to understand affect and emotion (Picard and Picard 1997). To create a chatbot capable of communicating with a user at the human level, it is necessary to equip the machine with the ability of perceiving and expressing emotions.

Existing studies show that addressing affect and emotion in dialogue systems or conversational agents can enhance user satisfaction (Prendinger, Mori, and Ishizuka 2005),

and lead to fewer breakdowns in dialogues (Martinovski and Traum 2003). Some initial work has also been done on adjusting dialogue behaviors to suit users' emotional states (Polzin and Waibel 2000), and on generating responses to users' utterances at both 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. Recently, neural models trained on large-scale data

*Corresponding author: Minlie Huang, aihuang@tsinghua.edu
.cn

aihuang@tsinghua.edu.cn,
-dcs@tsinghua.edu.cn, liub@cs.uic.edu

have advanced open-domain conversation generation significantly (Ritter, Cherry, and Dolan 2011; Vinyals and Le 2015; Shang, Lu, and Li 2015; Serban et al. 2016). Most of these models aim to improve the **content quality** of conversation generation (Gu et al. 2016; Li et al. 2016a; Xing et al. 2017; Mou et al. 2016; Li et al. 2016b). To the best of our knowledge, the **emotion factor** has not been addressed in existing neural models for conversation generation. Table 1 shows some examples with/without considering emotions. We can see that our ECM model is emotionally involved and more empathetic.

User: Worst day ever. I arrived late because of the traffic.
Basic Seq2Seq: You were late.
ECM (*Like*): I am always here to support you.
ECM (*Happy*): Keep smiling! Things will get better.
ECM (*Sad*): It's depressing.
ECM (*Disgust*): Sometimes life just sucks.
ECM (*Angry*): The traffic is too bad!

Table 1: Conversations with/without considering emotion.

There are several challenges in addressing the emotion factor in large-scale conversation generation. **First**, high-quality emotion-labeled data are difficult to obtain in a large-scale corpus, as emotion annotation is a fairly subjective task and emotion classification is also challenging. **Second**, it is difficult to consider emotions in a natural and coherent way because we need to balance grammaticality and expressions of emotions, as argued in (Ghosh et al. 2017). **Last**, simply embedding emotion information in existing neural models, as shown in our experiments, cannot produce desirable emotional responses but just hard-to-perceive general expressions (which contain only common words that are quite implicit or ambiguous about emotions, and amount to 73.7% of all emotional responses in our dataset).

In this paper, we address the problem of generating emotional responses in open-domain conversational systems and propose an emotional chatting machine (ECM for short). To obtain large-scale emotion-labeled data for ECM, we train a neural classifier on a manually annotated corpus. The classifier is used to annotate large-scale conversation data automatically for the training of ECM. To express emotion naturally and coherently in a sentence, we design a sequence-to-sequence generation model equipped with new mechanisms for emotion expression generation, namely, emotion category embedding for capturing high-level abstraction of emotion expressions, an internal emotion state for balancing grammaticality and emotion dynamically, and an external emotion memory to help generate more explicit and unambiguous emotional expressions.

In summary, this paper makes the following contributions:

- It proposes to address the emotion factor in large-scale conversation generation. To the best of our knowledge, this is the first work on the topic.
- It proposes an end-to-end framework (called ECM) to incorporate the emotion influence in large-scale conversation generation. It has three novel mechanisms: emotion category embedding, an internal emotion memory, and an external memory.
- It shows that ECM can generate responses with higher content and emotion scores than the traditional seq2seq model. We believe that future work such as the empathetic computer agent and the emotion interaction model can be carried out based on ECM.

Related Work

In human-machine interactions, the ability to detect signs of human emotions and to properly react to them can enrich communication. For example, display of empathetic emotional expressions enhanced users' performance (Partala and Surakka 2004), and led to an increase in user satisfaction (Prendinger, Mori, and Ishizuka 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 users' emotional states. 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 conversation generation. Recently, sequence-to-sequence generation models (Sutskever, Vinyals, and Le 2014; Bahdanau, Cho, and Bengio 2014) have been successfully applied to large-scale conversation generation (Vinyals and Le 2015), including neural responding machine (Shang, Lu, and Li 2015), hierarchical recurrent models (Serban et al. 2015), and many others. These models focus on improving the content quality of the generated responses, including diversity promotion (Li et al. 2016a), considering additional information (Xing et al. 2017; Mou et al. 2016; Li et al. 2016b; Herzig et al. 2017), and handing unknown words (Gu et al. 2016).

