TKUIM at NTCIR-14 STC-3 CECG task

Shih-chieh Wei[1], Chi-bin Cheng[2], Guang-zhong-yi Cao, Yi-jing Chiang, Chin-yi Wu, Shih-hsiang Lin, Kun-li Tsai

Tamkang University, Taiwan, R.O.C.

{sekewei[1], cbcheng[2]}@mail.tku.edu.tw

Abstract. In this work, we will report how we (TKUIM) built a system for the sub-task CECG of STC-3. Our system mainly consists of two parts, the response generation subsystem and the emotion classification subsystem. For the response generation subsystem, we trained five generative models using different training parameters. These models will output response candidates based on a Seq2Seq deep learning architecture with the attention mechanism. For the emotion classification subsystem, we trained an emotion classifier with probability output for each emotion class. According to the desired response emotion class, a corresponding emotion classifier is used to select the most probable response from the previous response candidates. An emotion accept threshold and a default response library are set up for each response emotion class. When the selected response does not pass the emotion accept threshold, a default response from the library for that emotion class is output to replace the poorly generated response. In this mission, we submitted only one valid result, which got an average total score of 0.726 within a maximum scale of 2.

Team Name. TKUIM

Subtasks. Chinese Emotional Conversation Generation (CECG)

Keywords: Natural Language Generation, Deep Learning, Sentiment Analysis

1 Introduction

Since the 12th NTCIR in 2016, a new evaluation task based on Chinese and Japanese corpora was set up. Until now this new Short Text Conversation Task (STC) is still the only open conversational competition in the world. In the 13th NTCIR STC2 competition, for the Chinese evaluation task, the champion team from Sogou and Tsinghua University used the S2SAttn model to achieve unprecedented results in the generation of conversation text (Zhao et al., 2017). This model is based on a Seq2Seq deep learning architecture that combines the attention mechanism and is superior in performance to the search retrieval model.

In the 14th NTCIR14 STC3, a subtask called Chinese Emotional Conversation Generation (CECG) is proposed. The subtask aims to generate conversation text of a desired emotion class. In order to produce the conversation text required for the task, we built a system composed of response generation and emotion classification. This system will be described in detail in Section 2.

2 Emotional Generative Model

The system architecture used in this task is shown in Figure 1.

