RSLNV at the NTCIR-15 Dialogue Evaluation (DialEval-1) Task

Ting Cao
Waseda University, Japan
caoting@toki.waseda.jp

Fan Zhang
Waseda University, Japan
zhangfan@fuji.waseda.jp

Haoxiang Shi
Waseda University, Japan
Hollis.shi@toki.waseda.jp

Zhaohao Zeng
Waseda University, Japan
zhaohao@fuji.waseda.jp

Sosuke Kato
Waseda University, Japan
sow@suou.waseda.jp

Tetsuya Sakai
Waseda University, Japan
tetsuyasakai@acm.org

Injae Lee
Naver Corporation, Korea
injae.lee84@navercorp.com

Kyungduk Kim
Naver Corporation, Korea
kyungduk.kim@navercorp.com

Inho Kang
Naver Corporation, Korea
tomf@cruise.com

ABSTRACT

In this paper, we present three models for the nugget detection and dialogue quality subtasks at the NTCIR-15 DialEval-1 task. Despite the recent progress in dialogue systems, we still face a number of unresolved challenges such as the dialogue system that often generates responses that cannot satisfy customers or responses that cannot help solve problems. Therefore, we submitted three models to the NTCIR-15 DialEval-1 task. The first model is run0: LSTM with attention-based dialog embedding, using a recurrent neural network with an attention layer to embed the previous dialogue context. The model used two representation vectors, an extracted dialogue context vector and a sentence vector of the target sentence. The second model is run1: transformer encoder architecture for English nugget detection. The third model is run2: BiLSTM with an attention layer that leverages the outputs of the BiLSTM to obtain a sentence-level representation. On the English dataset of nugget detection subtask, run0 model: LSTM with attention-based dialogue embedding outperforms the baseline, but on the Chinese dataset, it does not outperform the baseline. This suggests that attention-based dialog embedding is possibly helpful for a smaller English dataset.

TEAM NAME

RSLNV

SUBTASKS

Nugget Detection(Chinese, English), Dialogue Quality(Chinese, English)

1 INTRODUCTION

In recent years, several researchers have been working on the development of automated dialogue systems. However, the existing evaluation systems are dependent on manual labour, which is expensive and inefficient. Therefore, an automated evaluation system needs to be established to solve this problem.

To effectively and economically evaluate the dialogue system the NTCIR-15 DialEval-1 subtask [3] proposed automatically evaluating customer-helpdesk dialogues. There are two subtasks: (dialogue quality (DQ) subtask that aims to evaluate dialogue by quality score and (2) nugget detection (ND) subtask that aims to judge whether a turn of dialogue is a nugget. These subtasks are based on the Chinese and English datasets.

We present three models for ND and DQ. Our methods are trained and tested on the NQ and DQ subtasks of the NTCIR-15 DialEval-1 Task.

The remainder of this paper is organized as follows. Section 2 describes the background and baseline method, and Section 3 describes our run modules. Section 4 introduces the dataset used. Section 5 presents the evaluation metric and the official result. Finally, Section 6 provides concluding statements.

2 BACKGROUND

2.1 Dialog Context Embedding

In the DialEval-1 subtasks, we provided a dataset of Customer-Helpdesk dialogue. Accurately representing the context of dialogue may be the most important part of this task. There have been many successful studies [6] on word representation. These researchers have demonstrated that word embedding can well represent the semantic meaning behind words. However, the dataset comprises context-related dialogue text, which means that the feedback is originated from the context above. Compared with words, representing phrases and sentences may be more challenging and important. Therefore, in the semantic understanding of dialogue text, we must consider the use of sentence representation and text representation. There have been several studies involving recurrent neural networks (RNNs) with the aim of representing phrases and sentences [5]. They have reported that representing a sentence in a dialog is especially more challenging because considering the context coming from the previous sentences is necessary. Even with the same text, sentences may have different meanings depending on the context of the preceding sentences. Therefore, representing the overall context of the previous dialog has been an important part of sentence representation tasks.