However, no work has addressed the emotion factor in large-scale conversation generation. There are several studies that generate text from controllable variables. (Hu et al. 2017) proposed a generative model which can generate sentences conditioned on certain attributes of the language such as sentiment and tenses. Affect Language Model was proposed in (Ghosh et al. 2017) to generate text conditioned on context words and affect categories. (Cagan, Frank, and Tsarfaty 2017) incorporated the grammar information to generate comments for a document using sentiment and topics. Our work is different in two main aspects: 1) prior studies are heavily dependent on linguistic tools or customized parameters in text generation, while our model is fully data-driven without any manual adjustment; 2) prior studies are unable to model multiple emotion interactions between the input post and the response, instead, the generated text simply continues the emotion of the leading context.

Emotional Chatting Machine

Background: Encoder-decoder Framework

Our model is based on the encoder-decoder framework of the general sequence-to-sequence (seq2seq for short) model (Sutskever, Vinyals, and Le 2014). It is implemented with gated recurrent units (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 $\boldsymbol{h} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n)$, which is defined as:

$$\boldsymbol{h}_t = \mathbf{GRU}(\boldsymbol{h}_{t-1}, x_t). \quad (1)$$

The decoder takes as input a context vector $\boldsymbol{c}_t$ and the embedding of a previously decoded word $\boldsymbol{e}(y_{t-1})$ to update its state $\boldsymbol{s}_t$ using another GRU:

$$\boldsymbol{s}_t \quad = \quad \mathbf{GRU}(\boldsymbol{s}_{t-1}, [\boldsymbol{c}_t; \boldsymbol{e}(y_{t-1})]), \qquad (2)$$

where $[\boldsymbol{c}_t; \boldsymbol{e}(y_{t-1})]$ is the concatenation of the two vectors, serving as the input to the GRU cell. The context vector $\boldsymbol{c}_t$ is designed to dynamically attend on key information of the input post during decoding (Bahdanau, Cho, and Bengio 2014). Once the state vector $\boldsymbol{s}_t$ is obtained, the decoder generates a token by sampling from the output probability distribution $\boldsymbol{o}_t$ computed from the decoder's state $\boldsymbol{s}_t$ as follows:

$$\begin{aligned} y_t \sim \boldsymbol{o}_t &= P(y_t \mid y_1, y_2, \cdots, y_{t-1}, \boldsymbol{c}_t), &(3)\\ &= \mathrm{softmax}(\mathbf{W_o}\boldsymbol{s}_t). &(4) \end{aligned}$$

Task Definition and Overview

Our problem is formulated as follows: Given a post $\boldsymbol{X} = (x_1, x_2, \cdots, x_n)$ and an emotion category e of the response to be generated (explained below), 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 {*Angry, Disgust, Happy, Like, Sad, Other*}, adopted from a Chinese emotion classification challenge task.[1]

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

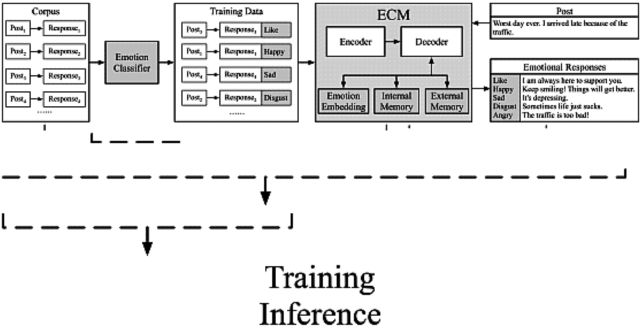

Figure 1: Overview of ECM (the grey unit). The pink units are used to model emotion factors in the framework.

In our problem statement, we assume that the emotion category of the to-be-generated response is given, because emotions are highly subjective. Given a post, there may be multiple emotion categories that are suitable for its response, depending on the attitude of the respondent. For example, for a sad story, someone may respond with sympathy (as a friend), someone may feel angry (as an irritable stranger), yet someone else may be happy (as an enemy). Flexible emotion interactions between a post and a response are an important difference from the previous studies (Hu et al. 2017; Ghosh et al. 2017; Cagan, Frank, and Tsarfaty 2017), which use the same emotion or sentiment for response as that in the input post.

Thus, due to this subjectivity of emotional responses, we choose to focus on solving the **core problem**: generating an emotional response given a post and an emotion category of the response. Our model thus works regardless the response emotion category. Note that there can be multiple ways to enable a chatbot to choose an emotion category for response. One way is to give the chatbot a personality and some background knowledge. Another way is to use the training data to find the most frequent response emotion category for the emotion in the given post and use that as the response emotion. This method is reasonable as it reflects the general emotion of the people. We leave this study to our future work.

