WUST at the NTCIR-14 STC-3 Dialogue Quality and Nugget Detection Subtask

Ming Yan, Maofu Liu, Junyi Xiang

School of Computer Science and Technology, Wuhan University of Science and Technology, Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System,
Wuhan 430065, China
liumaofu@wust.edu.cn

Abstract. The purpose of dialogue quality is to test the degree of completion and satisfaction of dialogue. Nugget detection aims at automatically identifying the status of dialogue sentences from the dialog system, such as problem extraction, problem solving and so on. Existing methods rely on feature extraction tools, which can result in error accumulation, but also ignore the context dependency between dialogues and the semantic information of sentences, which is helpful for the detection of dialogue quality and nugget detection. In this paper, a neural network method is proposed to extract the context dependency between dialogues by Bi-LSTM, attention mechanism is adopted to learn key sentences or phrases in dialogues. The two kinds of information are combined to improve the quality of dialogues and the recognition ability of nugget detection. The experimental results of STC-3 DQ and ND subtask in NTCIR-14 show that our proposed method is effective.

Keywords: Dialogue Quality, Nugget Detection, Neural Network, Attention Mechanism.

Team Name: WUST

Subtask: STC-3 dialogue quality and nugget detection (Chinese)

1 Introduction

Dialogue quality [1] aims to build an evaluation system to evaluate a task-oriented, multi-round, text-based dialogue system. Nugget detection [2] aims to extract the state of each sentence from multiple dialogues, due to obtain the customer's intention and helpdesk's effectiveness. The existing systems are all evaluation systems in specific areas, which are difficult to apply to other areas universally. Therefore, it is necessary to establish a universally applicable dialogue evaluation system. According to the task description of STC-3 DQ and ND subtask in NTCIR-14: DQ is to determine whether the dialogue is completed, whether the customer is satisfied with the answer, and whether the customer's question has been solved? ND needs to find out the corresponding state of each statement, the problem, the solution of the problem and so on. In order to better understand the evaluation task of STC-3 DQ and ND subtask in NTCIR-14, a complete structure of the dialogue is given below Example1:

Example 1：

{"annotations": [
{"quality": {"A": 0,"S": 0,"E": -2},
"nugget": ["CNUG0","HNUG","CNUG"]},
"turns": [
{"sender": "customer",
"utterances": ["请问 SJM 成员的 henry 的 SOLO trap 为什么不打榜？首先这是韩文歌曲 如果是因为不知道打哪个榜之前的打榜分会补回来吗 因为是实时打榜的规则 @音悦 Tai 客服"]},
{"sender": "helpdesk",
"utterances": ["亲,数据是有保留的."]},
{"sender": "customer",
"utterances": ["可是实时打榜很吃亏"]}]
}

This is a three-round dialogue, and manual annotation only extracts one of them. Its dialogue quality evaluation indicators are A (Accomplishment), S (Satisfaction), E (Effectiveness) three subjective indicators. The scale range of each indicator is -2 to +2, -2 means the worst, and + 2 means the best. For example, the results of manual labeling, in this case, are A:0, S: 0, E: -2. Nugget detection has seven tagging states. The nugget types are shown in the following Table1.The three-round dialogue in Example1 are CNUG0, HNUG, and CNUG respectively.

Table 1. Nugget types.

State	
CNUG0	Customer trigger (problem stated)
CNUG*	Customer goal (solution confirmed)
HNUG*	Helpdesk goal (solution stated)
CNUG	Customer regular
HNUG	Helpdesk regular
CNaN	Customer Not-a-Nugget
HNaN	Helpdesk Not-a-Nugget

2 Related work

Dialogue quality and nugget detection can help machines better understand natural languages, aiming to help them get important information from the text. Dialogue quality and nugget detection are just beginning to be studied. In 1997, Walker et al. [3] proposed PARADISE, an evaluation system for task-based dialogues, which incorporates some features such as the number of continuous rounds of dialogues into the linear equation [4]. In addition, an evaluation equation is defined to evaluate the task-based dialogue system.

Most of the nugget detection methods regard this problem as a classification task and construct a classifier based on some features of the text [5]. However, the natural language processing tools and resources that extract features may make errors and pass these errors to the final classification. The neural network has a strong feature and semantic learning ability. It can automatically learn text representation from data. Nguyen et al. [6] proposed a convolutional neural network to analyze nugget in the text. This method constrains the context to a fixed window, resulting in the loss of word meaning representation in long sentences.

In this paper, Bi-LSTM is used to extract the context dependency between dialogues. Meanwhile, the attention mechanism is adopted to learn key sentences or phrases in dialogues. These two kinds of information are combined to improve the quality of dialogues and the recognition ability of nugget detection. The experimental results of STC-3 DQ and ND subtask in NTCIR-14 show that the method is effective.