Attention mechanisms have been proven to improve the ability of natural language understanding. There have been few studies that attempted to represent a sentence in the context of the dialog using the attention mechanism. Chen et al. [1].used two separate encoders for context and the target sentence, and they added the attention distribution to represent the memory from the contextual sentence encoder. They measured the attention distribution over history utterances and used it as a weight for each memory. Chan et al. [4] proposed an RNN and attention layer to embed dialogue and combined it with FFNN to detect dialog breakdown in multi-classification tasks. Their study showed that it significantly outperforms dialog breakdown detection by using the attention-based embedding model. Our run0 model follows the model presented in Chan et al. [4]. We used two different encoders for the sentence and the context and had one attention layer over RNNs to understand the dialogue context. Finally, we apply the LSTM-based baseline model as the input of attention layers after dialog embedding and nugget detection and evaluate dialogue quality.

2.2 Bi-LSTM Baseline

LSTM is an improved RNN model suitable for modeling temporal data such as textual data. Combining the representations of words into a sentence can be done by methods such as summing, i.e. adding all the representations of words, or averaging, but these methods do not consider the order of the words before and after in the sentence. For example, in the sentence, ‘I do not think he’s good’, the word ‘do not’ is the negation of the word ‘good’, indicating that the emotional polarity of the sentence is negative. Longer distance dependencies can be better captured using the LSTM model. This is because LSTM learns through the training process in which information is remembered and what information is forgotten.

However, there is a problem with modelling sentences using LSTM: encoding information from the back to the front is not possible. When classifying at a finer granularity, such as for the five-category task of strong-degree positive, weak-degree positive, neutral, weak-degree pejorative, and strong-degree pejorative, attention needs to be paid to the interaction between emotion words, degree words, and negation words.

BiLSTM was proposed by Graves et al. [2]. It is a combination of forward LSTM and backward LSTM. Using this structure, BiLSTM could solve the problem of LSTM’s inability to encode information from the back to the front. Figure 1 shows the structure of the BiLSTM. After forward LSTM inputs I_0, I_1, I_2, vector $[\vec{h_0}, \vec{h_1}, \vec{h_2}]$is obtained. After backward LSTMinputs I_0, I_1, I_2,vector $[\overleftarrow{h_0}, \overleftarrow{h_1}, \overleftarrow{h_2}]$is obtained. Then the forward and backward vectors are concatenated to be vector $[h_0, h_1, h_2]$.

In this subtask, the baseline model used three layers of BiLSTM to obtain contextual features and semantic information from the dialogue.

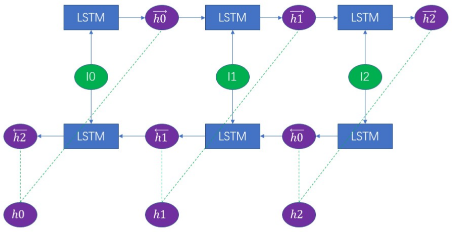

Figure 1: Bi-LSTM structure

3 RUN DESCRIPTION

In this section, we illustrate the proposed models. We first describe the overall framework structure and subsequently elaborate on the details of the model.

3.1 Run 0: LSTM with Attention-Based Dialog Embedding

3.1.1 Model Overview. Figure 2 shows that our run0 model mainly comprises three parts: sentence embedding, attention masking, and distribution regression. The architecture of our dialog context embedding model comprises two parts: one is sentenced embedding and the other is attention between sentences. We need to embed both the target system utterance and the dialog context and nugget labels because our model incorporates all of them to estimate the nugget and quality. We first embed every sentence in the dialog using the GloVe vector, which is the same as the baseline. With the sentence representations of the previous dialog, we extract one context vector using an attention layer to mask each sentence. Finally, we use an LSTM to map the resulting vectors for estimating the probability of nugget and evaluating dialogue quality.

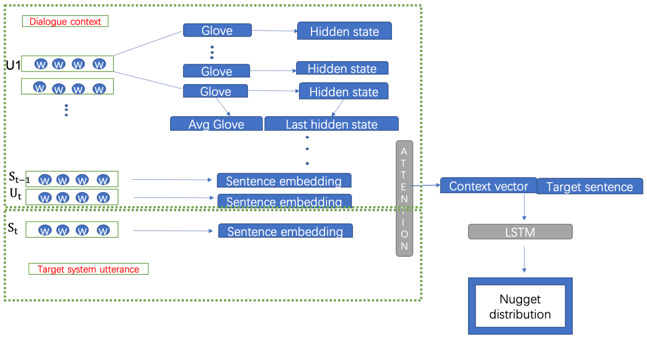

Figure 2: LSTM with attention-based dialog embedding

3.1.2 Attention-Based Dialogue Embedding. Our embedding of a sentence is the concatenation of two vectors: an average of the GloVe vector of all words in the sentence and the last RNN hidden state of each sentence. In the English language task, we used GloVe vectors of 100 dimensions trained by the Twitter data. In the Chinese language task, we used Baidu data. Subsequently, we put each word into the RNN encoder through the GRU activation function to generate the hidden state. After placing all words through the RNN, it outputs the last hidden state of the last word, which is considered to represent the context of the input sentence. After calculating the representations of each sentence from a dialog, we use an attention layer to extract a context vector for the entire dialogue. The attention layer outputs the weight of each sentence from past dialogues, and the context vector is computed as a weighted sum of the sentence embeddings.