Building upon the generation framework discussed in the previous section, we propose the Emotional Chatting Machine (ECM) to generate emotion expressions using three mechanisms: **First**, since the emotion category is a high-level abstraction of an emotion expression, ECM embeds the emotion category and feeds the emotion category embedding to the decoder. **Second**, we assume that during decoding, there is an internal emotion state, and in order to capture the implicit change of the state and to balance the weights between the grammar state and the emotion state dynamically, ECM adopts an internal memory module. **Third**, an explicit expression of an emotion is modeled through an explicit selection of a generic (non-emotion) or emotion word by an external memory module.

An overview of ECM is given in Figure 1. In the training process, the corpus of post-response pairs is fed to an emotion classifier to generate the emotion label of each response, and then ECM is trained on the data of triples: posts, responses and emotion labels of responses. In the inference process, a post is fed to ECM to generate emotional responses conditioned on different emotion categories.

Emotion Category Embedding

Since an emotion category (for instance, *Angry, Disgust, Happy*) provides a high-level abstraction of an emotion expression, the most intuitive approach to modeling 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 an emotion category $\boldsymbol{v}_e$, and then learn the vectors of the emotion category through training. The emotion category embedding $\boldsymbol{v}_e$, along with word embedding $\boldsymbol{e}(y_{t-1})$, and the context vector $\boldsymbol{c}_t$, are fed into the decoder to update the decoder's state $\boldsymbol{s}_t$:

$$\boldsymbol{s_t} = \mathbf{GRU}(\boldsymbol{s}_{t-1}, [\boldsymbol{c}_t; \boldsymbol{e}(y_{t-1}); \boldsymbol{v}_e]). \tag{5}$$

Based on $\boldsymbol{s}_t$, the decoding probability distribution can be computed accordingly by Eq. 4 to generate the next token y_t.

Internal Memory

The method presented in the preceding section is rather **static**: the emotion category embedding will not change during the generation process which may sacrifice grammatical correctness of sentences as argued in (Ghosh et al. 2017). Inspired by the psychological findings that emotional responses are relatively short lived and involve changes (Gross 1998; Hochschild 1979), and the dynamic emotion situation in emotional responses (Alam, Danieli, and Riccardi 2017), we design an internal memory module to capture the **emotion dynamics** during decoding. We simulate the process of expressing emotions as follows: there is an internal emotion state for each category before the decoding process starts; at each step the emotion state decays by a certain amount; once the decoding process is completed, the emotion state should decay to zero indicating the emotion is completely expressed.

The detailed process of the internal memory module is illustrated in Figure 2. At each step t, ECM computes a read

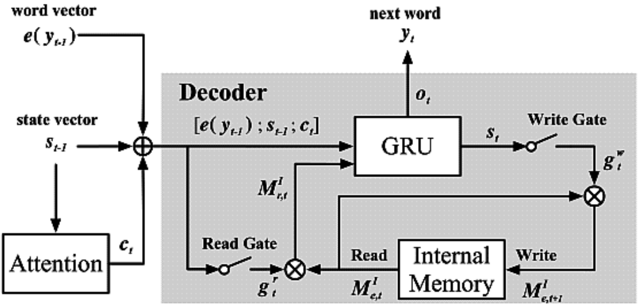

Figure 2: Data flow of the decoder with an internal memory. The internal memory $M^I_{e,t}$ is read with the read gate g^r_t by an amount $M^I_{r,t}$ to update the decoder's state, and the memory is updated to $M^I_{e,t+1}$ with the write gate g^w_t.

gate g^r_t with the input of 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. A write gate g^w_t is computed on the decoder's state vector s_t. The read gate and write gate are defined as follows:

$$g^r_t = \mathrm{sigmoid}(\mathbf{W^r_g}[e(y_{t-1}); s_{t-1}; c_t]), \quad (6)$$
$$g^w_t = \mathrm{sigmoid}(\mathbf{W^w_g} s_t). \quad (7)$$

The read and write gates are then used to read from and write into the internal memory, respectively. Hence, the emotion state is erased by a certain amount (by $\boldsymbol{g}_t^w$) at each step. At the last step, the internal emotion state will decay to zero. This process is formally described as below:

$$\boldsymbol{M}_{r,t}^I = \boldsymbol{g}_t^r \otimes \boldsymbol{M}_{e,t}^I, \tag{8}$$

$$\boldsymbol{M}_{e,t+1}^I = \boldsymbol{g}_t^w \otimes \boldsymbol{M}_{e,t}^I, \tag{9}$$

where $\otimes$ is element-wise multiplication, r/w denotes read/write respectively, and I means Internal. GRU updates its state $\boldsymbol{s}_t$ conditioned on the previous target word $\boldsymbol{e}(y_{t-1})$, the previous state of the decoder $\boldsymbol{s}_{t-1}$, the context vector $\boldsymbol{c}_t$, and the emotion state update $\boldsymbol{M}_{r,t}^I$, as follows:

$$\boldsymbol{s}_t = \mathbf{GRU}(\boldsymbol{s}_{t-1}, [\boldsymbol{c}_t; \boldsymbol{e}(y_{t-1}); \boldsymbol{M}_{r,t}^I]). \tag{10}$$

