THE HONG KONG POLYTECHNIC UNIVERSITY

INTERNET PROFESSIONAL ASSOCIATION

PAN PEARL RIVER DELTA +
COLLEGE STUDENT COMPUTER WORK COMPETITION

Emotional Chatting System: How to Endow Chatbots with Human Emotions Based on Dialog Analysis

Author
Jiashuo WANG

Supervisor
Dr. Maggie Wenjie LI

August 27, 2020

Abstract

The development of conversational agents is getting increasing attention from both academia and industries in recent years. At present, chatbots are widely used as an agent to communicate and interact with human in some service activities, such as customer services, shopping assistants, and even social chatting partners. One long-term goal of researches in chatbots is to humanize machines to engage and satisfy users. Some studies have proved that emotion is an important component to humanize machines. In recent years, many approaches based on neural networks equip dialog systems with human affect, as *Emotional Chatting Machine* [1] and *Affective Neural Response Generation* [2]. However, most of the existing methods neglect the diversity of replies and generate monotonous responses [3]. Additionally, responses generated by these methods correlate weakly with their queries at the content level. Moreover, few works consider the affective information contained in the input sentences, which provide weakens emotional consistency between queries and replies. To address these issues, this project proposes **Emotional Chatting System** (**ECS**) which can generate diversiform responses with relevant content and appropriate emotions, thanks to its capability to capture semantic, syntactic and emotional relations between queries and replies. **ECS**, based on a GRU sequence-to-sequence (Seq2Seq) neural network, addresses the problems mentioned above by employing three mechanisms that, (1) enhance the content relevance between queries and response utilizing attention one topic words, (2) reduce occurrences of trivial words, (3) pay attention to the input emotion and improve explicit emotion expressions with an external emotion vocabulary, respectively. The evolution of ECS at content level and emotion level has proved the validity and effectiveness of this model in semantic, syntactic, and emotional expression.

Table of Contents

List of Tables

List of Figures

1. Introduction

Chatting systems, also known as interactive conversational agents, dialogue systems and sometimes chatbots, are employed in a broad set of fields in human livings ranging from education to entertainment and business [4, 5], such as Amazon Alexa[1] and Microsoft Cortana[2]. These applications are capable of answering a wide range of questions and performing particular subtasks, which facilitates people's lives. Moreover, studies in this field still continue to humanize chatbots, in order to have a better engagement when communicating with users [6].

Early conversational agents are often designed based on *rule-based* approaches [7], which have low efficiency and poor performance especially for large scale conversation generation. With conversational data accumulating, the most common and widely studied and adopted approaches today in chatting systems lie on *data-driven* ones such as *retrieval-based* [8-10] methods and *generation-based* [11-13] methods. Compared with *retrieval-based* methods, *generation-based* chatting systems are able to generate new utterances, which have not appeared in the dataset [14]. This feature allows a higher diversity of generated responses. Therefore, the method proposed in this paper focuses on the second approach.

Emotional intelligence, one of the core parts of human intelligence, is the competence to identify, evaluate, and govern the emotions [15]. Sentiment in the communication not only reveals speakers' attitudes, but also helps communicators to clarify their intention. It has been proved that human-computer interaction with the involvement of sentiment can be more natural, enjoyable, and productive [16, 17]. There are various applications on review-related text across different domains, such as automatic analysis of citizens' attitude to a pending policy, where the involvement of emotional intelligence can make the chatting system generate more humanlike expression, improving participants' satisfaction [18]. To build an open domain chatbot which is able to communicate at a human level, it is necessary to endow the machine with the capability of perceiving, evaluating, and governing emotions.

Existing work utilizes recurrent neural networks (RNNs) architecture and sentiment analysis to build sentiment dialog systems, which proves that emotion factor in dialogs builds up dialogs with

[1] https://developer.amazon.com/alexa

[2] https://www.microsoft.com/en-us/cortana

higher qualities [1, 2, 19]. However, a common problem is that the dialogue systems generate trivial responses [3], where large homogeneous patterns may be observed. Additionally, the generated responses have a low correlation with their input utterances. Moreover, input emotional information is always ignored during response generation in most of current emotional chatbot models. This leads to the inconsistent emotional expression between the input utterance and the output utterances.

Acknowledging all the above, this paper tackles the issue of response generation with various emotions in the open-domain conversation system and designs an emotional reply generation system, **Emotional Chatting System** (**ECS**). This model aims to generate diversiform responses with coherent content enriched by correct and consistent emotion expression.

The remainder of this paper is organized as follows: **Section 2** will explore previous works in this domain of chatting systems and emotional analysis as well as existing problems. **Section 3** will be given to describe the dataset in the experiments and some applicable preprocessing. Form **Section 4** to **Section 5**, the technical approaches to building the model are addressed from scratch and the model design would also be illustrated. **Section 6**, combining the previous parts, provides the experiment details and the evaluation results. Finally, **Section 7** should conclude the paper and avenues for future researches will be explored as well.

1.1. Problem Definition

The goal of this project is to design an affective open-domain chatting system. The problem can be defined as follows:

> Given an input utterance and an expected emotion of reply, this dialogue system should generate a response appropriately at not only contents level but also emotion level with the help of the emotion tag. The emotion tags are from the following emotion categories: *Like*, *Sad*, *Disgusted*, *Angry*, *Happy*, and *the Other*.

1.2. Objective

In term of the background of current researches and problem definition **above**, the objective of this project is to:

1. Design and evaluate an open domain emotional dialog system. This response generation system should be able to generate grammatical and sentimental replies.
2. Reduce the repetition of terms and patterns in the responses in the current emotional dialog system.
3. Enhance the effect of emotion involved in the input post on the output generation.

This model will be evaluated at both the content level and emotion level. It will be examined whether this model is more diversified and more sentimental. Detailed analysis of these experimental results shall be completed to explore the features of this model, which will help to point out the future directions of this system.

1.3. Contribution

In summary, the primary contributions of this work include:

1. Reduces repetitive patterns in replies and increases the diversity of sentimental responses by introducing topics as prior knowledge.
2. Proves that the involvement of topics, emotions, and grammar words can improve the quality of generated responses.
3. Proposes a Seq2Seq framework (ECS) considering the influence of emotional information of carried by the input utterances during response generation.

2. Relate Works

Related works about this project can be primarily discussed from in aspects. Firstly, **Section 2.1** is about neural network approaches to generating responses. Secondly, the reason why sentimental factors should be involved in conversational systems is discussed and illustrated in **Section 2.2**. Finally, existing work introducing affective factor into chatting systems and their problems are argued in **Section 2.3**.

2.1. Open-domain Response Generation

Designing open-domain response generation systems, which in essence falls under the category of text generation problem, are widely studied and researched. Inchoate response generation systems are implemented with *rule-based* approaches [7], which lack the competence to deal with large scale data and produce high-quality responses. Driven by increasing large volume of conversation data, researches related to *corpus-based* methods predominate [20]. Currently, it is a trend to adopt neural network generation models, especially sequence-to-sequence (Seq2Seq) models, when dealing with text generation problems [21, 22]. LSTM Seq2Seq models have already been successfully used in the application of translation and response generation and achieve success [13, 23]. However, it has been proved empirically that gated recurrent unite (GRU) converges faster than LSTM, and each has its advantages in different language models [24]. Therefore, GRU Seq2Seq models sometimes have even more superior performance than LSTM. This adopts GRU Seq2Seq model because the fast convergence ability.

