RUCIR at NTCIR-14 STC-3 Task

Xiaohe Li, Jiaqing Liu, Weihao Zheng, Xiangbo Wang, Yutao Zhu, and Zhicheng Dou*

School of Information, Renmin University of China, Beijing, China
{lixiaohe,jiaqingliu,zheng-weihao,xiangbo28,ytzhu,dou}@ruc.edu.cn

Abstract. This paper describes RUCIR's system in NTCIR-14 Short Text Conversation (STC) Chinese Emotional Conversation Generation (CECG) subtask. In our system, we use the Attention-based Sequence-to-Sequence(Seq2seq) method as our basic structure to generate emotional responses. This paper introduces: 1) an emotion-aware Seq2seq model, 2) several features to boost the performance of emotion consistency. Official results show that we are the best in terms of the overall results across the five given emotion categories.

Team Name. RUCIR

Subtasks. Chinese Emotional Conversation Generation

Keywords: Emotional Conversation Generation, Sequence to Sequence Model, Attention Mechanism, Copy Mechanism

1 Introduction

Human-computer conversation is one of the most challenging tasks in natural language processing (NLP). Particularly, short text conversation (STC) which simulates human real-life dialogues has attracted more and more attention.

STC can be defined as a kind of single-turn conversation formed by two short texts, with the initial utterance given by a human user and the response given by a computer. STC task (STC-1) is first proposed in NTCIR-12 [8], which was taken as an information retrieval (IR) problem and aimed to retrieve an appropriate response in the repository to reply a user-issued utterance. At NTCIR-13 [7], STC-2 encouraged the participants to combine retrieval-based methods and generation-based methods to make a response for a new user-issued utterance. This year, we participated in NTCIR-14 STC-3 CECG subtask [12]. Compared with the former tasks, CECG aims at generating emotional Chinese responses that are not only reasonable in content but also suitable in a specific emotion. The pre-defined emotion categories include *like, sad, disgust, anger, happy* and *other*.

In general, conversation systems can be categorized into retrieval-based and generation-based. Retrieval-based methods maintain a large repository of conversation data and consider the user-issued utterance as a query, then return

* Corresponding author: Zhicheng Dou.

a most proper response through information retrieval techniques. Generation-based methods generate responses with natural language generation models learned from the conversation data. A typical generation method is the sequence-to-sequence (Seq2seq) neural network model [4, 6, 10, 11]. The Seq2seq model generally incorporates an encoder and a decoder. The encoder is used to represent the input message as a vector, based on which the decoder generates a new response. The encoder and the decoder are usually constructed by recurrent neural networks (RNNs). Since the structure of RNN is naturally suitable to model time-series data, Seq2seq model can capture semantic and syntactic relations between user-issued utterances and responses. An attention mechanism is often used to enhance the model on learning patterns from data [1, 5].

In this work, we use Seq2seq with attention mechanism as our basic model to build the conversation system. Our system consists of four modules. The first one is a rule-based template in which important information such as entities, weather and other keywords are taken into account. The second module comprises multiple fine-tuned Seq2seq models to generate responses in different emotions respectively. The third module is a single emotion-aware Seq2seq model with the input of emotion factors and emotion keywords. Finally, a reranker is designed to select the final responses based on emotion scores and term similarities.

The rest of paper is organized as follows: We will introduce our system architecture in detail at first. Then we report the experimental results in Section 3. Finally, we will make a brief conclusion of our work.

2 System Architecture

2.1 Data Pre-processing

Good quality of training data is essential for training a good model. We first process the dataset and remove the noisy information that is useless or even harmful to model training.

We retain the word segmentation of the original dataset and filter out post-response pairs that are not Chinese or too short (the post or response with less than three characters).

Then we artificially check the data and summarize some patterns for meaningless responses. More specifically, we first identify the responses that contain: 1) emoji and kaomoji, 2) dialect and online buzzwords, 3) repeated expressions in sentence level and word level, 4) meaningless beginning of sentence such as "Yes"("恩恩"), "Yes"("是啊") and "Haha"("哈哈"), 5) mention and repost characters('@' or '//@'). We filter out these meaningless expressions and emotion icons in the original posts or responses, and replace the dialect and buzzwords with Mandarin based on the dictionary.

Since the Seq2seq model tends to generate trivial and meaningless responses which appear many times in dataset such as "Hahaha..." ("哈哈哈。。。") and "What's up?" ("怎么了?"), we remove sentences that occur more than 100 times and simplify tokens that continuously and repeatedly appear more than twice.