Based on the state, the word generation distribution $\boldsymbol{o}_t$ can be obtained with Eq. 4, and the next word y_t can be sampled. After generating the next word, $\boldsymbol{M}_{e,t+1}^I$ is written back to the internal memory. Note that if Eq. 9 is executed many times, it is equivalent to continuously multiplying the matrix, resulting in a decay effect since $0 \leq \text{sigmoid}(\cdot) \leq 1$. This is similar to a DELETE operation in memory networks (Miller et al. 2016).

External Memory

In the internal memory module, the correlation between the change of the internal emotion state and selection of a word is **implicit and not directly observable**. As the emotion expressions are quite distinct with emotion words (Xu et al. 2008) contained in a sentence, such as *lovely* and *awesome*, which carry strong emotions compared to generic (non-emotion) words, such as *person* and *day*, we propose an external memory module 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 an emotion vocabulary or a generic vocabulary.

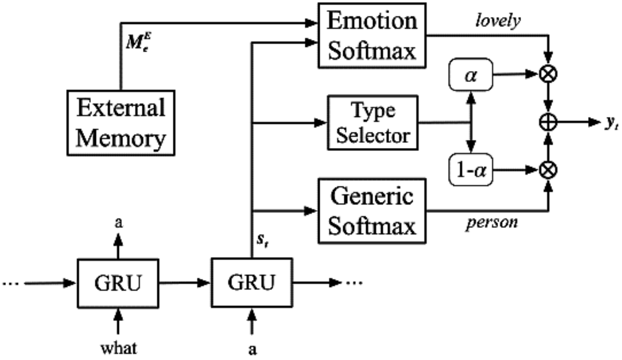

Figure 3: Data flow of the decoder with an external memory. The final decoding probability is weighted between the emotion softmax and the generic softmax, where the weight is computed by the type selector.

The decoder with an external memory is illustrated in Figure 3. Given the current state of the decoder $\boldsymbol{s}_t$, 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 type selector α_t controls the weight of generating an emotion or a generic word. Finally, the next word y_t is sampled from the next word probability, the concatenation of the two weighted probabilities. The process can be formulated as follows:

$$
\begin{aligned}
\alpha_t &= \mathrm{sigmoid}(\mathbf{v_u}^\top \boldsymbol{s}_t), &(11)\\
P_g(y_t = w_g) &= \mathrm{softmax}(\mathbf{W^o_g} \boldsymbol{s}_t), &(12)\\
P_e(y_t = w_e) &= \mathrm{softmax}(\mathbf{W^o_e} \boldsymbol{s}_t), &(13)\\
y_t \sim \boldsymbol{o}_t = P(y_t) &= \begin{bmatrix} (1-\alpha_t) P_g(y_t = w_g) \\ \alpha_t P_e(y_t = w_e) \end{bmatrix}, &(14)
\end{aligned}
$$

where $\alpha_t \in [0, 1]$ 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.

Loss Function

The loss function is the cross entropy error between the predicted token distribution $\boldsymbol{o}_t$ and the gold distribution $\boldsymbol{p}_t$ 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:

$$L(\theta) = -\sum_{t=1}^{m} \boldsymbol{p}_t \log(\boldsymbol{o}_t) - \sum_{t=1}^{m} q_t \log(\alpha_t) + \parallel \boldsymbol{M}_{e,m}^{I} \parallel, \tag{15}$$

where $\boldsymbol{M}_{e,m}^{I}$ is the internal emotion state at the last step m, α_t 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.

Data Preparation

Since there is no off-the-shelf data to train ECM, we firstly trained an emotion classifier using the NLPCC emotion classification dataset and then used the classifier to annotate the STC conversation dataset (Shang, Lu, and Li 2015) to construct our own experiment dataset. There are two steps in the data preparation process:

1. Building an Emotion Classifier. We trained several classifiers on the NLPCC dataset and then chose the best classifier for automatic annotation. This dataset was used in challenging tasks of emotion classification in NLPCC2013[2] and NLPCC2014[3], consisting of 23,105 sentences collected from Weibo. It was manually annotated with 8 emotion categories: *Angry*, *Disgust*, *Fear*, *Happy*, *Like*, *Sad*, *Surprise*, and *Other*. After removing the infrequent classes (*Fear* (1.5%) and *Surprise* (4.4%)), we have six emotion categories, i.e., *Angry*, *Disgust*, *Happy*, *Like*, *Sad* and *Other*.