3.1.3 LSTM Layer. The last layer is LSTM, which is the same as a baseline, to map all representations to the actual distribution of nugget labels and quality labels. For the DQ subtask, the last hidden state is used as the representation of the dialogues. Finally, dialogue representation is fed into dense layers to estimate the distributions of dialogue quality. For the ND subtask, it predicts the nugget distribution for customer turns and helpdesk turns.

3.2 Run 1:Transformer Encoder Architecture for English Nugget Detection

3.2.1 Model Overview. In this subtask, we propose a transformer model to improve the performance in the English detection subtask.

3.2.2 Word Embedding and Transformer Architecture. Word embedding: we loaded Glove 300-dimension pre-trained embedding files to embed the token in each turn to tensor and use the bag-of-word method to express all turns.

The transformer is an encoder-decoder architecture. The structure is shown in Figures 3 and 4. We only used transformer encoder parts that included six layers, and each layer contained a self-attention mechanism and feed-forward neural network. After loading pre-trained word embedding, we place the embedding matrix into transformer models and by passing a fully connected layer and Softmax layer to calculate the loss between label and prediction.

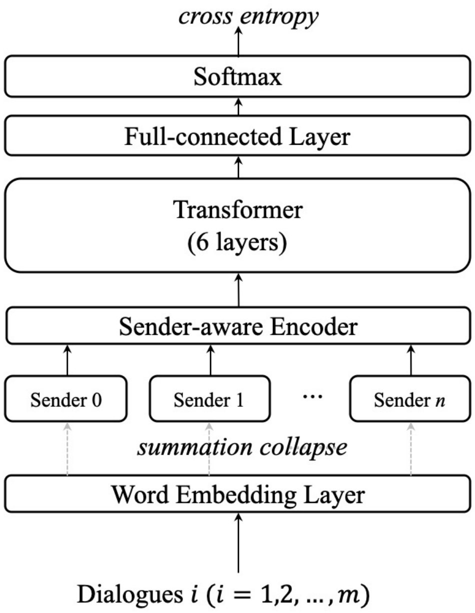

Figure 3: Architecture of the transformer encoder

3.3 Run 2: BiLSTM with Attention Layer

3.3.1 Attention mechanism. The attention mechanism has been used across a wide variety of natural language processing tasks. This approach mimics human attention, indicating that it can help the model learn which part of the text should be paid more attention to, resulting in more reasonable sentence representations. A crucial application of the attention mechanism is to use it with the LSTM model. It can be used to solve the problem of the difficulty in obtaining a final reasonable vector representation when the input sequence of the LSTM model is long.

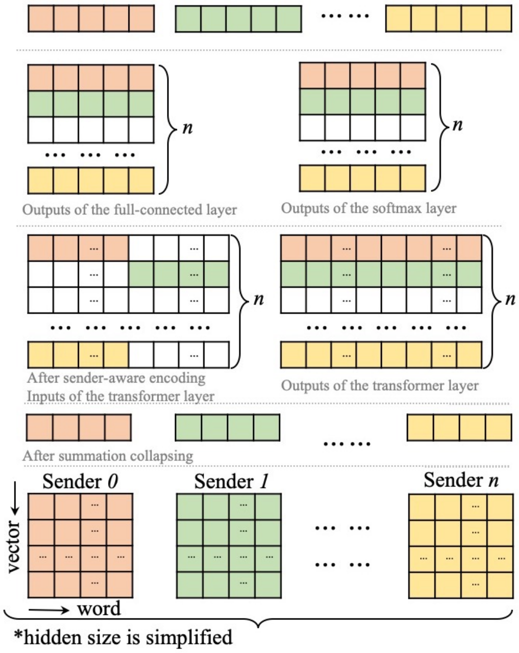
n
Outputs of the full-connected layer
Outputs of the softmax layer
After sender-aware encoding
Inputs of the transformer layer
Outputs of the transformer layer
After summation collapsing
Sender 0
Sender 1
Sender n
vector
word
*hidden size is simplified