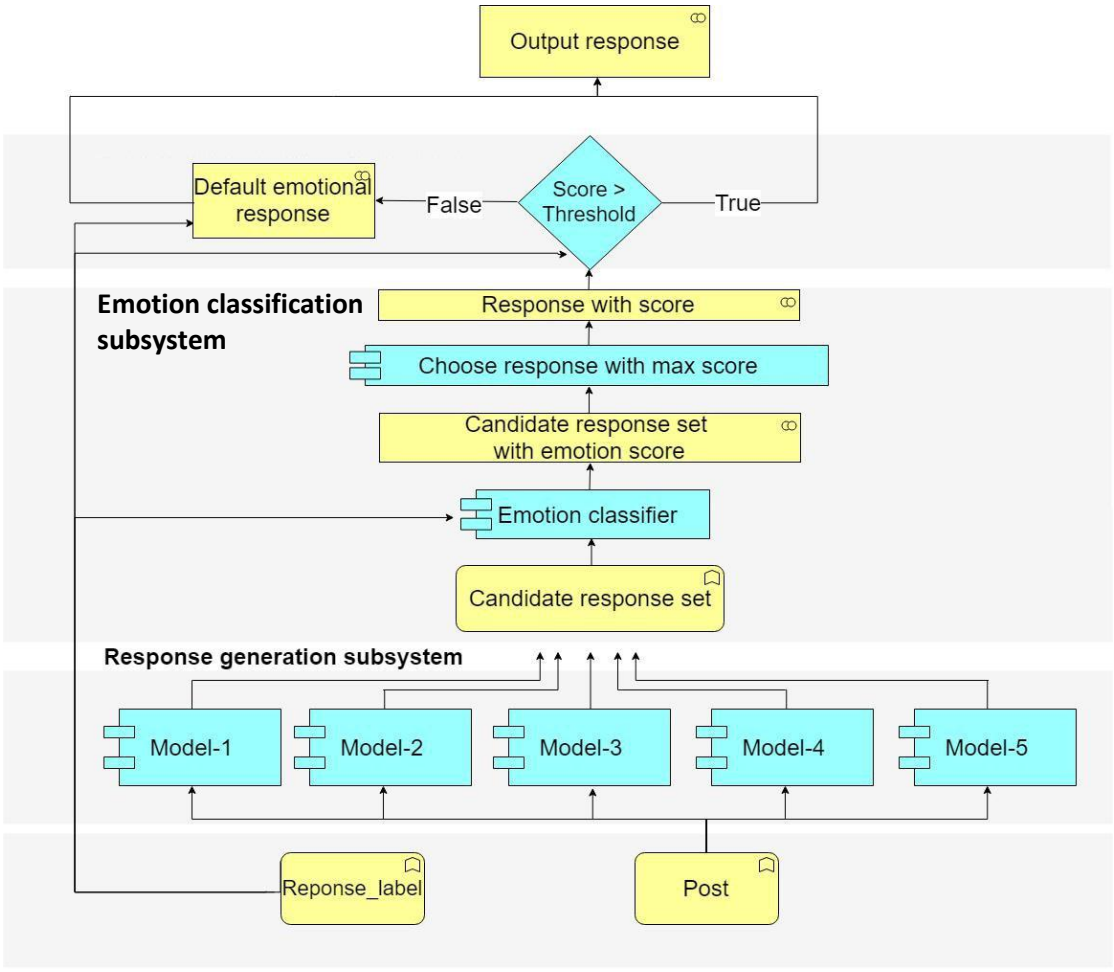

Figure 1. The system architecture used in this task

The lower part of Figure 1 is our response generation subsystem, and the upper part is the emotion classification subsystem. For each post we will generate the response through the five Seq2Seq models, which constitute the candidate response set, all of which are responses to the same input.

The candidate response set will then be input to the emotion classification subsystem, producing the expected probabilities for a desired emotion class (excluding emotion class 'Others'). Then we choose the response with the maximum probability.

For this probability, we set an emotion accept threshold for each emotion class. The threshold determines whether the response can be the final output. If the response with the highest probability cannot pass the emotion accept threshold for the desired emotion class, it will be replaced with a random sentence from the library of default emotional responses for the desired emotion class.

2.1 Generative-model-based chatbot

In recent years, many new models of chatbots have emerged in the NLP field. Most chatbots are based on generative models, which can be improved under the Seq2Seq framework.

There are two main advantages of this Seq2Seq framework. One is that this framework consists of two similar parts, which are easier to correct and expand. The other is that the generative model can also be adjusted according to the training corpus.

In the following we will introduce several techniques related to the generative chatbot used in this study. They include the sequence to sequence model, the encoder and decoder, the attention mechanism, word embedding and sentiment classification.

2.2 Sequence-to-sequence model

The sequence-to-sequence network or encoder decoder network is a model consisting of two separate recurrent neural networks (RNNs) called encoders and decoders.
The encoder layer transforms the word sequence of the input into a context vector, which ideally can be understood as the "meaning" of the entire input sequence. The decoder layer decodes the sentence from the context vector. So the model is free from the sequence length constraint.

2.3 Encoder and decoder

The encoder of the Seq2Seq network is essentially an RNN that outputs a vector and a hidden state from the previous hidden state for each word in the input sentence. The gated recurrent unit (GRU) is often used in the RNN structure to solve the problem of vanishing gradients.
At present, GRU performs very well in language modeling and machine translation tasks. Although its performance is not decisive compared with long short term memory (LSTM), it is certain that GRU has advantages in simplicity and can effectively reduce the time cost of model training.
For the decoder layer, we used a GRU decoder with attention mechanism. Luong et al. (2015) proposed three global attention models to improve the traditional attention model. The difference among the three global attention models is in the way of calculating attention scores.
The method used to calculate the score in this study is direct use of the dot operation, which, as the name suggests, is a simple inner product between hidden states of encoder and decoder.

2.4 Attention mechanism

The fixed length context vector bears the burden of coding the entire "meaning" of the input sequence, no matter how long it may be. This is a very difficult problem in various languages.
Imagine two almost identical sentences, twenty words long and only one word different. Both the encoder and the decoder must use the context vector to represent this change as a slightly different point in space.
Bahdanau et al. (2016) proposed a mechanism of attention to solve the long text problem. By training, the decoder can learn to pay attention to a specific part of the input sentence instead of relying on a single context vector.

2.5 Word embedding

Word embedding refers to the transformation of words into vectors in a high dimensional space through a specific training algorithm. After the transformation, neighboring words in the corpus will occupy adjacent vector points in the space. In this way, we can use the distance between words to indicate relevance.
There are two most commonly used word vector training methods: CBOW and Skip-Gram. In our Seq2Seq architecture, we use a separate word embedding layer for the encoder and the decoder, respectively.