2.2. Sentiment Analysis in Conversational Systems

Sentiment processing is always a significant mission for the research of natural language processing. Early works, such as emotion detection and classification, remained on apparent emotional information. With emotion playing an increasingly important role in applications, inner affective information is investigated and explored, and related research directions, such as emotion cause analysis and affective expression, are developing. Recently, endowing human-machine interactions with sentiment is one of the most active and challenging research topics [18, 25]. The involvement of affective analysis in dialog significantly influences users' satisfaction. Besides, human-interactive with sentiment consideration can continue the conversation, which is productive in the practice [17]. For example, anger emotion often leads to a digression, and

conciliation strategies, when being taken appropriately in dialogs, can help to continue the conversation and boost the user's experience [19]. Machine chatting systems can take advantage of this kind of affective processing. While there are many approaches to generate emotional responses [1, 2, 26], this task can be primarily comprehended as two sub-components. The first step is to study human's affect and introduce the sentimental information. The second sub-task is to design approaches to react reasonably.

2.3. Existing Works and Problems

Furthermore, there are also numerous studies on equipping chatting systems with sentiment, all of which have enhanced the sentimental expression of the dialogue systems notably. *Affective-LM* [26] incorporates affection in the neural text generation. It generates sentences conditioned on affective category information learned from tokens in the context, yet it concentrates on language models instead of dialogue systems. *Emotional Chatting Machine* [1] addresses emotion factors in the large-scale open-domain conversation systems, introducing an internal and external memory, both of which function at the decoder side. The internal memory adjusts emotion state, while the external memory decides to select a generic or emotional word. These two mechanisms proposed in this paper have a great influence on later research. However, there are many repetitive terms and patterns in the responses. *Affective Neural Response Generation* [2] adopts affective word embeddings and affectively diverse beam search algorithm. This work has proved that the sentimental factors improve the quality of responses. However, one disadvantage in this model is that responses cannot be generated with assigned emotions. One thing worth mentioning is that all of above models neglect the emotion categories of the input post and make transformation mainly at the decoder side. *Mojitalk* [27] proposes a model that utilizes a conditional variational autoencoder (CVAE) framework as the encoder and embeds the input emotion category into the encoding state, although this work focuses on the emoji categories from tweet, instead of human emotion categories.

One common problem that existing works encounter is that there are always repetitive and trivial responses, such as "I don't know.", "Ha-ha!", and "Don't say that.", due to the high frequency of such patterns in the conversation corpus [3]. Although these universal responses make sense in semantics, conversations are inefficient and can be terminated immediately. Such responses make users bored and unsatisfied. Besides, as mentioned, there are few works pay attention to the input post emotion categories, which impact considerably the expression of output responses. For instance, replies with the *happy* label to an *angry* post and a *like* post should be

different in term of the affective intensity and wording. Disregard of input emotion will constrain the emotion expression of output replies.

3. Dataset

3.1. Dataset Characteristics

I leverage the conversation dataset from the NLPCC emotion classifier challenge [1, 28]. The contents are one-round (containing one query and one response) Chinese daily conversation collected from Weibo from 2013 through 2015. All data has already been labelled with emotion category tags (*Like*, *Sad*, *Disgusted*, *Angry*, *Happy*, and *the Other*), as defined in **Section 1.1**. The text in the dataset has been segmented into Chinses terms. These characteristics may come in handy in the process of training.

Another point worth mentioning is that some of the sentences contain swear words, indicating negative emotions, such as disgust and anger. For example, people say “Stupid jerk!” to convey anger. Therefore, I choose to keep these swear words as one of the features in some emotion categories.

3.2. Data Pre-processing

When processing the data, illogical and ungrammatical sentences, which can influence training, are filtered manually. Besides, some short or unclear queries, lacking contextual information, are deleted along with their responses. For instance, a single sentence “Yes, it is.” is hard to answer as a beginning query, because this sequence is an approval or agreement of its previous sentence and it is impossible to tell the topic of this conversation. Besides, some frequent sentences containing only modal particles, such as “Ha-ha~” and “I see”, are also deleted. Dialogue models trained with dataset containing such sentences will hinder the systems from learning, leading to trivial responses, because the training objective of maximum likelihood trends to produce high frequency responses. Moreover, word segmentation is manually checked and corrected accordingly.

3.3. Data Distribution

After the pre-processing, I randomly select around 1 million sentence pairs as the training dataset, and another 100,000 sentence pairs as the testing dataset from the whole processed NLPCC dataset. Random selection guarantees the same data distribution of training data and testing data. The training data distributes as below:

Query Emotion Category	**Response Emotion Category**						
	Like	*Sad*	*Disgust*	*Angry*	*Happy*	*Others*	*Total*
Like	29,365	57,656	25,957	28,333	16,767	45,501	203,579
Sad	15,475	13,408	31,349	16,186	12,387	17,630	106,435
Disgust	24,949	18,188	22,197	45,718	22,955	22,168	156,175
Angry	10,858	6,628	9,625	13,772	17,389	10,769	69,041
Happy	16,021	20,942	18,393	16,103	12,949	34,831	119,239
Others	61,100	36,418	37,062	43,611	27,677	41,305	247,173
Total	157,768	153,240	144,583	163,723	110,124	172,204	901,642

Table 1 Distribution of training data

4. Technical Approaches

Several necessary technical techniques and models are referred and applied in this project. These approaches help to compose, train, and evaluate the model proposed in this paper. The first one is the framework Sequence-to-Sequence (Seq2Seq) model shown in **Section 4.1**. The conditional variational autoencoder (CVAE) model is displayed in **Section 4.2**. Another is Beam Search illustrated in **Section 4.3**.

4.1. Sequence-to-Sequence Model (Seq2Seq)

For response generation, a Seq2Seq model with the attention mechanism, which is a growing interest in text generation [13, 23, 29], is employed as the basic framework. This structure is implemented with two GRUs as encoder and decoder. **Figure 1** displays the architecture of this configuration:

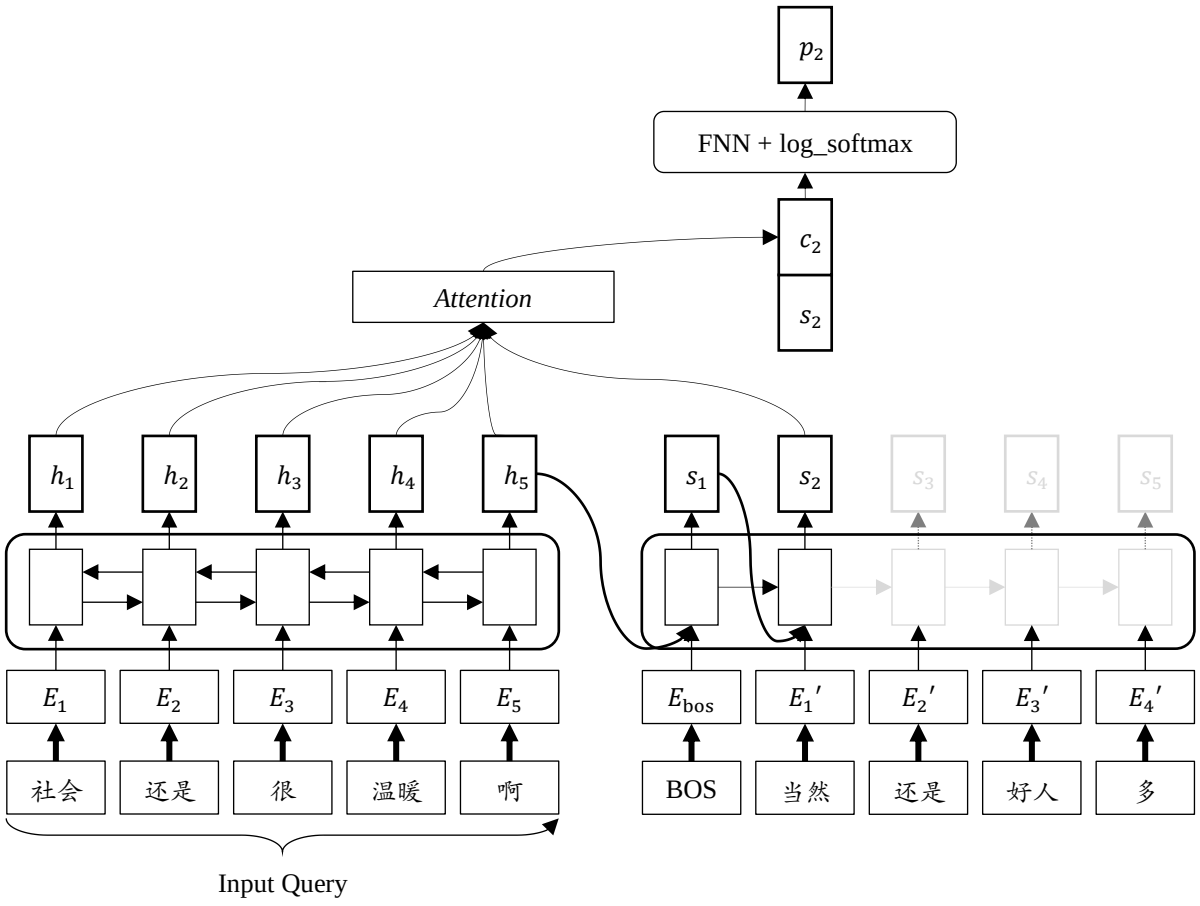

Figure 1 Architecture for Seq2Seq with attention mechanism[3]

For the encoder, a bi-direction GRU reads the source sentence $Q = (q_1, \dots, q_N)$, and then encodes it into a sequence of hidden states $H = (h_1, \dots h_N)$ where $h_n = GRU(x_n, h_{previous})$. Another GRU generates a consistent sequence $R = (\widehat{r_1}, \dots, \widehat{r_L})$ in terms of the hidden states *H*. The word probability distribution of $\widehat{\boldsymbol{r_l}}$ is calculated by:

$$\begin{aligned} \widehat{r_l} \sim o_l &= P(\widehat{r_l}|r_1, \dots, r_{l-1}, H), \\ &= log\_softmax(W_o \cdot s_l) \end{aligned} \quad (1)$$

where $\mathbf{s_l}$ is the updated state:

$$s_l = GRU([c_l; s_{l-1}]);\ s_0 = h_N \quad (2)$$

$[c_l; s_{l-1}]$ is the concatenation of these two vectors. $\boldsymbol{c_l}$ is the context vector, which dynamically attends on the input tokens. It is a weighted sum of *H* with coefficients $(\alpha_1, \dots, \alpha_N)$, which is defined as:

$$\alpha_k = \frac{score(s_l, h_k)}{\sum_{i=1}^{N} score(s_l, h_i)} \quad (3)$$

This neural network is trained to minimize the negative log likelihood loss with respect to parameters θ:

$$\theta = \underset{\theta}{argmin} - \sum_{l=1}^{L} P(r_l|r_1, \dots, r_{l-1}, H) \quad (4)$$

4.2. Conditional Variational Autoencoder (CVAE)

Usually, the encoder of a Seq2Seq model is RNN. However, recent research has found that this model tends to generate dull and generic replies, instead of specific and meaningful answers [30]. Many attempts have explained and solved this problem. One method is utilization of conditional variational autoencoders model (CVAE) [31], which introduces a latent variable $\boldsymbol{z}$ capturing the distribution over the valid responses. CVAE defines the conditional distribution:

$$p(x, z|c) = p(x|z, c)p(z|c) \quad (5)$$

[3] FNN: Feedforward Neural Network. BOS: a token indicating the beginning of a sentence.

and the goal is to use deep neural networks (usually MLP) to approximate $p(z|c)$ and $p(x|z,c)$, where $\boldsymbol{c}$ is the dialog context, and $\boldsymbol{x}$ is the response utterance.

4.3. Beam Search (BS)

After the Seq2Seq model predicts the token distribution given the query and previous tokens in the response, a method is required to help generate a complete output utterance. This project adopts Beam Search (BS) to accomplish this process. Compared with Greedy Search (BS with $B = 1$), BS can grab more information and select the result with better global performances [32]. Usually, beam search stores top-B most possible subsequences in the memory, where B is the beam size. At each step i, the top-B subsequences generated at step $(i - 1)$ are expended with all possible tokens. Then the top-B most likely branches are retained, and the rest are pruned. BS returns the sequence when the current token is a terminating signal, or the current step exceeds the required sequence length.

5. Model Design: ECS

This section demonstrates the design and vital techniques inside the proposed model. To generate more consistent and content-related replies with emotions, a new dialog system with sentiment called **Emotional Chatting System (ECS)** is proposed. The overall design of **ECS** can be represented as **below**:

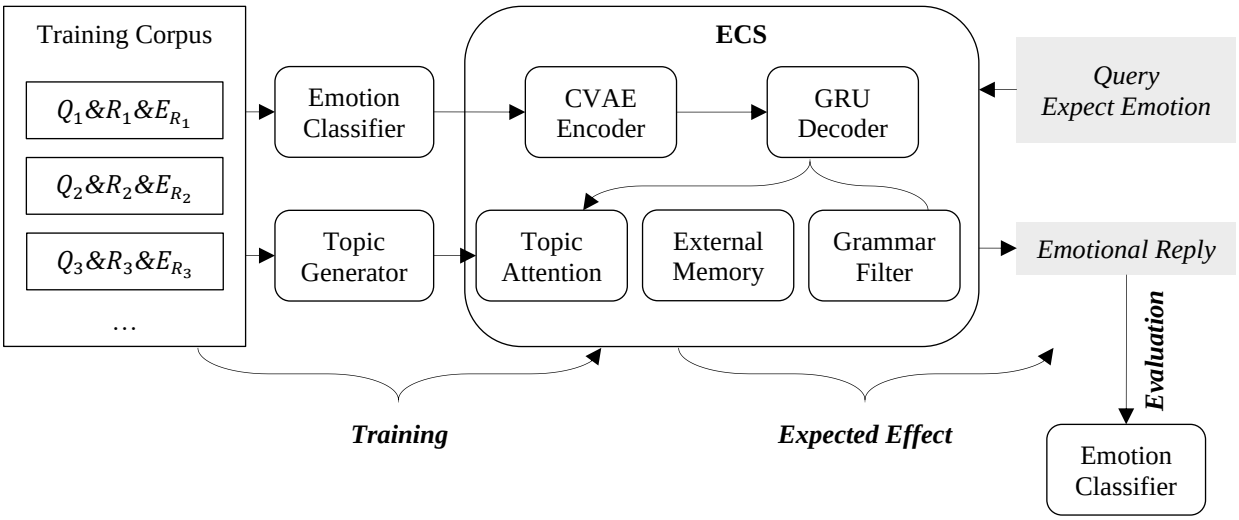

Figure 2 Overall Structure of Emotional Chatting System

ECS takes the attention-based Seq2Seq with CVAE encoder as the basic framework. At the encoder side, the emotion category of the input post, which is predicted by a trained emotion classifier, and the expected response emotion are incorporated into the encoder state. In order to express relevant content and coherent emotion, the decoder is equipped with three essential mechanisms -- a **Topic Attention** to capture relevant sematic from input sequences, an **External Memory** to increase the occurrence of affective tokens, and a **Grammar Filter** to avoid repetition of indifferent words and phrases. The emotional CVAE encoder and these three mechanisms will be illustrated in **Section 5.4**, **Section 5.6**, **Section 5.7** and **Section 5.7**, respectively.