Figure 4: Transformer the encoder architecture for English nugget detection

3.3.2 Model Overview. The structure of run2 is shown in Figure 5. The embedding part was the same as the baseline. The Bi-LSTM with attention layer parts is based on the work of Zhou et al. [7]. We use the hidden outputs of the BiLSTM as the feature vector. Subsequently, we take these feature vectors as input and compute the weight vectors in the attention layer as follows:

$$U = \tanh(H) \tag{1}$$

$$\alpha = softmax(W^T U) \tag{2}$$

$$r = H\alpha^T \tag{3}$$

where H is a matrix of the output vectors produced by the BiLSTM. In Eq. (2), W is the weight vector. The representation r in Eq. (3) is the output vector of this attention model. Then, we use this output to detect the distribution in the nugget distribution part.

4 EVALUATION AND ANALYSIS

4.1 Official Result of Submitted Runs

According to the official results [3] on the English and Chinese datasets, the results of the run we submitted have no significant improvement over the baseline. Only in the English ND task, our model improved compared to the baseline.

In the DialEval-1 task, our run0 submitted the result to the tasks of Chinese, English ND, and DQ subtasks. Among them, only run0 ranks second among all teams in the ND(English) task, which is little improvement of the average score over the baseline, but not a significant improvement. As we attempt to apply fine-tuning to GloVe before averaging for sentence embedding in the English task,

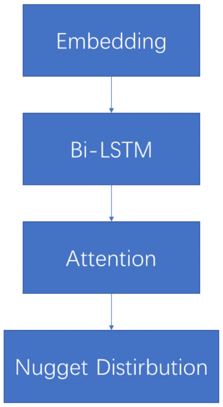

Figure 5: Structure of run2

fine-tuning can improve embeddings; in English ND, it did not outperform the baseline on the mean score, which may be because the hyperparameters of this model did not adjust well when the number of classes increased. In Chinese tasks, the average score of run0 is not as good as that of the baseline. We used the Baidu dataset and applied Bag of Words to obtain word vectors that did not improve the word embeddings, and the attention layer between sentences did not capture the feature well in Chinese language sentences. Throughout the experiments, we observed that the change in the sentence embedding method affects the overall performance of the model, and fine-tuning can improve embedding. Using the attention layer to present sentence embedding may sometimes result in different performances in the different language models.

Run1 was submitted to English nugget detection. Our run1 model used a dialogue evaluation dataset to train our transformer encoder models, and the result is not as good as we expect. In our experiment, we proved that in the small dataset, the average score of the transformer cannot outperform a simple model such as Bi-LSTM.

Run2 was submitted to the Chinese nugget detection task. The mean score of run2 is worse than that of Bi-LSTM.

We show the official evaluation scores of our runs and the Bi-LSTM baseline model and underline the top scores in Tables 1–8.

Table 1: Chinese Dialogue Quality (A-score) results

Run	Mean RSNOD	Mean NMD
Run0	0.2345	0.1606
Run1	N/A	N/A
Run2	N/A	N/A
BL-lstm	0.2305	0.1598

Table 2: Chinese Dialogue Quality(S-score) Results

Run	Mean RSNOD	Mean NMD
Run0	0.2141	0.1483
Run1	N/A	N/A
Run2	N/A	N/A
BL-lstm	0.2088	0.1455

Table 3: Chinese Dialogue Quality(E-score) Results

Run	Mean RSNOD	Mean NMD
Run0	0.1811	0.1393
Run1	N/A	N/A
Run2	N/A	N/A
BL-lstm	0.1782	0.1386

Table 4: Chinese Nugget Detection Results

Run	Mean JSD	Mean RNSS
Run0	0.0746	0.1749
Run1	N/A	N/A
Run2	0.0768	0.1760
BL-lstm	0.0709	0.1673

Tables 1–4 show the mean evaluation scores for the DQ (Chinese) subtask in terms of A-score, S-score, and E-score, and Table 4 shows the mean evaluation scores for the ND (Chinese) subtask.