2.6 Sentiment classification

In the subsystem of emotion classification, we use a deep learning architecture of convolutional neural network (CNN) with an embedding layer. Kim (2014) mentioned a series of well-performing CNN text classification experiments. The embedding is based on Word2Vec, and the hyper-parameters of deep learning are rarely modified. Kim pointed out that the training of word vector is one of the most important components of sentiment classification.
As the corpus in this competition consists of a large amount of online everyday conversation text, we train the word vectors for each of the five emotion classifiers, respectively. The dimension of word vectors is 128.

3 Experiment

We divided the 600,000 official post-response pairs into five segments of training sets, and assigned them to five experimental groups for training five generative models using different parameters. The generated response sentences of the five models are

collected as the candidate response set, which is input to the next subsystem.

3.1 Pre-processes

For the response generation subsystem, our focus is on the fluency of the response and its coherence to the post. Therefore, we trained our model without use of the emotion label, but only extracted the post-response pairs as the training data for the response generation subsystem.

As for emotion classification subsystem, we mainly focus on the correlation between the conversation text and its emotion label. So for this part of the training, we do not distinguish between posts and responses. We take all of the posts, responses, and their emotion labels to train five classifiers for each emotion class, respectively.

For a specific emotion classifier, the conversation text with a label of that emotion class is marked as positive. The conversation text with labels of other emotion classes is marked as negative. The emotion classifier outputs the probability of the input text belonging to that emotion class. In addition, we give the same number of positive and negative sentences to guarantee fairness in the training process.

The format of the data set is as follows:

[[[post, post_label], [response, response_label]], [[post, post_label], [response, response_label]], ...].

A [[post, post_label], [response, response_label]] is called a post-response pair. The data set has a total of 600,000 post-response pairs. The 'post_label' denotes the emotion class of the post. The 'response_label' denotes the emotion class of the response. Both label values vary from 0 to 5, corresponding to the classes of 'Other', 'Like', 'Sadness', 'Disgust', 'Anger', and 'Happiness', respectively.

Take the first post-response pair of the data set as an example.

[['现在 刷 朋友 圈 最大 的 快乐 就是 看 代购 们 各种 直播 。 。 。 。 。', '5'], ['卧 槽 我 也 是', '4']]

In this post-response pair, the post is '现在 刷 朋友 圈 最大 的 快乐 就是 看 代购们 各种 直播 。 。 。 。 。', which has an emotion class of 5, or 'Happiness'. The response is '卧 槽 我 也 是', which has an emotion class of 4, or 'Anger'.

3.2 Emotion accept threshold and default emotional responses

The emotion accept threshold $T_{emotion}$ is used in the emotion classification subsystem to determine whether we accept the generated response based on the probability of the response belonging to the desired emotion class.

The exact value of $T_{emotion}$ for each emotion class depends on a certain quartile value of the output probability of an emotion classifier for the training set. For example, there are a total of 155,758 sentences whose emotion class is 1 or 'Like'. The emotion classifier for 'Like' will produce 155,758 probability values between 0 and 1. After these values are sorted in descending order, a certain quartile value is decided as the emotion accept threshold for the 'Like' emotion. In this way, the emotion accept thresholds for the five emotion classes can be different for tuning purposes.

In addition, for each emotion class, a library of default emotional responses is manually selected from the official training data set. When each sentence in the candidate response set is assigned a probability by the desired emotion classifier, the sentence with the highest probability is selected as the response. However, if the response with the highest probability cannot pass the emotion accept threshold for the desired emotion class, it will be replaced with a sentence from the library of default emotional responses for the desired emotion class.

3.3 Test and Evaluation Metric

The submitted post-response pairs are evaluated by the following metrics.

Emotion Consistency: whether the emotion class of a generated response is the same as the pre-specified class.

Coherence: whether the response is appropriate in terms of both logical coherence and topic relevance in content.
Fluency: whether the response is fluent in grammar and acceptable as a natural language response.
The labeling procedure is shown by the following pseudocode:

```
IF Coherence and Fluency
    IF Emotion Consistency
        LABEL 2
    ELSE
        LABEL 1
ELSE
    LABEL 0
```

All the submitted results were resorted to Baidu Data Crowdsourcing Service for manual evaluation, and each post-response pair is annotated by three curators. The average overall score for each run is computed by the following formula.

$$AvgOverallScore = \frac{1}{N_t}\sum_{i=0}^{2} i \cdot num_i$$

where num_i is the number of pairs which has a label of i for each submission run, and N_t is the total number of pairs for each run. Therefore, the highest *AvgOverallScore* for each run will be 2.
Since the data set has a relatively small proportion of sentences belonging to emotion class 4 or 'Anger', the response generative model is accepted at a lower rate for this class, resulting in more frequent use of the default sentence for the emotion class 'Anger'. Table 1 shows part of the result we submitted in this competition. The column names from left to right denote the post, the emotion class of the post, and five responses to the post. 'ans1' to 'ans5' are the responses corresponding to emotion classes 'Like', 'Sadness', 'Disgust', 'Anger', and 'Happiness' in order.
The red rectangle in ans4 shows the responses of emotion class 'Anger'. The blue rectangle in ans5 shows the responses of emotion class 'Happiness'. It can be found that we use a lot of default sentences in these two emotion classes.