Before this main body of ECS, there are some pre-analysis of the input data. **Section 5.1** defined notations and the task. Sentimental analysis at word level and sentence level are presented in **Section 5.2**. Approaches to obtain the topic words of input utterances are discussed in **Section 5.3**.

Analysis results of queries acquired from sentiment analysis and topic generation, and the original input conversational data are fed in the main body of the ECS.

5.1. Notations and Task Definitions

Basic notations and symbols in this model are defined in the **below** table:

Notation	Description
$\boldsymbol{V}, \boldsymbol{v}$	Size of the vocabulary
$\boldsymbol{N}, \boldsymbol{n}$	Length of the query sequence
$\boldsymbol{M}, \boldsymbol{m}$	Number of the topic word
$\boldsymbol{L}, \boldsymbol{l}$	Length of the response sequence
$\boldsymbol{H}, \boldsymbol{h}$	Hidden state(s) in encoder
$\boldsymbol{W}$	Weights
$\boldsymbol{S}, \boldsymbol{s}$	Hidden state(s) in decoder
$\boldsymbol{P}, \boldsymbol{p}$	Probability or probability distribution
Rectangle in the figure	Vector(s) / Array(s)
Oval / Rectangle (Rounded Corners) in the figure	Neural Networks

Table 2 Notations Description

Acknowledging all the above, an overall task can be defined as:

> Given an input utterance $Q =< q_0, \dots, q_n, \dots q_N >$ and an expected emotion $\boldsymbol{e}$ of reply, the goal is to generate a response $R =< r_1, \dots, r_l, \dots r_L >$. This response R is expected to belong to the emotion category $\boldsymbol{e}$.

5.2. Sentiment Analysis

This paper adopts six emotion categories, and they are *Like*, *Sad*, *Disgusted*, *Angry*, *Happy*, and *Other*. Emotions can be detected and analyzed from two aspects – tokens and sequences, which are evaluated through the affective score in **Section 5.2.1** and the emotion classifier in **Section 5.2.2**, respectively.

5.2.1. Affective score

To differentiate words in terms of their emotions, an affective score $\boldsymbol{s_{w,e}}$ is assigned to every word $\boldsymbol{w}$ with emotion $\boldsymbol{e}$. Each word has five scores corresponding to different emotion categories that it belongs to (except *the Other*). I calculate the *TD-IDF* value of word $\boldsymbol{w}$ in sentences labelled with emotion $\boldsymbol{e}$, and set this value as the score $\boldsymbol{s_{w,e}}$. This score $\boldsymbol{s_{w,e}}$ measures the contribution of the word $\boldsymbol{w}$ to the emotion category $\boldsymbol{e}$. According to these scores, an emotional vocabulary used in **External Memory** in **Section 5.6** is able to be created.

5.2.2. Emotion classifier

I fine-turn a Bert model [33] to predict emotions of utterances. The architecture of this emotion classifier is displayed in **Figure 3**:

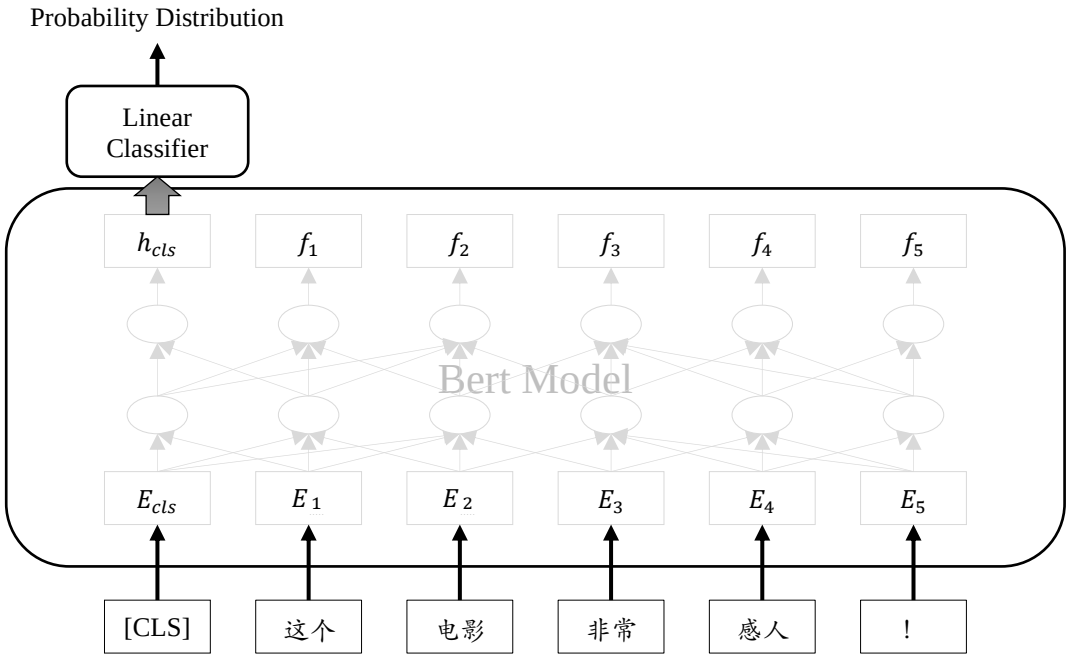

Figure 3 Architecture for emotion classifier

The input of the classifier is a sequence, while the output is the probability distribution over the six emotion categories. In the training process, I feed the model with the sentences and their expected emotion labels. The model has 768 hidden layers. I collect the last layer hidden states of the first token in the sequence, which should be a special token called "[CLS]" in Bert model, and put all these values into a linear classifier. The process to predict the emotion of a sequence $\boldsymbol{S}$ can be formulated as the following:

$$h_{cls}^{0}, \dots h_{cls}^{767} = Bert(S) \quad (6)$$

$$p_{Like}, p_{Sad}, p_{Disgusted}, p_{Angry}, p_{Happy}, p_{Other} = softmax\ (Linear\ Classifier(h_{cls}^{0}, \dots h_{cls}^{767})) \quad (7)$$

$$l_x = argmax(p_l) \quad (8)$$

where h_{cls}^{x} is the x^{th} hidden state of "[CLS]", and l_x is the predicted emotion category.

One advantage of Bert models is that when fining turning the model, it is simply needed to train and adjust parameters inside the linear classifier at the last layer of the whole model. This feature not only speeds up the training process, but also helps to obtain higher accuracy in the classification task.