We then partitioned the NLPCC dataset into training, validation, and test sets with the ratio of 8:1:1. Several emotion classifiers were trained on the filtered dataset, including a lexicon-based classifier (Liu 2012) (we used the emotion lexicon in (Xu et al. 2008)), RNN (Mikolov et al. 2010), LSTM (Hochreiter and Schmidhuber 1997), and Bidirectional LSTM (Bi-LSTM) (Graves, Fernández, and Schmidhuber 2005). Results in Table 2 show that all neural classifiers outperform the lexicon-based classifier, and the Bi-LSTM classifier obtains the best accuracy of 0.623.

Method	Accuracy
Lexicon-based	0.432
RNN	0.564
LSTM	0.594
Bi-LSTM	0.623

Table 2: Classification accuracy on the NLPCC dataset.

2. Annotating STC with Emotion. We applied the best classifier, Bi-LSTM, to annotate the STC Dataset with the six emotion categories. After annotation, we obtained an

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

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

emotion-labeled dataset, which we call the *Emotional STC (ESTC) Dataset*. The statistics of the *ESTC Dataset* are shown in Table 3. Although the emotion labels for *ESTC Dataset* are noisy due to automatic annotation, this dataset is good enough to train the models in practice. As future work, we will study how the classification errors influence response generation.

Training	Posts	217,905	
	Responses	Angry	234,635
		Disgust	689,295
		Happy	306,364
		Like	1,226,954
		Sad	537,028
		Other	1,365,371
Validation	Posts	1,000	
Test	Posts	1,000	

Table 3: Statistics of the *ESTC Dataset*.

Experiments

Implementation Details

We used Tensorflow[4] to implement the proposed model[5]. 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). To generate diverse responses, we adopted beam search in the decoding process of which the beam size is set to 20, and then reranked responses by the generation probability after removing those containing *UNK*s, unknown words.

We used 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 trained a seq2seq model on the *STC* dataset with pre-trained word embeddings. And we then trained our model on the *ESTC* Dataset with parameters initialized by the parameters of the pre-trained seq2seq model. We ran 20 epoches, and the training stage of each model took about a week on a Titan X GPU machine.

Baselines

As aforementioned, this paper is the first work to address the emotion factor in large-scale conversation generation. We did not find closely-related baselines in the literature. Affect-LM (Ghosh et al. 2017) cannot be our baseline because it is unable to generate responses of different emotions for the same post. Instead, it simply copies and uses the emotion of the input post. Moreover, it depends heavily on linguistic resources and needs manual parameter adjustments.

---

[4]https://github.com/tensorflow/tensorflow

[5]https://github.com/tuxchow/ecm

Nevertheless, we chose two suitable baselines: a general seq2seq model (Sutskever, Vinyals, and Le 2014), and an emotion category embedding model (Emb) created by us where the emotion category is embedded into a vector, and the vector serves as an input to every decoding position, similar to the idea of user embedding in (Li et al. 2016b). As emotion category is a high-level abstraction of emotion expressions, this is a proper baseline for our model.

Automatic Evaluation

Metrics: As argued in (Liu et al. 2016), BLEU is not suitable for measuring conversation generation due to its low correlation with human judgment. We adopted 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 adopted emotion accuracy as the agreement between the expected emotion category (as input to the model) and the predicted emotion category of a generated response by the emotion classifier.

Method	Perplexity	Accuracy
Seq2Seq	68.0	0.179
Emb	62.5	0.724
ECM	65.9	**0.773**
w/o Emb	66.1	0.753
w/o IMem	66.7	0.749
w/o EMem	**61.8**	0.731

Table 4: Objective evaluation with perplexity and accuracy.

Results: The results are shown in Table 4. As can be seen, ECM obtains the best performance in emotion accuracy, and the performance in perplexity is better than Seq2Seq but worse than Emb. This may be because the loss function of ECM is supervised not only on perplexity, but also on the selection of generic or emotion words (see Eq.15). In practice, emotion accuracy is more important than perplexity considering that the generated sentences are already fluent and grammatical with the perplexity of 68.0.

In order to investigate the influence of different modules, we conducted ablation tests where one of the three modules was removed from ECM each time. As we can see, ECM without the external memory achieves the best performance in perplexity. Our model can generate responses without sacrificing grammaticality by introducing the internal memory, where the module can balance the weights between grammar and emotion dynamically. After removing the external memory, the emotion accuracy decreases the most, indicating the external memory leads to a higher emotion accuracy since it explicitly chooses the emotion words. Note that the emotion accuracy of Seq2Seq is extremely low because it generates the same response for different emotion categories.

