WUST at the NTCIR-15 DialEval-1 Task

Xuan Zhang, Maofu Liu, Zhanzhao Zhou, Junyi Xiang
School of Computer Science and Technology,
Wuhan University of Science and Technology,
Wuhan 430065, China
liumaofu@wust.edu.cn

ABSTRACT

The NTCIR-15 Dialogue Evaluation Task (DialEval-1) hosts two subtasks, Dialogue Quality (DQ) and Nugget Detection (ND). The purpose of the DQ subtask is to assess the quality of the dialogue from three aspects. The ND subtask is to identify the current status of dialog turn. Both DQ and ND subtasks aim to evaluate customer-helpdesk dialogues automatically. In this paper, we use neural network to extract context dependency between dialogues by Bidirectional Long Short-Term Memory (Bi-LSTM), and adopt the attention mechanism in DQ subtask to learn the keywords and sentences better. Compared with the current feature extraction method which ignores the dependency between dialogues, our method holds the stronger emphasis on the context dependency. Finally, the experimental results of the two subtasks show that our method is effective.

KEYWORDS

Dialogue Quality, Nugget Detection, Neural Network, Bi-LSTM, Attention Mechanism

TEAM NAME

WUST

SUBTASK

NTCIR-15 DialEval-1 Dialogue Quality (DQ) and Nugget Detection (ND) Subtask (Chinese)

1 INTRODUCTION

According to the NTCIR-15 DialEval-1 Dialogue Quality and Nugget Detection task definition [1], Dialogue Quality subtask is an evaluation system that can automatically evaluate the task-oriented, multi-round, text-based dialogue between the customer and helpdesk. Nugget Detection is to classify whether a customer or helpdesk turn is a nugget, where being a nugget means that the turn helps towards problem solving. Given a customer-helpdesk dialogue, the DQ subtask needs to illustrate an estimated distribution of dialogue quality ratings for the entire dialogue in terms of three criteria, i.e. task accomplishment, customer satisfaction, and efficiency, and the ND subtask returns an estimated distribution of labels over nugget types for each turn. In order to better understand the evaluation task of NTCIR-15 DialEval-1 DQ and ND subtask, a complete dialogue sample is shown in Table 1 and Table 2 in the following Example 1.

Example 1:

Table 1: Dialogue quality score

quality indicators	score
A	0
S	0
E	1

Table 2: Details and nugget type for each turn

turns	sender	utterances	nugget
turn1	customer	@音悦 Tai 客服 这个该咋办啊？求救	CNUG0
turn2	helpdesk	亲,请问是只有这一个视频不能播放么？	HNUG
turn3	customer	不是,都不能,	CNUG
turn4	helpdesk	亲,麻烦提供下邮箱,我们给您传个测试版的客户端您安装试试.	HNUG
turn5	customer	好的,1550179050@qq.com	CNUG

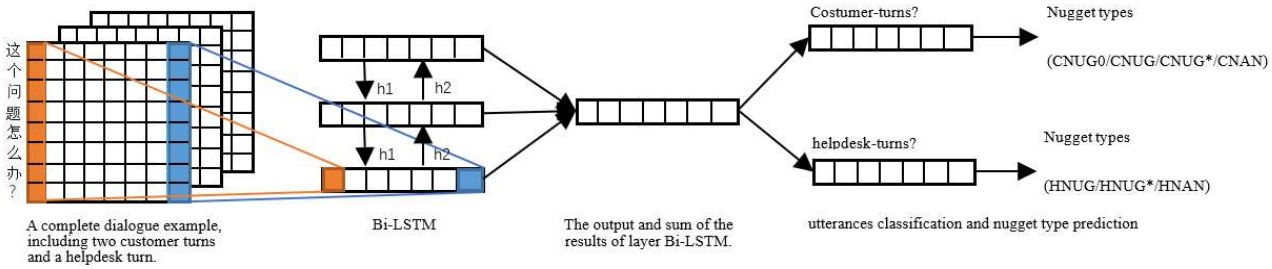

Figure 1: system structure

This is a sample dialogue with 5 turns, and its "id" is "3784236539025819". Each sample in the training and development dataset should have 19 annotation information, because there are 19 annotators annotating the conversation, while in the ground truth dataset, there are 20 annotators [2]. There are three dialogue quality assessment indicators, i.e. A-score, S-score, and E-score, they correspond to task accomplishment, customer satisfaction, and efficiency respectively. For each indicator, the possible options are [2, 1, 0, -1, -2], and our method needs to give specific scores for these three indicators among these options. In Example 1, the A-score, S-score, and E-score of the dialogue are 0, 0, and 1, respectively; the nugget types of each turn in order are: "CNUG0", "HNUG", "CNUG", "HNUG*", "CNUG". All nugget types have been shown in Table 3.