3 System Description

Based on the analysis of the data set, we consider the DQ and ND subtask as a classification problem. We refer to the baseline system[1] and modify the structure of the system. We adopted the attention mechanism to the baseline system. The structure of the DQ system is shown Figure 1.

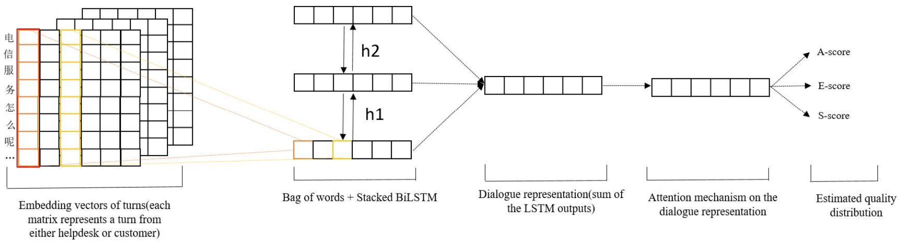

Fig. 1. System Architecture.

3.1 Data Preprocessing

In the data preprocessing part, this part mainly deals with segmentation and deactivation. In the text segmentation module, Jieba[2] participle is used to segment the data. The stop words[3] will cause noise interference to the effective information of the text, and the deactivation words have no special meaning. In order to reduce the noise interference caused by deactivation words to the sentences, first remove the stop words before extracting the features of the text.

---

[1] https://github.com/sakai-lab/stc3-baseline
[2] http://pypi.python.org/pypi/jieba/
[3] https://github.com/goto456/stopwords

3.2 Network Structure

In order to get all the information of the dialogue, we add the word vectors after word segmentation to sentence vectors. Graves et al. [7] proposed in 2005 that Bi-LSTM is composed of a forward-trained LSTM and a backward-trained LSTM. In this way, the information of "past" and "future" of the text can be retained at the same time. In this paper, we use three layers of Bi-LSTM to overlay, so that we can fully learn the semantic information and context dependency between dialogues.

LSTM has three gates: input gate, forgetting gate and output gate. The input gate determines which information is stored in the state of the neuron according to x_t, h_{t-1} and C_{t-1}. The forgetting gate decides which information is discarded from the state of the previous moment, and the output gate decides what the final output vector h_t is. Specific formulas are as follows: h_{t-1} and C_{t-1} are the output information and status information of the previous moment, x_t is the input of the current moment, h_t, i_t and O_t are the output information of three gates, and h_t and Ct are the output information and status information of the current moment. Forgotten Gate calculates a vector f_t with dimension n based on input x_t and output h_{t-1} at the last moment. Through this vector, we can decide which information will be discarded and which information will be retained.

The forgetting gate takes the input x_t of the current layer and the output h_{t-1} of the upper layer as input, and calculates which part of the information needs to be forgotten. The state output at $t-1$ time is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

When the neural network forgets part of the state,i_t needs to add new information from the current state. There are two main parts: a) the sigmoid layer of the input gate determines the value to be updated, and the information to be updated is expressed in i_t; b) the tanh layer creates a new vector $\tilde{C}_t$ to be added to the current state. Then the old state is multiplied by f_t, the information that needs to be forgotten is lost, and the information candidate information $i_t * \widetilde{C_t}$ is added to get the updated state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\widetilde{C_t} = \sigma(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (4)$$

Once the status C_t is updated, the output h_t of the current hidden layer can be calculated from the current output gate O_t.

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = O_t * tanh(C_t) \quad (6)$$

In long short-term memory neural networks, the transmission of States is one-way, that is, from the forward to the backward. However, in some news corpus, the state of the current moment is not only related to the previous state, but also to the later state, that is, the context of clauses needs to be fully utilized. At this point, bi-direction LSTM is used to solve this kind of problem.

In this paper, attention mechanism is introduced to extract important information and selectively ignore useless information. Finally, the vectors of this information are combined as output to further improve the performance of the model.

The attention model [8] is a model that simulates the attention of the human brain. In the field of natural language processing, the model is mainly used to express the correlation between words in text sentences and output results, so that the model focuses on important information in data during training, then enlarges important information, and selectively ignores unimportant information, so as to better represent text information and make the trained model achieve better performance. The specific calculation formula is shown below.

$$u_t = tanh(W_w h_t + b_w) \tag{7}$$

$$\alpha_t = \frac{exp(u_t^T u_w)}{\sum_t exp(u_t^T u_w)} \tag{8}$$

$$V = \sum_t \alpha_t h_t \tag{9}$$

Among them, W_w and b_w in formula (7) are the adjustable weights and paranoid terms of attention model, h_t is the output of Bi-LSTM layer, u_w in formula (8) is also the weight value, and the calculated result α_t represents the important information of each word in the sentence. The V in formula (9) is the output vector calculated by the attention model. Then classify by softmax function.

4 Experiments

4.1 Settings

In this paper, the training and test dataset, supplied by the STC-3 DQ and ND subtask in NTCIR-14, are 3700 and 390 instances separately. The hyper-parameters of the experiment are shown in **Table 2**.