5.3. Topic Generation

To obtain the topic words of a given query, a topic word generator is constructed based on a word embedding model -- **Skip-gram with Negative Sampling (SGNS)** and **gaussian mixture models** (**GMM**). Topic words are selected from the results of the combination of these two methods. These topic vocabularies play an important role in **Topic Attention** which is demonstrated in **Section 5.4**.

5.3.1. Word embedding model

To train the word embedding of frequent vocabulary, a **Skip-gram with Negative Sampling** (**SGNS**) model is applied. **Skip-gram** [34] is a popular method to train word embeddings. Given a single word $\boldsymbol{w}$, this model predicts its potential neighboring words. **Negative Sampling** is to randomly select a small number of negative words, which are never the neighbor words of the $\boldsymbol{w}$, to represent negative words. This technique helps to seep up the training process and improve the quality of word vectors. Before training, we delete some trivial terms which make little contribution to the semantic context, such as "你", "了", and "也". Then we combine the query and its response as one sequence to enhance the relationship between words in one-round conversation, because empirically, one paired dialog should discuss the same topics.

5.3.2. Gaussian mixture models (GMMs)

With word embeddings produced by the skip-gram model as the input features, Gaussian mixture models (GMMs) are used to cluster words according to their semantic meanings. Each

cluster contains vocabulary that can appear when talking about one topic. GMMs are probabilities models and utilize a soft clustering method. Each node has probabilities belongs to every cluster. Besides, Gaussian model considers both means and variances of features. Therefore, compared with another popular approach K-means, GMMs should perform more efficiently in this subtask.

5.3.3. Topic words selection

In order to obtain the most favorable topic words of a query, it is necessary to collect the most correlated words of one given word. For each term in the input post, it is decided to select at most 5 most similar words, according to the word embeddings, from the cluster it belongs to. The topic words of a sentence are randomly selected from the collection of all topic words of tokens in the sentence. Moreover, this random selection can enhance statement diversity.

5.4. Emotional CVAE Encoder

The encoder in this model undertakes important mission. Besides to approximate the response distribution by a latent variable $\boldsymbol{z}$ mentioned in **Section 4.2**, it captures the relationship between the query emotion category $\boldsymbol{e_q}$ and the response emotion category $\boldsymbol{e_r}$. These emotion categories are embedded and concatenated:

$$e = [embed(\boldsymbol{e_q});\ embed(\boldsymbol{e_r})] \tag{9}$$

then, the emotion vector and the latent variable are used to update the encoder state $\boldsymbol{h}$:

$$h = [z; e; h] \tag{10}$$

5.5. Topic Attention

This section discusses the design of a new mechanism called **Topic Attention**, which increases the capability to strengthen the semantic correlation between queries information and responses. The overall structure of **Topic Attention** is shown **below**:

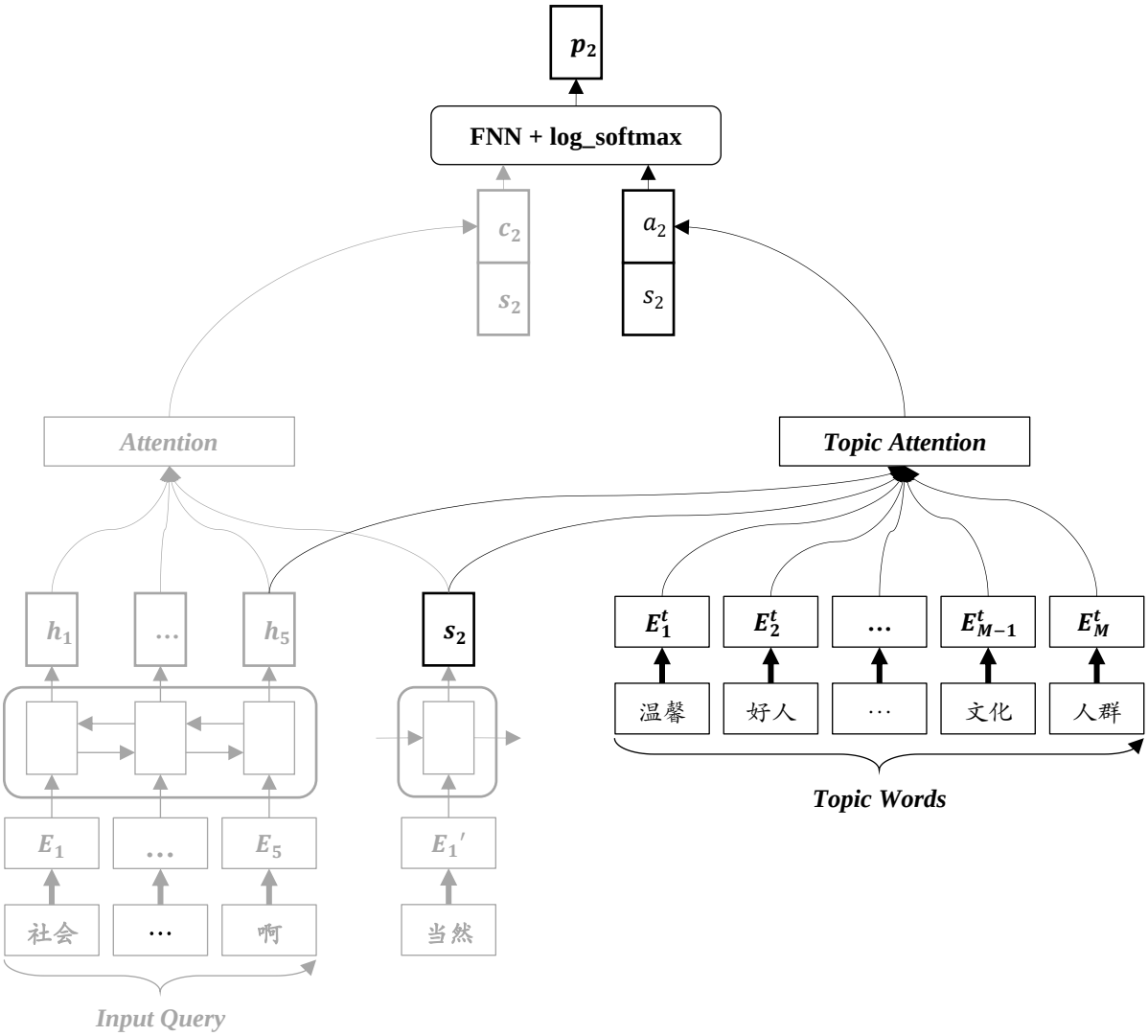

Figure 4 Architecture for Topic Attention

Topic Attention works through a series of processes and computations. First, it creates a sequence of topic attention vectors $A = (a_1, \dots a_L)$ dynamically paying attention to the information of the last hidden state $\boldsymbol{h_N}$ (called $\boldsymbol{t_0}$ in this section) from encoder topic, and topic words $T = (t_1, \dots t_M)$ generated in **Section 5.3**. This calculation method is similar to the one applied when computing the context vectors, which is defined in Equation 3. The process can be elaborated through the following formulas:

$$\beta_{lm} = \frac{score(s_l, t_m)}{\sum_{i=0}^{M} score(s_l, t_i)} \quad 11$$

$$a_l = \sum_{k=0}^{M} \beta_{lk} t_k \quad 12$$

For each attention vector a_l, we calculate the topic word probability distribution $P^t = (p_1^t, \dots p_M^t)$. These probabilities are added to corresponding probabilities figured by the original Seq2Seq $P^o = (p_1^o, \dots p_V^o)$ as below:

$$p_v^t = \begin{cases} p_m^t \; if \; word \; v \; in \; T, and \; m \; is \; the \; corresponding \; index \\ 0, else \end{cases} \quad 13$$

$$p_v = p_v^o + p_v^t \quad 14$$

The new outputs are defined as follows:

$$\widehat{r_l} \sim o_l = P(\widehat{r_l} | r_1, \dots, r_{l-1}, H, T) \quad 15$$

This idea is motivated by communication among humans. People reply with words implying the same topic of the previous utterance. For instance, when people talking about the weather, there are usually words in terms of weather, such as “sunny”, “bad”, and “umbrella”. Even if the reply is “I agree”, it omits what the replier agrees with, which exactly should contain specific key words. The principle of **Topic Attention** refers to such a pattern in communication. From the perspective of neural networks, a response is generated based on the odds that terms appear given a precondition. Thus, increasing the incidence of certain words, which have a strong relationship with the query, is able to optimize the replies.