Table 3: Nugget types

Nugget type	Customer	Helpdesk
Trigger	CNUG0: tell the problem to Helpdesk	
Regular	CNUG	HNUG
Goal	CNUG*: tell Helpdesk that the problem has been solved	HNUG*: tell Customer the solution to the problem
Not-a-nugget	CNaN	HNaN

2 REALATED WORK

Effective DQ and ND systems can make machines understand natural language better, which can be used to construct effective helpdesk systems. In the early dialogue evaluate systems, the most widely used automatic method was based on comparing utterances with reference answers (Hirschman et al. 1990 [3]) [4]. PARADISE [5] (PARAdigm for DIalog System Evaluation) (Walker et al. 1997) is the most known general integrative evaluation framework proposed for task-oriented systems, which can be applied to any task-oriented system.

Generally, DQ and ND subtasks are both regarded as text classification problems and rely on feature extraction [6]. However, traditional text classification methods have many shortcomings. For example, relying on existing natural language processing tools can easily lead to the accumulation of errors in the processing process, which affects the final classification results [7]. Algorithms based on neural networks have obvious advantages in the field of natural language processing. Jin Wenzhen et al. [8] proposed a text classification method based on a deep learning feature fusion model, which uses convolutional neural networks and two-way gated recurrent units to extract the context information of the text, effectively extracts the semantic feature information between the texts, and reduces the text representation Impact on classification results.

In this paper, Bi-LSTM is used to learn semantic information and extract context dependencies. Using attention mechanism to learn important information in DQ subtask.

3 SYSTEM DESCRIPTION

In the NTCIR-15 DialEval-1 task, the evaluation system is partially improved on the basis of the original baseline[1]. The general framework of ND subtask system is shown in Figure 1.

3.1 Data Preprocessing

In the data preprocessing stage, first use Jieba[2] to segment the sentence into word representations, and remove low-frequency words and stop words[3], which is to eliminate noise. Appropriately reducing the frequency of stop words can effectively increase keyword density.

3.2 Network Structure

In order to extract important contextual semantic features, after word segmentation, a Bi-LSTM (Bi-directional Long Short-Term Memory) needs to be used for semantic feature extraction. In this paper, using three-layer Bi-LSTM repeated processing can more fully learn the semantic information in the dialogue, using attention mechanism to learn important information and ignore unimportant information.

[1] https://github.com/DialEval-1/LSTM-baseline
[2] https://github.com/fxsjy/jieba
[3] https://github.com/goto456/stopwords

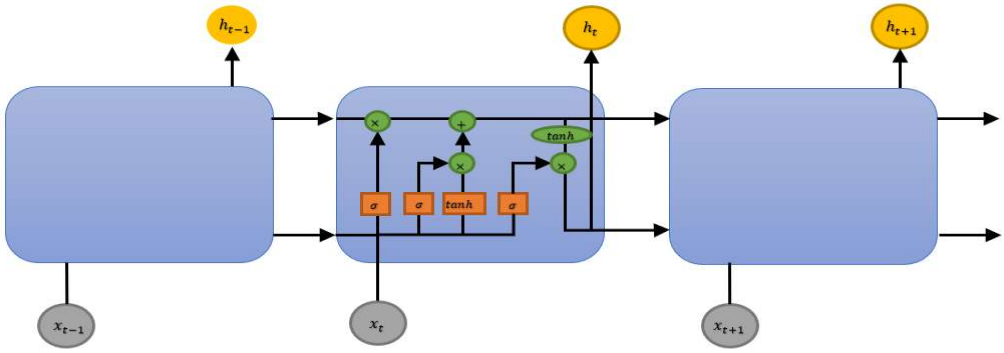

Figure 2: A simple LSTM model

Bi-LSTM consists of two independent LSTM. The input sequence will be input through two LSTM which are in positive and reverse order respectively. At the same time, the hidden layer output will be merged to obtain the hidden layer output vector at each time.

3.2.1 LSTM. Long short-term memory network (LSTM) is developed from recurrent neural network (RNN). In order to overcome the problem of gradient diffusion or gradient explosion in RNN, in 1997, Hochreiter et al. [9] proposed a long and short-term memory network, and introduced recurrent memory neurons (Memory Cell) on the basis of recurrent neural networks. A simple LSTM model is shown in Figure 2.