Table 2. The settings of hyper-parameters.

Hyper-parameters	Value
word embedding dimension	300
learning rate	0.0003
dropout	0.2
batch-size	32
The number of hidden layers	300

Since the classes of DQ subtask are non-nominal, cross-bin measures is more suitable for bin-by-bin measures. As discussed by Sakai [9], bin-by-bin measures such as Jensen-Shannon Divergence are not adequate for this subtask as they do not consider the distance between classes. Thus, we utilize two cross-bin measures: Normalized Match Distance (NMD) and Root Symmetric Normalized Order-aware Divergence (RSNOD)

In contrast to DQ subtask, the classes in ND subtask are nominal, so bin-by-bin measures should be more suitable. Specifically, two measures are used in ND subtask: Root Normalized Sum of Squares (RNSS) and Jensen-Shannon Divergence (JSD).

4.2 Experimental results

We submitted the results of Chinese DQ and ND tasks for NTCIR-14 STC-3. The results given by the organizers are as follows Table 3 to 6.

Table3. The results of A-score in dialogue quality.

Run	Mean RSNOD	Run	Mean NMD
WUST-run0	0.1251	WUST-run0	0.0836
WUST-run1	0.1274	WUST-run1	0.0860
WUST-run2	0.1263	WUST-run2	0.0845

Table4. The result of S-score in dialogue quality.

Run	Mean RSNOD	Run	Mean NMD
WUST-run0	0.1226	WUST-run0	0.0779
WUST-run1	0.1270	WUST-run1	0.0808
WUST-run2	0.1248	WUST-run2	0.0779

Table5. The result of E-score in dialogue quality.

Run	Mean RSNOD	Run	Mean NMD
WUST-run0	0.1200	WUST-run0	0.0780
WUST-run1	0.1236	WUST-run1	0.0828
WUST-run2	0.1167	WUST-run2	0.0774

Table6. The result of nugget detection.

Run	Mean JSD	Run	Mean RNSS
WUST-run0	0.0223	WUST-run0	0.0909
WUST-run1	0.0233	WUST-run1	0.0931
WUST-run2	0.0250	WUST-run2	0.0980

The results of this experiment are calculated by the evaluation tools provided by the organizers. As shown in the table, WUST-run0 performs better on A-score and S-score, but not statistically significantly different from WUST-run1 and WUST-run. However, WUST-run2 performs better on E-score, and WUST-run0 performs better on ND.

Bi-LSTM can get the context dependence in the dialogue very well, so it can judge the degree of completion of the dialogue. Attention can catch the information of thanks（“感谢”） and thank you（“谢谢你”） from the dialogue customers, and can judge whether the dialogue is effective or not and the satisfaction of the customers in the dialogue. But because of using BOW to get sentence vectors, it may lead to two sentences with completely different meanings, but get the same sentence vectors, which will lead to the final wrong evaluation score and classification.

5 Conclusions

This paper proposes a neural network method, which uses Bi-LSTM to extract the context dependency between dialogues, and adopts attention mechanism to learn the key sentences or phrases in the dialogue, and combines these two kinds of information to improve the dialogue quality and the identification of nugget detection ability. The experimental results of STC-3 DQ and ND subtask in NTCIR-14 show that the method is effective.

References

1. Zeng, Z., Kato, S., Sakai, T.: Overview of the NTCIR-14 Short Text Conversation Task: Dialogue Quality and Nugget Detection Subtasks, Online Proceedings of NTCIR-14, to appear, 2019.
2. Qi, L., Heng, J., Liang, H.: Joint event extraction via structured prediction with global features[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Aug 4-9, 2013. Stroudsburg: ACL, 2013:73-82
3. Marilyn, A., Diane, J., Candace, A, et al.: PARADISE: A framework for evaluating spoken dialogue agents. In: Proceedings of EACL, European, 1997, 271-280
4. Thomson, B., Young, S.: Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems[J]. Computer Speech and Language, 2010, 24(4):562-588.
5. Yu, H., Jianfeng, Z., Bin, M., et al: Using cross-entity inference to improve event extraction[C]. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Jun 19-24, 2011. Stroudsburg: ACL, 2011: 1127-1136
6. Yubo, C., Liheng, X., Kang, L., et al.: Event extraction via dynamic multi- pooling convolutional neural networks[C]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, Jul 26-31, 2015. Stroudsburg: ACL, 2015: 167-176.
7. Graves, A.: 2005 Special Issue: Framewise phoneme classification with bidirectional LSTM and other neural network architectures[M]. Elsevier Science Ltd. 2005.
8. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need[C]. In *Advances in Neural Information Processing Systems 30 (NIPS 2017).*
9. Sakai, T.: Comparing two binned probability distributions for information access evaluation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1073{1076. SIGIR '18, ACM, New York, NY, USA (2018)