Manual Evaluation

In order to better understand the quality of the generated responses from the content and emotion perspectives, we performed manual evaluation. Given a post and an emotion category, responses generated from all the models were randomized and presented to three human annotators.

Metrics: Annotators were asked to score a response in terms of *Content* (rating scale is 0,1,2) and *Emotion* (rating scale is 0,1), and also to state a preference between any two systems. *Content* is defined as whether the response is appropriate and natural to a post and could plausibly have been produced by a human, which is a widely accepted metric adopted by researchers and conversation challenging tasks, as proposed in (Shang, Lu, and Li 2015). *Emotion* is defined as whether the emotion expression of a response agrees with the given emotion category.

Annotation Statistics: We randomly sampled 200 posts from the test set. For each model we generated 1,200 responses in total: for Seq2Seq, we generated the top 6 responses for each post, and for Emb and ECM, we generated the top responses corresponding to the 6 emotion categories.

We calculated the Fleiss' kappa (Fleiss 1971) to measure inter-rater consistency. Fleiss' kappa for *Content* and *Emotion* is 0.441 and 0.757, indicating "Moderate agreement" and "Substantial agreement" respectively.

Method (%)	2-1	1-1	0-1	2-0	1-0	0-0
Seq2Seq	9.0	5.1	1.1	37.6	28.0	19.2
Emb	22.8	9.3	4.3	27.1	19.1	17.4
ECM	**27.2**	**10.8**	4.4	24.2	15.5	17.9

Table 5: The percentage of responses in manual evaluation with the score of *Content-Emotion*. For instance, 2-1 means content score is 2 and emotion score is 1.

Results: The results are shown in Table 6. ECM with all options outperforms the other methods in both metrics significantly (2-tailed t-test, $p < 0.05$ for *Content*, and $p < 0.005$ for *Emotion*). After incorporating the internal memory and the external memory modules, the performance of ECM in *Emotion* is improved comparing to Emb, indicating our model can generate more explicit expressions of emotion. Besides, the performance in *Content* is improved from 1.256 of Emb to 1.299 of ECM, which shows the ability of ECM to control the weight of emotion and generate responses appropriate in content.

For all emotion categories, the performance of ECM in *Emotion* outperforms the other methods. However, the performances of ECM in *Content* is worse than baselines in *Disgust* and *Angry* categories, due to the fact that there are not sufficient training data for the two categories. For instance, the *Angry* category has 234,635 responses in our *ESTC Dataset*, much less than the other categories.

To evaluate whether ECM can generate responses that are appropriate not only in content but also in emotion, we present results in Table 5 by considering content and emotion scores simultaneously[6]. As we can see, 27.2% of the responses generated by ECM have a *Content* score of 2 and an *Emotion* score of 1, while only 22.8% for Emb and 9.0%

[6]Note that *Content* and *Emotion* are two independent metrics.

Post	Model		Response(Original)	Response(Translated)
承诺，就是一个骗子说给一个傻子听的。 A promise is what a cheater says to a fool.	Seq2Seq		承诺，就是一个骗子说给一个傻子听的。	A promise is what a cheater says to a fool.
	ECM	Like	承诺是一个专一的傻子	Promise is a single-minded fool
		Happy	我就是一个开心的傻子	I'm just a happy fool
		Sad	我就是一个悲伤的傻子	I am a sad fool
		Disgust	一个骗子说给一个傻子听	A cheater says to a fool
		Angry	不管你信不信，反正我信了！	Whatever you believe or not, I trust in it!
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.
		Happy	笑得我眼泪都出来了。	I can't help laugh myself to tears.
		Sad	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?
		Angry	什么意思？没看懂。	What do you mean? I don't understand.

Figure 4: Sample responses generated by Seq2Seq and ECM (original Chinese and English translation, the colored words are the emotion words corresponding to the given emotion category). The corresponding posts did not appear in the training set.

Method	Overall		Like		Sad		Disgust		Angry		Happy	
	Cont.	Emot.	Cont.	Emot.	Cont.	Emot.	Cont.	Emot.	Cont.	Emot.	Cont.	Emot.
Seq2Seq	1.255	0.152	1.308	0.337	1.270	0.077	**1.285**	0.038	**1.223**	0.052	1.223	0.257
Emb	1.256	0.363	1.348	0.663	1.337	0.228	1.272	0.157	1.035	0.162	1.418	0.607
ECM	**1.299**	**0.424**	**1.460**	**0.697**	**1.352**	**0.313**	1.233	**0.193**	0.98	**0.217**	**1.428**	**0.700**

Table 6: Manual evaluation of the generated responses in terms of *Content* (Cont.) and *Emotion* (Emot.) .

for Seq2Seq. These indicate that ECM is better in generating high-quality responses in both content and emotion.