Table 1. Partial results submitted

	question	qEmo	ans1	ans2	ans3	ans4	ans5
70	睡得不想起,眼都不想睁肿么办[打哈欠]	3	我爱你	悲摧啊	是噢,我今天中午睡到3点半醒[哼]	什么情况?[怒]	哈哈,真的假的?
71	死了,还有好多没背,背了就忘。。。怎么搞的	3	我也觉得很好看!	唉,没法活了,不容易啊	哎···同感同感~~有我们文学系的多东西背吗[泪]	什么情况?[怒]	哈哈,真的假的?
72	心情坏得衣服都不想换脸也不要了就这样睡吧[哈欠]	3	哈哈,你是我的错了	你不觉得我很委屈啊	[哼]我也想睡	什么情况?[怒]	不拘小节,我很欣赏
73	真是奇怪,这种时候我居然还吃的下去。。。	3	我也是哎,我也是	我还没吃呢,我还没吃呢	晕,啥时候应该吃不下去?	什么情况?[怒]	哈哈,真的假的?
74	讨厌被质问的感觉...	3	哈哈,我也是	我也很无奈的	全部人都是这样想的	什么情况?[怒]	哈哈,真的假的?

4 Conclusion

Aiming to achieve Explainable Artificial Intelligence (XAI), our system is split into a response generation subsystem and an emotion classification subsystem. In this way, the results of each subsystem can be clearly observed for independent model design and tuning. In the response generation subsystem, we produce 5 sentences from each of the 5 generative models to ensure the fluency of the response sentences and their coherence to the post sentence. The emotion classification subsystem ensures that the response of choice will meet the desired emotion class. As Table 2 shows, in the CECG subtask of the NTCIR-14 STC3, the average overall score of our TKUIM team is 0.726. For emotion class 'Like', the average score is 0.82. For emotion class 'Sadness', the average score is 0.65. For emotion class 'Disgust', the average score is 0.63. For emotion class 'Anger', the average score is 0.63. Finally, for emotion class 'Happiness', the average score is 0.875. In this work our contribution is design and implementation of a dialogue system that can produce fluent response sentences of the desired emotion class. Our emotion classification subsystem can be easily expanded by other emotion tags, like casualness, respect, or other specific commercial use cases.

Table 2. Overall results and emotion-specific scores

Team Name	'Like' Score	'Sad' Score	'Disgust' Score	'Anger' Score	'Happy' Score	Overall Score
TKUIM	0.82	0.65	0.63	0.63	0.875	0.726

5 Acknowledgments

This research was supported in part by the TKU research grant. We would like to thank the support from College of Business and Management, Tamkang University, Taiwan, R.O.C.

References

1. Bahdanau, D., Cho, K., Bengio,Y. "Neural Machine Translation by Jointly Learning to Align and Translate," International Conference on Learning Representations (ICLR), (2016).
2. Cho, K., Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014).
3. Goyal, P., Pandey, S., Jain, K. Deep Learning for Natural Language Processing: Creating Neural Networks with Python, 1st ed., Apress, (2018).
4. Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y. "Exploring the Limits of Language Modeling," arXiv:1602.02410, (2016).
5. Kim, Y. "Convolutional Neural Networks for Sentence Classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746-1751, (2014).
6. Luong, M., Pham, H., Manning, C. D. "Effective Approaches to Attention-based Neural Machine Translation," Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 1412-1421, (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," NTCIR-13, (2017).
8. Sutskever, I., Vinyals, O., Le, Q. "Sequence to Sequence Learning with Neural Networks," NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, 3104-3112, (2014).
9. Thanaki, J. Python Natural Language Processing, Packt, (2017).
10. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z. "Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-based Chatbots," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 495-505, (2017).

11. Zaccone, G. Getting Started with Tensorflow, Packt, (2016).
12. Zhang, Y.,Huang, M. “Overview of the NTCIR-14 Short Text Generation Subtask: Emotion Generation Challenge,” Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies, (2019).
13. Zhao, H., Du, Y., Li, H. “SG01 at the NTCIR-13 STC-2 Task,” NTCIR-13, (2017).
14. Zhou, H., Huang, M., Zhang, T., Zhu, X., Liu, B. “Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory,” AAAI, (2018).