In terms of the above, **Topic Attention** can be described as a dynamic method with disparate topic word sets for different queries. This design is inspired by the attention mechanism [29], thus these two techniques have similar configurations. However, there is one distinct difference. Compared with the attention mechanism, **Topic Attention** concentrates on the prediction of topics words probability distribution, rather than that of all vocabularies. This construct makes sense. On one hand, the objective of the novel technique is to increase the occurrence of these topic words in responses. On the other hand, the topic words generated according to the input sentence make less contribution to the contextual information, which makes the prediction of all vocabulary probabilities through topic words unreasonable. In comparison to the attention mechanism in **Figure 1**, **Topic Attention** can be embodied in the right black part in black in **Figure 4**.

5.6. External Memory

The emotional involvement in the dialog system is mainly accomplished by **External Memory**, which is also a dynamic method like **Topic Attention**. The design of **External Memory** can be illustrated in **Figure 5**:

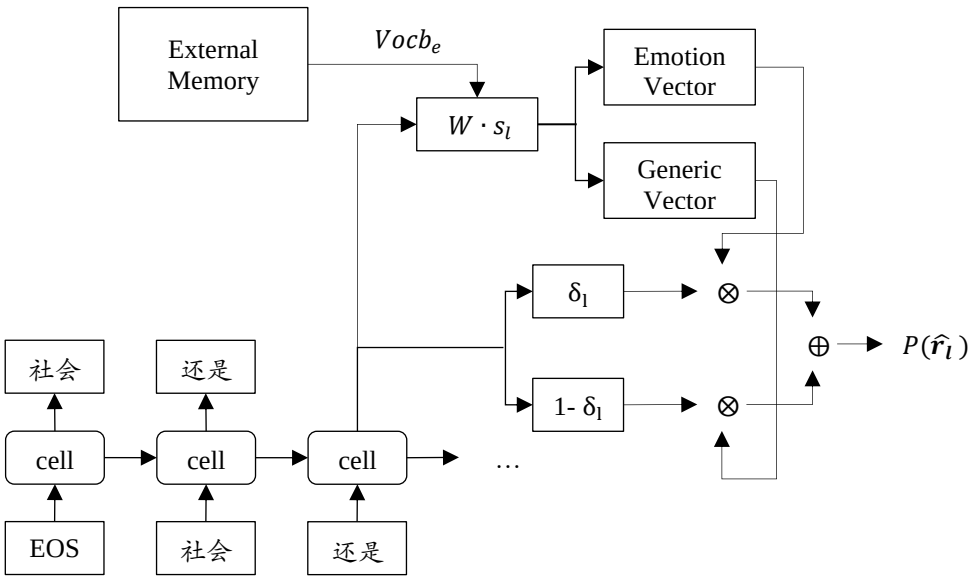

Figure 5 Architecture for External Memory

External Memory picks up vocabularies with strong certain sentiment, which has been described in **Section 5.2.1**. This mechanism assigns different probabilities to affective words and generic words in term of current hidden state values. In this way, the emotion of sentences can be reflected from updated word occurrences.

At the end part of the decode, given the current state $\boldsymbol{s_l}$, we employ a vector dot followed with a sigmoid function to generate a scale value. This value represents how much contribution of the emotional words we expect with the current state. The probability that we choose a sentimental word is defined as:

$$\delta_l = sigmoid(W_e \cdot s_l) \qquad 16$$

We sort the vocabulary and divide them into two categories in advance for acceleration. The two categories are: words with the assigned emotion $Vocb_e$ and the others. The coefficient for each word $\boldsymbol{w}$ is:

$$\varepsilon_l = \begin{matrix} \delta_l, if\ w\ in\ Vocb_e \\ (1-\delta_l), else \end{matrix} \quad 17$$

$$\widehat{r_l} \sim o_l = P(\widehat{\boldsymbol{r_l}}) = log\_softmax(\varepsilon_l(W_o \cdot s_l)) \quad 18$$

The **External Memory** is to add in affective factors at the decoder side, as **ECS** is engineered to generated responses with diverse emotional given one query. Sentimental computation in the decoder is more suitable for this objective. A similar idea to add in emotional factors when decoding is also applied in the designs of *Emotional Chatting Machine* [1] and *Affective Neural Response Generation* [2], which are both famous emotional response generation models.

5.7. Grammar Filter

To reduce the occurrence of some trivial terms, such as "你", "了", and "也", a **Grammar Filter** is proposed. These words frequently appear in almost each sentence, but they make little contribution to the semantic content. Without any extra disposal, the neural networks will generate sentences with these paltry terms frequently arising, which severely undermines the normal expression of responses. Therefore, I design a structure called **Grammar Filter** to deal with this problem. This technique reduces the occurrence of trivial by decreasing their probability distribution.

Given the current state in decoder $\boldsymbol{s_l}$ and the unnecessary words $Vocb_u$, we recalculate a coefficient:

$$\omega_l = sigmoid(W_u \cdot s_l) \quad 19$$

We obtain the new word probability distribution by multiplying this coefficient with the vector dot of words in $Vocb_u$.

6. Experiments and Evaluation

The proposed model **ECS** has been implemented with *Python* and *Pytorch*[4]. The source code is available at https://github.com/wangjs9/ecs.

This section analyzes the performance of **ECS**, and evaluates the generated replies at both content level and sentiment level through comparison mainly with the **baseline** models, which are defined in **Section** 错误!未找到引用源。. The testing dataset is introduced in **Section 3.2**. **Section 6.2** examines the content of replies from the aspects of word error rate and diversity. The performance of responses at the emotion level is verified in **Section 6.3**. Plus, a case study to show the features of **ECS** is exhibited and analyzed in **Section 6.4**.