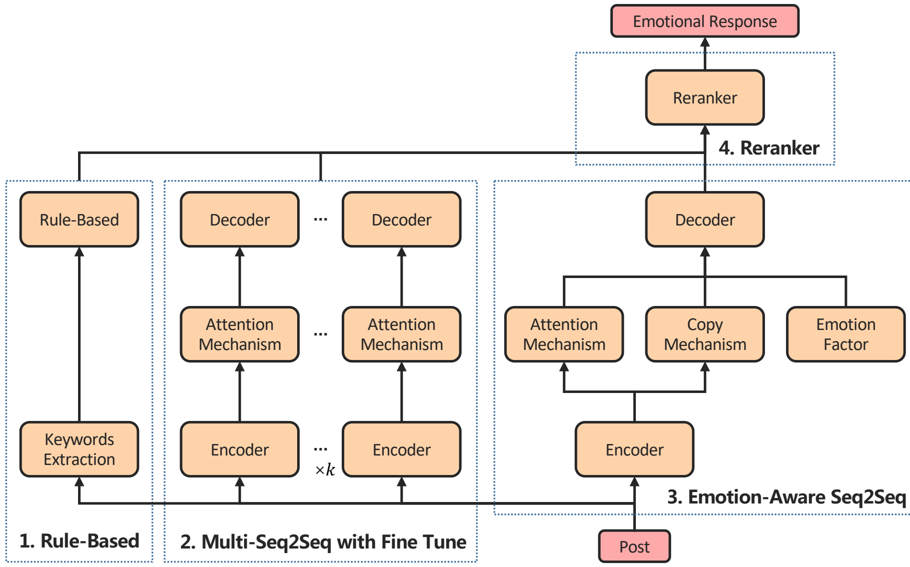

Fig. 1. The structure of our system

For example, "What's up?" ("怎么了?") appears 4,014 times in responses, thus all post-response pairs with such a response are removed from the dataset. And "Hahahahahaha......"("哈哈哈哈哈哈。。") is simplified as "Haha.." ("哈哈。。").

2.2 Seq2seq Model

The Seq2seq model is originally proposed for machine translation [10]. Then Shang et al. applied this model into neural response generation [6]. After that, tremendous approaches have been proposed for response generation based on the Seq2seq model [4, 11]. In this work, we also build our model based on it.

In general, the Seq2seq model consists of an encoder and a decoder. Both of them can be implemented with RNN and its variations such as long-short term memory (LSTM) [3] and gated recurrent unit (GRU) [2]. We use the GRU in this work, which can be formulated as

$$\begin{aligned} z &= \sigma(W_z x_t + U_z h_{t-1}), \\ r &= \sigma(W_r x_t + U_r h_{t-1}), \\ s &= \tanh(W_s x_t + U_s(h_{t-1} \circ r)), \\ h_t &= (1-z) \circ s + z \circ h_{t-1}. \end{aligned} \tag{1}$$

Assume $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ is a sequence of input post containing n words, and $\mathbf{y} = (y_1, y_2, \cdots, y_m)$ is a generated response. The encoder transforms $\mathbf{x}$ into a sequence of hidden states $h = (h_1, h_2, \cdots, h_n)$, which is defined as:

$$h_t = \mathrm{GRU}_{\mathrm{encoder}}(x_t, h_{t-1}), \tag{2}$$

where h_t is the hidden states of the encoder at time step t .

The decoder is another GRU maximizes the conditional probability of a target word y_t, which can be formulated as:

$$p(y_t|\{y_1, y_2, \cdots, y_{t-1}; \mathbf{x}\}) = p(y_t|s_t) = \mathrm{softmax}(W_o s_t), \tag{3}$$
$$s_t = \mathrm{GRU}_{\mathrm{decoder}}(y_{t-1}, s_{t-1}), \tag{4}$$

where s_t is the hidden states of the decoder at time t. y_0 is the start of sentence (SOS) token and s_0 is equal to h_n.