Pref. (%)	Seq2Seq	Emb	ECM
Seq2Seq	-	38.8	38.6
Emb	60.2	-	43.1
ECM	**61.4**	**56.9**	-

Table 7: Pairwise preference of the three systems.

Preference Test: In addition, emotion models (Emb and ECM) are much more preferred than Seq2Seq, and ECM is also significantly (2-tailed t-test, $p < 0.001$) preferred by annotators against other methods as shown in Table 7. The diverse emotional responses are more attractive to users than the generic responses generated by the Seq2Seq model. And with the explicitly expressions of emotions as well as the appropriateness in content, ECM is much more preferred.

Analysis of Emotion Interaction and Case Study

Figure 5 visualizes the emotion interaction patterns of the posts and responses in the *ESTC Dataset*. An emotion interaction pattern (EIP) is defined as $< e_p, e_r >$, the pair of emotion categories of the post and its response. The value of an EIP is the conditional probability $P(e_r|e_p) = P(e_r, e_p)/P(e_p)$. An EIP marked with a darker color occurs more frequently than a lighter color. From the figure, we can make a few observations. **First**, frequent EIPs show that there are some major responding emotions given a post emo-

Other Like Sad Disgust Angry Happy

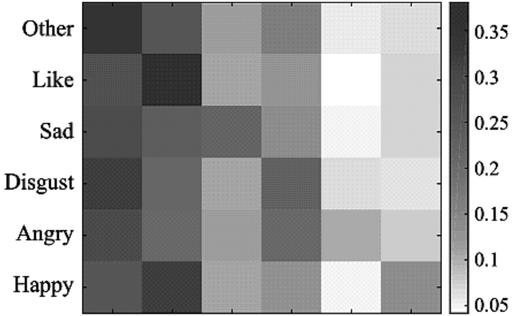

Figure 5: Visualization of emotion interaction.

tion category. For instance, when a post expresses *Happy*, the responding emotion is typically *Like* or *Happy*. **Second**, the diagonal patterns indicate emotional empathy, a common type of emotion interaction. **Third**, there are also other EIPs for a post, indicating that emotion interactions in conversation are quite diverse, as mentioned earlier. Note that class *Other* has much more data than other classes (see Table 3), indicating that EIPs are biased toward this class (the first column of Figure 5), due to the data bias and the emotion classification errors.

We present some examples in Figure 4. As can be seen, for a given post, there are multiple emotion categories that are suitable for its response in conversation. Seq2Seq generates a response with a random emotion. ECM can generate emotional responses conditioned on every emotion category. All these responses are appropriate to the post, indicating the existence of multiple EIPs and the reason why an emotion category should be specified as an input to our system.

We can see that ECM can generate appropriate responses if the pre-specified emotion category and the emotion of the post belong to one of the frequent EIPs. Colored words show that ECM can explicitly express emotion by applying the external memory which can choose a generic (non-emotion) or emotion word during decoding. For low-frequency EIPs such as $< Happy, Disgust >$ and $< Happy, Angry >$ as shown in the last two lines of Figure 4, responses are not appropriate to the emotion category due to the lack of training data and/or the errors caused by the emotion classifier.

Conclusion and Future Work

In this paper, we proposed the Emotional Chatting Machine (ECM) to model the emotion influence in large-scale conversation generation. Three mechanisms were proposed to model the emotion factor, including emotion category embedding, internal emotion memory, and external memory. Objective and manual evaluation show that ECM can generate responses appropriate not only in content but also in emotion.

In our future work, we will explore emotion interactions with ECM: instead of specifying an emotion class, the model should decide the most appropriate emotion category for the response. However, this may be challenging since such a task depends on the topics, contexts, or the mood of the user.

Acknowledgments

This work was partly supported by the National Science Foundation of China under grant No.61272227/61332007, and a joint project with Sogou. We would like to thank our collaborators, Jingfang Xu and Haizhou Zhao.

References

[Alam, Danieli, and Riccardi 2017] Alam, F.; Danieli, M.; and Riccardi, G. 2017. Annotating and modeling empathy in spoken conversations. *CoRR* abs/1705.04839.

[Bahdanau, Cho, and Bengio 2014] Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473.

[Cagan, Frank, and Tsarfaty 2017] Cagan, T.; Frank, S. L.; and Tsarfaty, R. 2017. Data-driven broad-coverage grammars for opinionated natural language generation (onlg). In *ACL*, volume 1, 1331–1341.