LSTM has three gate structures: forget gate, memory gate and output gate; two cell states: current cell state and temporary cell state; and a hidden layer.

The forget gate is used to control whether to forget the information from the previous time step. h_{t-1} (The state of the hidden layer at the previous moment) and x_t (the input word at this moment pass) through an activation function to obtain f_t (the output of the forget gate). The formula is as follows.

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

The memory gate decides whether to use the information from the previous time step. After inputting h_{t-1}, x_t, and f_t into the memory gate, we can obtain i_t (the value of the memory gate), $\widetilde{C_t}$ (the temporary cell state) and C_t (the current cell state) through different combination operations. The formula is as follows.

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

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C_t} \tag{4}$$

The output gate will output the hidden layer vector at the current moment. h_{t-1} and x_t pass through an activation function to obtain o_t (the value of the output gate).

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

Finally, the current cell state is combined with the output gate through the tanh function to obtain h_t (the current hidden layer state).

$$h_t = o_t \times \tanh(C_t) \tag{6}$$

3.2.2 Attention Mechanism. Using the attention mechanism to focus on the important part of the input and weaken the unimportant part, thus can help to obtain a representation that contains more semantic features. The calculation formula for the attention mechanism is shown below.

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

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

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

W_w, b_w and u_w are all variable parameters. h_t is the hidden layer state output from Bi-LSTM, input it into a small neural network to get u_t, and then through the $softmax$ function to get α_t, which represents the amount of attention should pay to each word. The H in formula (9) is the output vector calculated by the attention model.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Dataset Setting. We will use the DCH-1 data used for training and testing in the NTCIR-14 STC-3 task for training and development. There are 3700 training dialogues and 390 test dialogues. These conversations are real (i.e., human-human) customer-helpdesk dialogues collected from Weibo. On DialEval-1, the DCH-1 training data is still used for training, and the test data is used as development data to tune the model.

4.1.2 Hyper-parameters Setting. The hyper-parameters of the experiment are shown in Table 4.

4.1.3 Baseline Setting. There are four systems in the experiment, of which three BL are the official baselines:

BL-LSTM: A baseline model which leverages Bidirectional Long Short-term Memory;

BL-uniform: A baseline model which always predict the uniform distribution;

BL-popularity: A baseline model which predicts the probability of the most popular label as one, and predicts other labels as 0.

Note that the it accesses the golden truth to find the most popular label. This baseline is to show the upper bound of a single label.
From the results in the table, our model (WUST-run0) performs well on the ND subtask, but not very well on the DQ subtask.
4.1.3 Evaluation Metrics. For Dialogue Quality: Since the classes of DQ subtask are non-nominal, cross-bin metrics are more suitable than bin-by-bin metrics. As discussed by Sakai, bin-by-bin metrics 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 metrics: Normalized Match Distance (NMD) and Root Symmetric Normalized Order-aware Divergence (RSNOD). For Nugget Detection: In contrast to DQ subtask, the classes in ND subtask are nominal, so bin-by-bin metrics are more suitable. Specifically, two metrics are used in ND subtask: Root Normalized Sum of Squares (RNSS) and Jensen-Shannon Divergence (JSD).

Table 4: Hyper-parameters Settings

Hyper-parameters	Value
word embedding dimension	256
learning rate	0.001
dropout	0.3
batch-size	128
Optimizer	Adam
The number of hidden layers	150

4.2 Experimental results

We submitted the results of the Chinese DQ and ND subtasks of NTCIR-15 DialEval-1, and then we got the final evaluation results from the organizer. The results given by the organizer are shown in Table 5 to 8.

Table 5 to 7 are the results of the DQ subtask. It can be seen from the information in the table that the RSNOD evaluation results of the BL-LSTM and WUST-run0 system are similar, ranking first and second, and BL-LSTM is slightly ahead; but in the NMD evaluation, the A-score and S-score results of BL-popularity are both ahead of WUST-run0. This is because the BL-popularity model marks the most popular label as 1, and other labels as 0. Compared with BL-LSTM and WUST-run0 that use predicted probability to represent the value of each label, BL-popularity marks the score information more directly, so it has a better score in the NMD evaluation.

Table 5: Chinese Dialogue Quality (A-score) Results.

Run	Mean RSNOD	Run	Mean NMD
BL-LSTM	0.2345	BL-LSTM	0.1598
WUST-run0	0.2427	BL-popularity	0.1643
BL-popularity	0.2473	WUST-run0	0.1724
BL-uniform	0.2706	BL-uniform	0.2522

Table 6: Chinese Dialogue Quality (E-score) Results.