Attention Mechanism is often used to improve the model on learning patterns from data [1, 5]. In a vanilla Seq2seq model, the decoder generates the next word y_t only depends on the word y_{t-1} and the s_{t-1} at time step $t-1$. Since s_0 is equal to the final encoder hidden state h_n, the useful information of words in the front part of source sequence is neglected. Besides, at different decoding steps, vanilla Seq2seq model can not measure the importance of different words in the source sequence. On the contrary, in an attention mechanism, each word y_t corresponds to a context vector c_t calculated by a weighted sum of the encoder hidden states h, which can be formulated as:

$$s_t = \mathrm{GRU}_{\mathrm{decoder}}(y_{t-1}, s_{t-1}, c_t), \tag{5}$$
$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j, \tag{6}$$
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})}, \tag{7}$$
$$e_{ij} = \tanh(W_e[s_{t-1}; h_j]). \tag{8}$$

Emotion Factor We concatenate the emotion category embeddings as the emotion factor to each decoding step, which can introduce additional emotion information when generating a response with a given emotion. Therefore the decoder can generate responses more emotional under the given emotion while predicting the next word. Each emotion factor is represented by a real-valued, dense and randomly initialized vector. With the emotion factor, the decoder can be updated as:

$$s_t = \mathrm{GRU}_{\mathrm{decoder}}(y_{t-1}, s_{t-1}, c_t, e_i), \tag{9}$$

where e_i is the emotion embedding of the specific emotion category i.

Copy Mechanism Intuitively, emotion expressions usually have some distinct emotion words. For example, “I lost sleep last night.” (“我昨晚失眠了。”) and

"I was so sad about insomnia last night." ("昨晚失眠了，我好难过。"). Both of them express sadness, but the former one seems to describe a fact, while the latter one expresses sadness directly. Therefore, we can divide emotion expressions into implicit expressions and explicit expressions. Apparently, an emotion word is so expressive that can be easily perceived and recognized by humans. Zhou et al. used type selector which can control the distribution of generic and emotion words to control the generation of emotion words [13].Song et al. applied copy mechanism to enrich the useful and informative words in conversation system [9]. Inspired by these, we use copy mechanism to increase the probability of emotional words in the generation process and make expressions more emotional. The final generated word probability distribution is given by:

$$p(y_t|s_t) = p_{\mathrm{ori}}(y_t|s_t) + p_{\mathrm{emo}}(y_t|s_t, E), \tag{10}$$

$$p_{emo}(y_t|s_t, E) = \mathrm{softmax}(EW_e s_t), \tag{11}$$

where p_{ori} is the original probability distribution, p_{emo} is the additional emotion words distribution. If y_t is not an emotional word, the corresponding probability p_{emo} would be zero. E is the word embeddings of words in emotion vocabulary. W_e are the parameters for matching E and s_t. The composition of emotion vocabulary will be introduced in Section 2.5.

2.3 Fine Tune Multi-Seq2seq Models

The previous part of our system uses only one Seq2seq model. In this part, we train six vanilla Seq2seq models with attention mechanism for six emotion categories respectively to generate responses under different emotion categories. All these models are pre-trained on all post-response pairs in dataset and fine tune on pairs with the specific emotion. Given a user-specific emotion, the corresponding model generates some candidates with this emotion. All candidates including those generated by other models will be fed into a ReRanker, which will be introduced later.

2.4 Rule-based Model

When we express emotions, our emotions usually point to some objects . So keywords and objects in posts are important to generate responses. We use RUCNLP[1] tool to extract entities, and find keywords based on dictionary and rules. These keywords and objects are combined with artificial templates to generate more fluent and point-explicit responses with emotions. If keywords are detected, these rule-based responses will be ranked at the top.

2.5 ReRanker

Now we have many candidates generated by models. We design a reranker to rank these candidates and the response with the highest score will be selected as the

[1] http://183.174.228.47:8282/RUCNLP/

final reply. As we aforementioned, emotion words is extremely important. Thus we consider two metrics in reranker, namely emotion score and term similarity. Based on the emotional vocabulary ontology library published by DUTIR [14], and emotion words extracted by χ^2 value from different emotion text data, we construct emotion vocabulary with corresponding weights for {*Like, Sad, Disgust, Anger, Happy*}. The weights reflect the importance of words in the specific emotion (e.g., "happiness" has greater weight than "joy" in happiness emotional dictionary), where the value of weight is composed of the weight given in library and the frequency in training set. And we also consider the degree words to adjust emotional score and categorize them into different levels, which can increase, decrease or reverse the emotional expression, e.g., "very" ("很"), "a little" ("一点"), "not" ("没有"). The degree levels are reduced in the order of most, very, especially, little, inverse and others and the weight are set to 2, 1.5, 1.25, 0.5, -1, 1, respectively. Therefore given a sentence, the emotion score is calculated by:

$$\epsilon_m = \prod_{j \in index(m-1,m)} l_{y_j} \cdot \gamma_m \cdot w_m, \tag{12}$$

$$\mathcal{E}(\mathbf{y}) = \sum_{i=1}^{M} \epsilon_m, \tag{13}$$

where M are emotion words in the candidate response $\mathbf{y}$. $\mathcal{E}(\mathbf{y})$ and ϵ_m are the emotion score of candidate $\mathbf{y}$ and emotion word m, respectively. $index(m-1,m)$ is the index scope from the previous emotion word to the current emotion word. $index(0) = 0$ for first word. l_{y_j} is the level of degree word y_j in this scope to reflect the influence of increasing, decreasing or reversing the original emotion. w_m is the weight of emotion word m. γ_m indicates whether the emotion word m is in its corresponding emotional category. If the word m in the corresponding dictionary of emotion (e.g., "happy" for happiness emotion), then we set γ_m as 1 to reflect this positive effect. Otherwise, we set γ_m as -1 to reflect the negative effect (e.g., "sad" occur in happiness emotion).

However, only emotional score can not measure the quality of response comprehensively. For example, given "I won the prize." ("我获奖了。") as post, "I am so happy and excited." ("我非常开心和激动。") may get higher emotional score, but "I am very happy that you won the prize." ("我为你获奖而感到开心。") is more appropriate with coherent information than the former response. Therefore, we calculate the term similarity between response and post to encourage our model generate results with consistent information. We select the number of same terms between the response and the post as the measure of consistency.

$$\mathcal{T}(\mathbf{y}) = Count(\mathbf{x}, \mathbf{y}), \tag{14}$$

where $Count(\cdot)$ counts the same term between post $\mathbf{x}$ and candidate response $\mathbf{y}$. Finally, the ranking score of $\mathbf{y}$ is computed by:

$$\Phi(\mathbf{y}) = \lambda\mathcal{E}(\mathbf{y}) + (1-\lambda)\mathcal{T}(\mathbf{y}), \tag{15}$$

wthere the λ is set to 0.2 after many tests verified.

3 Experiment and Analysis

3.1 Implementation and Submissions

We submit 2 runs in this task. The settings of each run are shown as follows:

- RUCIR_1: a combination of candidates from full version Seq2seq model, multi-Seq2seq model and rule-besed model introduced in Section 2.2, 2.3, 2.4 respectively, then reranked by ReRanker to get the top one.
- RUCIR_2: the top candidate of full version Seq2seq introduced in Section 2.2. This is submitted as a baseline for RUCIR_1.

The released dataset contains 1,719,207 Weibo post-response pairs. After data pre-processing, there are 1,603,167 pairs in our dataset. We randomly select 5,000 pairs as validation set and testing set respectively. The rest pairs compose training set. We construct two separate vocabularies for posts and responses by using 10,000 most frequent words on each side, covering 95.98% and 96.38% usage of words for posts and responses respectively. And the emotion vocabulary size is 500 in total for all emotions. The words out of vocabulary are replaced with a special token "<UNK>". We use Tensorflow[2] to implement all models. A four-layered GRU cell with 1,024 dimensions is employed for both the encoder and the decoder. The dropout probability is set to 0.3. All model parameters are initialized with uniform distribution in [-0.08, 0.08]. Word embeddings and emotion embedding are randomly initialized and learned during training with 200 dimensions and 50 dimensions respectively. All candidates are generated using beam search with 10 beam width. We train the models on NVIDIA TITAN Xp GPU using the Adam optimizer with an initial learning rate 5e-4 and a decay factor 0.9. The batch-size is 64.

3.2 Results and Analysis

In the NTCIR-14 STC-3 CECG subtask [12], the submitted post-response pairs are evaluated by human annotation. The evaluation metrics are Fluency, Coherence and Emotion Consistency. The evaluation set has 200 posts and we submit the responses in five emotion categories except *other*.

Table 1 shows the overall results of all runs in CECG. Table 2 shows the top 3 runs of emotion-specific results on each emotion category. We can see that RUCIR_1 achieves best performances in overall results and in four of the five emotion-specific results. Moreover, the performance under the *happy* emotion category is also very close to the top. Even without rule-based module and multi-Seq2seq candidates, our emotion-aware Seq2seq with emotion information still achieves fourth in all runs. And RUCIR_2 has more label 1 terms than

[2] https://www.tensorflow.org

Table 1. Official CECG subtask results of the overall score and average score.