[Cho et al. 2014] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078.

[Chung et al. 2014] Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555.

[Fleiss 1971] Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76(5):378.

[Ghosh et al. 2017] Ghosh, S.; Chollet, M.; Laksana, E.; Morency, L.; and Scherer, S. 2017. Affect-lm: A neural language model for customizable affective text generation. In *ACL*, 634–642.

[Graves, Fernández, and Schmidhuber 2005] Graves, A.; Fernández, S.; and Schmidhuber, J. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *ICANN*, 799–804. Springer.

[Gross 1998] Gross, J. J. 1998. The emerging field of emotion regulation: An integrative review. *Review of general psychology* 2(3):271.

[Gu et al. 2016] Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, 1631–1640.

[Herzig et al. 2017] Herzig, J.; Shmueli-Scheuer, M.; Sandbank, T.; and Konopnicki, D. 2017. Neural response generation for customer service based on personality traits. In *Proceedings of the 10th International Conference on Natural Language Generation*, 252–256.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

[Hochschild 1979] Hochschild, A. R. 1979. Emotion work, feeling rules, and social structure. *American journal of so-*

ciology 551–575.

[Hu et al. 2017] Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward controlled generation of text. In *ICML*, 1587–1596.

[Li et al. 2016a] Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016a. A diversity-promoting objective function for neural conversation models. In *NAACL*, 110–119.

[Li et al. 2016b] Li, J.; Galley, M.; Brockett, C.; Spithourakis, G.; Gao, J.; and Dolan, W. B. 2016b. A persona-based neural conversation model. In *ACL*, 994–1003.

[Liu et al. 2016] Liu, C.; Lowe, R.; Serban, I.; Noseworthy, M.; Charlin, L.; and Pineau, J. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*, 2122–2132.

[Liu 2012] Liu, B. 2012. *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers.

[Martinovski and Traum 2003] Martinovski, B., and Traum, D. 2003. Breakdown in human-machine interaction: the error is the clue. In *Proceedings of the ISCA tutorial and research workshop*, 11–16.

[Mayer and Salovey 1997] Mayer, J. D., and Salovey, P. 1997. What is emotional intelligence? *Emotional Development and Emotional Intelligence* 3–31.

[Mikolov et al. 2010] Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, 3.

[Miller et al. 2016] Miller, A. H.; Fisch, A.; Dodge, J.; J. 2008. Affective lexicon ontology. *Journal of information*
Karimi, A.; Bordes, A.; and Weston, J. 2016. Key- 27(2):180–185.
value memory networks for directly reading documents. In
EMNLP, 1400–1409.
[Mou et al. 2016] Mou, L.; Song, Y.; Yan, R.; Li, G.; Zhang,
L.; and Jin, Z. 2016. Sequence to backward and for-
ward sequences: A content-introducing approach to gener-
ative short-text conversation. In *COLING*, 3349–3358.
[Partala and Surakka 2004] Partala, T., and Surakka,
V. 2004. The effects of affective interventions in
human–computer interaction. *Interacting with computers*
16(2):295–309.
[Picard and Picard 1997] Picard, R. W., and Picard, R. 1997.
Affective computing, volume 252. MIT press Cambridge.
[Polzin and Waibel 2000] Polzin, T. S., and Waibel, A. 2000.

Emotion-sensitive human-computer interfaces. In *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*, 201–206.

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

[Prendinger, Mori, and Ishizuka 2005] Prendinger, H.; Mori, J.; and Ishizuka, M. 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.

[Ritter, Cherry, and Dolan 2011] Ritter, A.; Cherry, C.; and Dolan, W. B. 2011. Data-driven response generation in social media. In *EMNLP*, 583–593.

[Serban et al. 2015] Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2015. Hierarchical neural network generative models for movie dialogues. *CoRR* abs/1507.04808.

[Serban et al. 2016] Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 3776–3784.

[Shang, Lu, and Li 2015] Shang, L.; Lu, Z.; and Li, H. 2015. Neural responding machine for short-text conversation. In *ACL*, 1577–1586.

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

[Sutskever, Vinyals, and Le 2014] Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing*

systems, 3104–3112.
[Vinyals and Le 2015] Vinyals, O., and Le, Q. 2015. A neural conversational model. *CoRR* abs/1506.05869.
[Xing et al. 2017] Xing, C.; Wu, W.; Wu, Y.; Liu, J.; Huang, Y.; Zhou, M.; and Ma, W. 2017. Topic aware neural response generation. In *AAAI*, 3351–3357.
[Xu et al. 2008] Xu, L.; Lin, H.; Pan, Y.; Ren, H.; and Chen,