6.1. Baselines and Ablation Study

Since the objectives of this paper are to equip the dialog system with sentiment and to reduce the repetitive patterns in emotional responses, whether sentiment and topic are involved in the model matters. Therefore, we adopt GRU Seq2Seq model as the baseline. To understand the efficacy of these components and analyze **ECS** thoroughly, we conduct an ablation study on **ECS**. Therefore, we implement six models, which are listed as follows:

- S2S: a basic GRU Seq2Seq model
- S2S + GF: a GRU Seq2Seq model with the Grammar Filter mechanism
- S2S + GF + TA: a GRU Seq2Seq model with the Grammar Filter and the Topic Attention mechanism
- S2S + GF + TA: a GRU Seq2Seq model with the Grammar Filter and the External Memory mechanism
- ECS – CVAE: a GRU Seq2Seq model with the Grammar Filter, the Topic Attention, and the External Memory mechanism
- ECS: a Seq2Seq model with emotional CVAE model, the Grammar Filter, the Topic Attention, and the External Memory mechanism

[4] *Pytorch* is an open source machine learning framework. https://pytorch.org/

6.2. Content Level Evaluation

This section evaluates how relevant and grammatical the output sentences of the **ECS** are. *BLEU* is a common evaluation metric in multiple text generation tasks [35]. However it has been proved that *BLEU* is not suitable for measuring dialog systems, as it correlates weakly with human judgments [36]. It has been proved that *Perplexity* and word error rate are related linearly and have good correlations [37], when the test sets close to the training set distribution. As mentioned, since the training dataset and testing dataset are randomly selected from the WeiBo corpus, they have similar characteristics and distribution. Thus, *Perplexity* is adopted as the evaluation metric for content correctness.

$$Perplexity = exp\,(-\frac{1}{L}\sum log\,P(w)), where\ w\ is\ the\ predicted\ word \qquad 20$$

Besides, a chatbot should reply to the users without limited and repetitive words and sentences. Thus, diversity of content should also be considered, which can be reflected in *Type-Token Ratio (TTR)* for unigrams and bigrams [38].

$$TTR = \frac{number\ of\ distinct\ words\ (pairs)}{number\ of\ total\ tokens\ (pairs)} \times 100\% \qquad 21$$

The results are shown in **Table 3**:

Model	**Perplexity**	**TTR**	
		Unigrams	Bigrams
S2S	222.22	0.10%	0.36%
S2S + GF	196.55	0.31%	1.47%
S2S + GF + TA	190.09	1.56%	6.34%
S2S + GF + EM	185.87	0.71%	3.39%
ECS – CVAE	159.01	2.38%	10.47%
ECS	**132.08**	**2.56%**	**12.73%**
Testing Corpus (Human)		9.68%	57.29%

Table 3 Evaluation of Content

As can be observed, **ECS** performs best among all models at the content level in term of word correctness and diversity.

From these results, there are several consequential points: After the participation of **Grammar Filter**, there is a significant decline of *perplexity*, which is decreased by *11.55%*, and a remarkable

increase in *TTRs*. **Topic Attention** plays a critical role in the increasing of *TTRs*, which grow by hundreds of percent over those of **Grammar Filter**. These two comparisons prove that the reduction of trivial words and the increase of essential terms benefit the quality of responses in content. Another thing worth mentioning is that **External Memory** overmatches **Topic Attention** according to perplexity, which is an effect opposite to the expectation of my design. However, the lower perplexity influenced by **External Memory** depends on receptive phases and patterns in replies.

Compared with other models, **ECS** is better at generating fluent, grammatical, and diversiform responses. The diversity of responses can be enhanced by **ECS**, which is superior to models without considering conversation topic or emotion.

6.3. Sentiment Level Evaluation

Sentiment level evaluation is another important part. The emotion accuracy, defined as a comparison between emotion tags for generated replies and the expected ones, is the evaluation for emotional correctness. To predict the emotions of sentences, an emotion classifier is designed, as mentioned in **Section 5.2.2**. This section will first discuss the performance of the emotion classifier and then evaluate **ECS** at the sentiment level with the prediction of this emotion classifier.

6.3.1. Emotion Classifier

This emotion classifier is implemented with *transformers*[5]. To train the accuracy of this emotion classifier, 1.2 million sentences are selected from the NLPCC dataset [28] as the training data, where the numbers of sentences belong to the six emotion categories equal. 25,000 sentences are randomly selected from the NLPCC dataset [28] as the testing dataset. The accuracies of the emotion classifier during training and testing are as follows:

Training Data		**Testing Data**	
Loss	Acc (100%)	Loss	Acc (100%)
0.41	94.07%	0.54	87.71%

Table 4 Emotion Classification Accuracy on the NLPCC Dataset

[5] *transformers* is a library provides general-purpose architectures for NLP. https://huggingface.co/transformers/.

This accuracy of emotion classifier is high enough to evaluate the performance of the emotional expression of **ECS**. Besides, the classification accuracy on the responses in the testing dataset of **ECS** is *88.18%*.

6.3.2. Emotion Accuracy

For better examination and comparison, I take into account two emotion tags $<e^1,e^2>$ of each query, which have the maximal probabilities predicted by the emotional classifier. The results of models are evaluated and compared with the assistant of 1st emotion accuracy and 2nd accuracy, which are defined as:

- For 1st emotion accuracy, if the actual emotion category of the sentence is e^1, it is regarded as a successful case; else, this sentence is regarded as a wrong output with a wrong emotion.
- For 2nd emotion accuracy, if the actual emotion category of the output sentence is either e^1 or e^2, this sentence has the right emotion.

The results are displayed in **Table 5**:

Model	1st Emotion Accuracy	2nd Emotion Accuracy
S2S	22.82%	39.26%
S2S + GF	26.22%	44.12%
S2S + GF + TA	23.94%	40.84%
S2S + GF + EM	**69.97%**	**81.35%**
ECS – CVAE	37.89%	54.54%
ECS	54.01%	76.45%
Testing Corpus (Accuracy of the Emotional Classifier)	88.18%	96.82%

Table 5 Evaluation of Sentiment

From these results, **External Memory** and the emotional CVAE encoder significantly contribute to emotion expression, while **Topic Attention** has a negative influence on the sentiment of responses. However, Moreover, higher emotion accuracy is not a consequent implication for preferable sentimental responses. The emotion effect generated by **External Memory** relies on modal particle or a single word, such as "哎唉！", "恨！", and "不错！". Although these replies may be logical and grammatical, they are monotonous due to the weak correlation with queries

and the high repeatability, which is one of the problems this paper tries to fix. Thus, although the ECS appears somehow inferiors compared to **External Memory + Grammar Filter**, the emotion accuracy is much higher than other model.

6.4. Case Study

This part reports a case study of replies generated by different models. Given an input query and an emotion category, we compare the replies generated by **GRU Seq2Seq** model and **ECS**, which are shown in **Table 6**. **ECS** is able to generate sentimental replies conditioned on emotion categories, and these responses have close relationship with the queries.

Query	**Model**	**Emotion**	**Response**
表回忆了，直接删除，新生活开始。 Don’t recall, put down memories of the past, and a new life will begin.	S2S		我也想哭了。 I want to cry, too.
	ECS	Like	加油！ Come on!
		Sad	所以说，希望会让人更幸福。 Thus, hope makes people happy.
		Disgusted	生活就需要生活。 Livelihood is life.
		Angry	矮油，别这样说。 Oh no, don’t say that.
		Happy	恭喜！ Congratulations!
Query	Model	Emotion	Response
这部剧真的值得看！我大爱！ This drama is really worth watching! I really love it!	S2S		我也喜欢的 I also like it.
	ECS	Like	喜欢！ Like it!
		Sad	可惜还没看 Unfortunately, I haven’t watched it
		Disgusted	是呀，现在还没看完 Yeah, haven’t finished it
		Angry	真的假的？ Really?
		Happy	哈哈！ Ha-ha!
Query	Model	Emotion	Response
糟蹋原作者的创作，不要脸。 Spoil the creation of the original author, what a shame!	S2S		我也觉得很无奈 I also feel helpless
	ECS	Like	赞！ Great!
		Sad	可怜滴孩子 A poor child
		Disgusted	没本事，不要脸！人肉 Stupid and shameless!
		Angry	信不信，反正我是信了。

			Believe it or not, I do.
		Happy	笑死人。 Hilarious!