Run	Mean RSNOD	Run	Mean NMD
BL-LSTM	0.1782	WUST-run0	0.1386
WUST-run0	0.1795	BL-LSTM	0.1386
BL- uniform	0.2425	BL-popularity	0.1781
BL- popularity	0.2614	BL-uniform	0.2110

Table 7: Chinese Dialogue Quality (S-score) Results.

Run	Mean RSNOD	Run	Mean NMD
BL-LSTM	0.2088	BL-popularity	0.1442
WUST-run0	0.2131	BL-LSTM	0.1455
BL-popularity	0.2288	WUST-run0	0.1540
BL-uniform	0.2811	BL-uniform	0.2497

Table 8: Chinese Nugget Detection Results.

Run	Mean JSD	Run	Mean RNSS
WUST-run0	0.0695	WUST-run0	0.1633
BL-LSTM	0.0709	BL-LSTM	0.1673
BL-popularity	0.1301	BL-popularity	0.2068
BL-uniform	0.2858	BL-uniform	0.4190

From the Table 8, it is easy to see that the evaluation results of WUST-run0 and BL-LSTM on the JSD and RNSS indicators in the ND subtask are very close, and the results rank higher than other systems. Their Mean JSD is around 0.07, and Mean RNSS is around 0.16; While the other two systems' Mean JSD reaches around 0.13 or even 0.28, and Mean RNSS reached around 0.20 or even 0.42. This is because WUST-run0 and BL-LSTM both use Bi-LSTM to process the dialogue and extract the inter-context dependency information, while the other two systems do not. This shows that the use of Bi-LSTM to extract text dependencies between dialogues plays a very good role in classification tasks.

Let's look at a concrete sample shown in Example 2.

Example 2 shows a sample in the test dataset. Label each round of dialogue as Turn1, Turn2, Turn3, Turn4. The annotation information of 20 annotators on the nugget subtask is counted in Table 10 and Table 11.

Table 10 and Table 11 count how many annotators marked the current turn as the current nugget. For example, "14" in Table 10 means that 14 annotators marked "Turn1" as "CNUG0".

Example 2:

Table 9: A dialogue sample

turns	sender	utterances
turn1	customer	回复@携程客服:已经给过你们了//@携程客服:尊敬的@吃土少女李大 chen ,您好,请提供下订单号、详情,以便游游为您核实处理.
turn2	helpdesk	,您好,请问您是通过哪个渠道反馈的?
turn3	customer	@携程机票客服 就她
turn4	helpdesk	好的,已为您反馈@携程机票客服 .

Table 10: Statistics of Customer turns

Turn	CNUG0	CNUG	CNUG*	CNaN
Turn1	14	4	0	2
Turn3	0	19	0	1

Table 11: Statistics of Helpdesk turns

Turn	HNUG	HNUG*	HNaN
Turn2	18	1	1
Turn4	10	5	5

Now, in order to prove the effect of extracting contextual relevance and attention mechanism, we use three systems to conduct comparative verification. The three systems are as follows:

System 1: A system which uses the Bi-LSTM structure;
System 2: A system which only uses the simple LSTM structure;
System 3: A system which uses both Bi-LSTM structure and attention mechanism.

Comparing the evaluation results of System 1 and System 2, we can get the effect of extracting context relevance on improving the evaluation performance; comparing the evaluation results of System 1 and System 3 on the DQ subtask, we can get the effect of attention mechanism on improving the evaluation performance.

The results of all the three systems are evaluated according to the official evaluation method. The comparison of the evaluation results of System 1 and System 2 on the ND and DQ subtasks is shown in Table 12 to 15; the comparison of the evaluation results of System 1 and System 3 on the DQ subtasks is shown in the Table 12 to 14.

Result 1:

Table 12: Dialogue Quality (A-score) Results.

Run	Mean RSNOD	Run	Mean NMD
System 1	0.2427	System 1	0.1723
System 2	0.2398	System 2	0.1769
System 3	0.2314	System 3	0.1679

Table 13: Dialogue Quality (E-score) Results.

Run	Mean RSNOD	Run	Mean NMD
System 1	0.1794	System 1	0.1385
System 2	0.1962	System 2	0.1515
System 3	0.1716	System 3	0.1299

Table 14: Dialogue Quality (S-score) Results.