Table 5: English Dialogue Quality(A-score) Results

Run	Mean RSNOD	Mean NMD
Run0	0.2311	0.1603
Run1	N/A	N/A
Run2	N/A	N/A
BL-lstm	0.2271	0.1591

Table 6: English Dialogue Quality(S-score) Results

Run	Mean RSNOD	Mean NMD
Run0	0.2169	0.1454
Run1	N/A	N/A
Run2	N/A	N/A
BL-lstm	0.2111	0.1413

Table 7: English Dialogue Quality(E-score) Results

Run	Mean RSNOD	Mean NMD
Run0	0.1789	0.1354
Run1	N/A	N/A
Run2	N/A	N/A
BL-lstm	0.1687	0.1248

Table 8: English Nugget Detection Results

Run	Mean JSD	Mean RNSS
Run0	0.0743	0.1753
Run1	0.0989	0.2142
Run2	N/A	N/A
BL-lstm	0.0762	0.1781

Tables 5–8 show the mean evaluation scores for the DQ subtask (English) in terms of A-score, S-score, and E-score, and Table 8 shows the mean evaluation scores for the ND (English) subtask.

4.2 Confusion Matrix of Run2

To investigate the hidden information behind the mean scores, we also created a confusion matrix for run2. As run2 submits the result of the nugget detection task, we separately create confusion matrices for the customer and the helpdesk to observe the predictive accuracy and distribution for each type. The results are shown in Figures 6 and 7. We can observe that the accuracy of predicting 'HNUG' is not particularly high. Models incorrectly predict 'HNUG*' and 'HNAN' as 'HNUG' in many conversations. The lower mean score may possibly have been caused by this part.

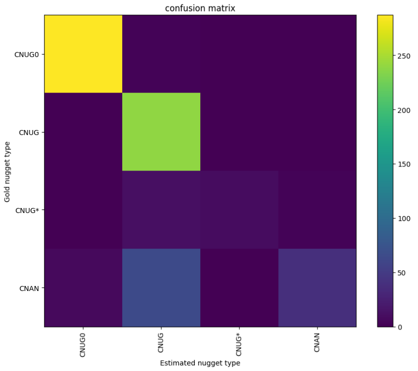

Figure 6: Confusion matrix of customer

5 CONCLUSION

In this paper, we proposed three models to the subtasks dialogue quality (DQ) and nugget detection (ND) on NTCIR-15 DialEval-1, using attention and transform to modify its baseline in embedding and neural network structures. The experimental results suggest that our model does not perform well on the Chinese language subtask. However, in the English ND subtask, our model slightly outperformed the baseline, although it was not statistically significant. The run1 model uses transformer encoder models, and the result is not as good as we expected. For the run2 model, we used Bi-LSTM with an attention layer. However, the performance was worse than the baseline. The results suggest that on small datasets, a simple machine learning method such as Bi-LSTM has relatively good performance. We believe that a better embedding method can make the models more effective; therefore, we will conduct more experiments using the embedding of the pre-trained models in the future.

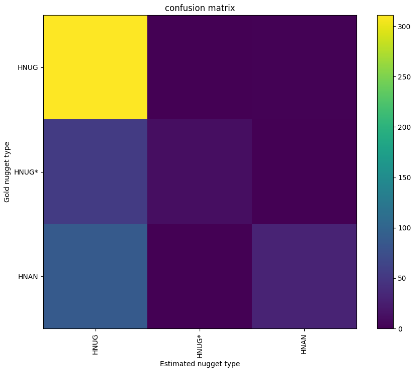

Figure 7: Confusion matrix of helpdesk

ACKNOWLEDGEMENT

We thank the DialEval-1 participants and the NTCIR chairs for making this task happen.

REFERENCES

[1] Yun-Nung Chen, Dilek Hakkani-Tür, Gökhan Tür, Jianfeng Gao, and Li Deng. 2016. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding.. In *Interspeech*. 3245–3249.

[2] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.

[3] R. Kato, S.and Suzuki, Z.and Sakai Zeng, and I T., Kang. 2020. Overview of the NTCIR-15 Mission Impossible Task. In *Proceedings of the NTCIR-15 Conference*.

[4] Chanyoung Park, Kyungduk Kim, and Songkuk Kim. 2017. Attention-based dialog embedding for dialog breakdown detection. In *Proceedings of the dialog system technology challenges workshop (DSTC6)*.

[5] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[6] Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, 384–394. https://www.aclweb.org/anthology/P10-1040

[7] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 207–212.