Table 6 Response Generated with Different Emotions

From these three examples, the responses generated by **ECS** indeed have the same topic with the input queries. The sentiment is expressed visibly and varies according to the emotion labels coherently. In summary, **ECS** has a great balance between sentiment and content of responses.

7. Conclusion and Future Work

To improve the quality of responses generated by sentimental dialogue systems, a novel model **Emotional Chatting System** (**ECS**) is proposed to generate diversified responses with coherent content and consistent emotions.

This project comes up with an emotional CVAE encoder and three mechanisms at decoder side, which are **Topic Attention**, **External Memory**, and **Grammar Filter** respectively. The novel encoder captures emotional and contextual information, which improves response expression. Each mechanism at the decoder introduces one prior knowledge; thus, topic words, sentimental term and even trivial words provide additional information to this chatting system. In this model, a GUR Seq2Seq model is firstly constructed as the framework, and then extra pieces of knowledges are imported by assembling mechanisms at the decoder side of the model. The final model combines all of the extra information. Experiments and the evaluation results show that ECS can generate replies with related contextual information and correct emotions, and these replies contain the same topics with the input queries.

There are two directions for my future works:

Firstly, the current work can be improved. Current methods to obtain extra information can be optimized. For instance, the approaches to generated topic words can be replaced by a Twitter-LDA [39]. During training, I find that the topic distribution and the trivial words list influence the content of replies significantly. This influence may not be reflected by the evaluation metrics, it indeed can be observed by humans. In addition, dialogue systems trained by reinforcement learning is a trend [40, 41], therefore applying to reinforcement learning in this work can be an orientation in the future work.

Secondly, the scope of this project can be extended. The future work will explore the automatic emotion estimation: the model should determine the most proper sentiment class for the response. This work should involve the sentiment of the queries, which is not utilized in this project. The relationship of sentiment and context will be explored more detailed. In addition, the approach to obtain topic words can be optimized.

Reference

[1] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, "Emotional chatting machine: Emotional conversation generation with internal and external memory," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[2] N. Asghar, P. Poupart, J. Hoey, X. Jiang, and L. Mou, "Affective neural response generation," in *European Conference on Information Retrieval*, 2018: Springer, pp. 154-166.

[3] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," *arXiv preprint arXiv:1510.03055*, 2015.

[4] B. A. Shawar and E. Atwell, "Chatbots: are they really useful?," in *Ldv forum*, 2007, vol. 22, no. 1, pp. 29-49.

[5] P. B. Brandtzaeg and A. Følstad, "Why people use chatbots," in *International Conference on Internet Science*, 2017: Springer, pp. 377-392.

[6] E. Go and S. S. Sundar, "Humanizing chatbots: The effects of visual, identity and conversational cues on humanness perceptions," *Computers in Human Behavior*, vol. 97, pp. 304-316, 2019.

[7] J. D. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393-422, 2007.

[8] R. Yan, Y. Song, and H. Wu, "Learning to respond with deep neural networks for retrieval-based human-computer conversation system," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 55-64.

[9] Z. Ji, Z. Lu, and H. Li, "An information retrieval approach to short text conversation," *arXiv preprint arXiv:1408.6988,* 2014.

[10] A. Leuski, R. Patel, D. Traum, and B. Kennedy, "Building effective question answering characters," in *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, 2009: Association for Computational Linguistics, pp. 18-27.

[11] T.-H. Wen *et al.*, "A network-based end-to-end trainable task-oriented dialogue system," *arXiv preprint arXiv:1604.04562,* 2016.

[12] O. Vinyals and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869,* 2015.

[13] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *arXiv preprint arXiv:1503.02364,* 2015.

[14] Y. Song, R. Yan, X. Li, D. Zhao, and M. Zhang, "Two are better than one: An ensemble of retrieval-and generation-based dialog systems," *arXiv preprint arXiv:1610.07149,* 2016.

[15] O. Serrat, "Understanding and developing emotional intelligence," in *Knowledge solutions*: Springer, 2017, pp. 329-339.

[16] H. Prendinger and M. Ishizuka, "THE EMPATHIC COMPANION: A CHARACTER-BASED INTERFACE THAT ADDRESSES USERS'AFFECTIVE STATES," *Applied artificial intelligence,* vol. 19, no. 3-4, pp. 267-285, 2005.

[17] B. Martinovski and D. Traum, "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*, 2003, pp. 11-16.

[18] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1-135, 2008.

[19] F. Burkhardt, M. Van Ballegooy, K.-P. Engelbrecht, T. Polzehl, and J. Stegmann, "Emotion detection in dialog systems: Applications, strategies and challenges," in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, 2009: IEEE, pp. 1-6.

[20] F. Mairesse and S. Young, "Stochastic language generation in dialogue using factored language models," *Computational Linguistics*, vol. 40, no. 4, pp. 763-799, 2014.

[21] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[22] A. Ritter, C. Cherry, and W. B. Dolan, "Data-driven response generation in social media," in *Proceedings of the conference on empirical methods in natural language processing*, 2011: Association for Computational Linguistics, pp. 583-593.

[23] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104-3112.

[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[25] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "SemEval-2016 task 4: Sentiment analysis in Twitter," *arXiv preprint arXiv:1912.01973*, 2019.

[26] S. Ghosh, M. Chollet, E. Laksana, L.-P. Morency, and S. Scherer, "Affect-lm: A neural language model for customizable affective text generation," *arXiv preprint arXiv:1704.06851*, 2017.

[27] X. Zhou and W. Y. Wang, "Mojitalk: Generating emotional responses at scale," *arXiv preprint arXiv:1711.04090*, 2017.

[28] M. Huang. "Emotional Coversation Generation." Minlie Huang. https://biendata.com/ccf_tcci2018/datasets/ecg/ (accessed June 8th, 2019).

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[30] I. V. Serban *et al.*, "A hierarchical latent variable encoder-decoder model for generating dialogues," *arXiv preprint arXiv:1605.06069*, 2016.

[31] T. Zhao, R. Zhao, and M. Eskenazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," *arXiv preprint arXiv:1703.10960*, 2017.

[32] C. M. Wilt, J. T. Thayer, and W. Ruml, "A comparison of greedy search algorithms," in *third annual symposium on combinatorial search*, 2010.

[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[34] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111-3119.

[35] A. Stent, M. Marge, and M. Singhai, "Evaluating evaluation methods for generation in the presence of variation," in *international conference on intelligent text processing and computational linguistics*, 2005: Springer, pp. 341-351.

[36] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," *arXiv preprint arXiv:1603.08023,* 2016.

[37] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Communication*, vol. 38, no. 1-2, pp. 19-28, 2002.

[38] B. Richards, "Type/token ratios: What do they really tell us?," *Journal of child language*, vol. 14, no. 2, pp. 201-209, 1987.

[39] W. X. Zhao *et al.*, "Comparing twitter and traditional media using topic models," in *European conference on information retrieval*, 2011: Springer, pp. 338-349.

[40] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," *arXiv preprint arXiv:1606.01541,* 2016.

[41] H. Cuayáhuitl, "Simpleds: A simple deep reinforcement learning dialogue system," in *Dialogues with social robots*: Springer, 2017, pp. 109-118.