Run	Mean RSNOD	Run	Mean NMD
System 1	0.2130	System 1	0.1539
System 2	0.2272	System 2	0.1632
System 3	0.2226	System 3	0.1548

Table 15: Nugget Detection Results.

Run	Mean JSD	Run	Mean RNSS
System 1	0.0695	System 1	0.1633
System 2	0.0883	System 2	0.1972

From the evaluation results, the evaluation effect of System 1 (using Bi-LSTM) is slightly better than that of System 2 (using only simple LSTM), especially in the ND subtask. Table 12 to 14 showing the evaluation results of the DQ subtasks. The System 1 has no advantages in the evaluation of A-score, but in the evaluation of the other two indicators (E-score and S-score), Mean RSNOD and Mean NMD both dropped by about 0.01 to 0.02; Table 15 showing the evaluation results of the ND subtasks. Compared with System 2, Mean JSD and Mean RNSS of the evaluation results of System 1 both dropped by about 0.02 to 0.03. This shows that System 1 has better performance. In other words, using Bi-LSTM to extract contextual relevance has a good effect on the performance of the system.

In addition, the evaluation result of system 3 (using Bi-LSTM and attention mechanism) is slightly better than that of system 1 (using Bi-LSTM), but the optimization effect is not obvious, and the evaluation effect of some indicators is almost the same or even worse. Perhaps because the extraction of text information after Bi-LSTM is sufficiently comprehensive, so the effect of using the attention mechanism to improve the system evaluation is not obvious.

However, because the system uses BOW to obtain sentence vectors, this may cause two sentences with completely different meanings to have the same vector expression, which means that the problem of polysemous words cannot be solved, such as the sentence "这系统不大好用" have two completely different meanings, "这系统/不大好用" means that the system is bad, and "这系统不大/好用" means that the system is light and easy to use. The error caused by this situation is likely to cause the system to give a wrong evaluation score. Just like the sample just given, two different meanings of the sentence show the user's low satisfaction and high satisfaction respectively. This kind of ambiguity problem makes the system's running result on DQ subtasks not ideal.

5 CONCLUSIONS

This paper uses a neural network method, Bi-LSTM to extract the text dependence between dialogues, and the attention mechanism to learn key information. Synthesize the acquired semantic information and use it to deal with DQ and ND subtasks that are regarded as text classification problems. The final experimental results prove that this method is effective in the NTCIR-15 DialEval-1 task.

REFERENCES

[1] Zeng, Z., Kato, S., Sakai, T.: Overview of the NTCIR-15 Dialogue Evaluation (DialEval-1) Task, Conference'17, July 2017, Washington, DC, USA.

[2] Zeng, Z., Kato, S., and Sakai, T.: Overview of the NTCIR-14 Short Text Conversation Task: Dialogue Quality and Nugget Detection Subtasks, Proceedings of NTCIR-14, pp.289-315, 2019.

[3] Hirschman L, Dahl DA, McKay DP, Norton LM, Linebarger MC (1990) Beyond class A: a proposal for automatic evaluation of discourse. In: Proceedings of the speech and natural language workshop, Hidden Valley, Pennsylvania, USA, HLT, pp 109–113.

[4] Deriu, J., Rodrigo, A., Otegi, A. et al. Survey on evaluation methods for dialogue systems. Artif Intell Rev (2020). DOI:https://doi.org/10.1007/s10462-020-09866-x.

[5] Walker MA, Kamm CA, Litman DJ (2000) Towards developing general models of usability with PARADISE. Nat Lang Eng 6(3–4):363–377. DOI:https://doi.org/10.1017/S1351324900002503.

[6] Ming Y, Maofu L, Junyi Xi. WUST at the NTCIR-14 STC-3 Dialogue Quality and Nugget Detection Subtask. NTCIR-14 Conference: Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies, June 10-13, 2019 Tokyo Japan.

[7] You Y, Yu F, Xiaoping W. A review of Chinese text classification methods[J]. Journal of Network and Information Security, 2019,5(05):1-8. DOI:10.11959/j.issn.2096-109x.2019045.

[8] Jin, W., Zhu, H., Yang, G.: An Efficient Character-Level and Word-Level Feature Fusion Method for Chinese Text Classification. 2019, 1229(1). DOI:10.1088/1742-6596/1229/1/012057.

[9] Hochreiter S, Schmidhuber J (1997), Long short-term memory. Neural Computation 9:1735–1780. DOI:https://doi.org/10.1162/neco.1997.9.8.1735.

[10] Ashish V, Noam S, Niki P, Jakob U, Llion J, Aidan N. G, Łukasz K, and Illia P. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.