Team Name	Label 0	Label 1	Label 2	Total	Overall Score	Average Score
1191_1	581	320	99	1,000	518	0.518
1191_2	831	109	60	1,000	229	0.229
AINTPU_1	716	200	84	1,000	367	0.336
CKIP_1	845	29	126	1,000	281	0.281
CKIP_2	840	28	132	1,000	292	0.292
IMTKU_1	580	248	172	1,000	592	0.592
IMTKU_2	954	32	14	1,000	60	0.060
TMUNLP_1	777	126	97	1,000	320	0.320
TUA1_1	443	293	264	1,000	821	0.821
TUA1_2	454	278	268	1,000	814	0.814
WUST_1	601	211	188	1,000	587	0.587
WUST_2	999	0	1	1,000	2	0.002
TKUIM_2	507	260	233	1,000	726	0.726
RUCIR_1	392	263	**345**	1,000	**953**	**0.953**
RUCIR_2	460	**342**	198	1,000	738	0.738

Table 2. Top 3 runs of official emotion-specific results on each emotion.

Emotion Category	Team Name	Label 0	Label 1	Label 2	Total	Overall Score	Average Score
Like	RUCIR_1	88	36	76	200	**188**	**0.940**
	RUCIR_2	96	44	60	200	164	0.820
	TKUIM_2	90	56	54	200	164	0.820
Sad	RUCIR_1	72	48	80	200	**208**	**1.040**
	TUA1_1	84	31	85	200	201	1.005
	RUCIR_2	83	57	60	200	177	0.885
Disgust	RUCIR_1	71	76	53	200	**182**	**0.910**
	TUA1_2	92	82	26	200	134	0.670
	TUA1_1	82	105	13	200	131	0.655
Anger	RUCIR_1	88	63	49	200	**161**	**0.805**
	TKUIM_2	112	45	43	200	131	0.655
	TUA1_2	85	107	8	200	123	0.615
Happy	TUA1_2	76	25	99	200	**223**	**1.115**
	TUA1_1	71	36	93	200	222	1.110
	RUCIR_1	73	40	87	200	214	1.070

others which means the responses generated by RUCIR_2 are more coherent and fluent. These prove the effectiveness of our model. We can infer that our improved seq2seq model can guarantee the fluency and coherence of response at least. And the keywords extracted from the posts bring more emotional information to the responses and improve the performances.

4 Conclusion

In this paper, we introduce our approaches in the CECG subtask of NTCIR-14 STC-3 task. We introduce an emotion-aware Seq2seq model with emotion factors and emotion words to generate responses. And we use the emotion score as an addition feature to rerank the response candidates. The experimental results verify the effectiveness of our methods.

In the future, we will focus on several aspects: extracting other types of information from sentences, building a more advanced model to combine keywords extraction and keywords placement during training.

Acknowledgements

Zhicheng Dou is the corresponding author. This work was supported by National Key R&D Program of China No. 2018YFC0830703, National Natural Science Foundation of China No. 61872370, and the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China No. 2112018391.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
4. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055 (2015)
5. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
6. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. arXiv preprint arXiv:1503.02364 (2015)
7. Shang, L., Sakai, T., Li, H., Higashinaka, R., Miyao, Y., Arase, Y., Nomoto, M.: Overview of the ntcir-13 short text conversation task (2017)
8. Shang, L., Sakai, T., Lu, Z., Li, H., Higashinaka, R., Miyao, Y.: Overview of the ntcir-12 short text conversation task pp. 473–484 (2016)
9. Song, Y., Li, C.T., Nie, J.Y., Zhang, M., Zhao, D., Yan, R.: An ensemble of retrieval-based and generation-based human-computer conversation systems. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. pp. 4382–4388. International Joint Conferences on Artificial Intelligence Organization (7 2018)

10. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3104–3112. Curran Associates, Inc. (2014)
11. Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W.Y.: Topic aware neural response generation. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
12. Zhang, Y., Huang, M.: Overview of NTCIR-14 short text generation subtask: Emotion generation challenge. In: Proceedings of the 14th NTCIR Conference (2019)
13. Zhou, H., Huang, M., Zhang, T., Zhu, X., Liu, B.: Emotional chatting machine: Emotional conversation generation with internal and external memory. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
14. 徐琳宏, 林鸿飞, 潘宇, 任惠, 陈建美: 情感词汇本体的构造. 情报学报 **27**(2), 180–185 (2008)