SLSTC at the NTCIR-14 STC-3 Dialogue Quality and Nugget Detection Subtasks

Sosuke Kato[1], Rikiya Suzuki[1], Zhaohao Zeng[1], and Tetsuya Sakai[1]

Waseda University, Tokyo, Japan
sow@suou.waseda.jp, rikiyasuzuki@ruri.waseda.jp,
zhaohao@fuji.waseda.jp, tetsuyasakai@acm.org

Abstract. This paper describes our approaches to the Nugget Detection and Dialogue quality subtasks at the NTCIR-14 STC3 task. We tried to make a few changes to the baseline BiLSTM model, and submitted three models, including BiLSTM with multi-head attention, BiLSTM with multi-task learning, and BiLSTM with BERT. On the Chinese dataset, BiLSTM with multi-task learning and BiLSTM with BERT outperformed the baseline, but the improvement is not statistically significant. On the smaller English dataset, the multi-task learning model is the best of our submitted runs, but it does not outperform the BiLSTM baseline in both ND and DQ subtasks. Also, with BERT the baseline model also performs better than the baseline on the English dataset, which may suggest that multi-task learning and pre-trained embedding are helpful on the smaller English dataset.

Team Name

SLSTC

Subtasks

Dialogue Quality, Nugget Detection

1 Introduction

Due to recent advances in AI, increasing researchers are working on task-oriented dialogue agents. However, there are very few approaches to evaluating such systems, and most of existing ones are expensive and even inefficient, such as hiring human to read the dialogues and judge or asking users to provide feedback. In this paper, we attempt to utilise machine learning methods to evaluate task-oriented dialogues automatically.

We proposed three models to read textual dialogues and make judgements on them. Our methods are trained and tested on Nugget Detection (NQ) and Dialogue Quality (DQ) subtask of NTCIR14 Short Text Conversation Task [7]. The remainder of this paper details our models and the experimental results in this task.

2 Background

In the STC3 task, the data are customer-helpdesk dialogues, and there are two kinds of judgements the dataset provide: quality and nugget, where dialogue quality is three 5-point Likert scale scores to measure (1) A-score: task accomplishment (2) S-score: customer satisfaction (3) E-score: dialogue efficiency; and nugget detection is to classify each dialogue turn if there are useful to solve the problem. The Chinese dataset of STC3 ND and DQ subtasks contains 3,700 annotated Chinese dialogues for training and 390 dialogues for testing. Its English version is built by manually translating a subset (1,672 for training and 390 for testing) of the Chinese dataset.

Suppose that we are given a Customer-Helpdesk dialogue $\mathbb{D} = (\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^T)$, where $\mathbf{p}^t$ is a dialogue turn uttered by $s^t \in \{C, H\}$ as we only have two possible speakers: Customer (C) and Helpdesk (H). Each turn consists of a list of tokens $\mathbf{p}^t = (\mathbf{x}_1^t, \mathbf{x}_2^t, ..., \mathbf{x}_{m_t}^t)$, where m_t denotes the length of the t-th turn. We use one-hot encoding to represent the i-th token of the t-th turn $\mathbf{x}_i^t \in \mathbb{R}^{V \times 1}$ and V is the vocabulary size. For each dialogue, the system predicts their dialogue quality labels $(+2, +1, 0, -1, -2)$ in the dialogue quality subtask. For nugget detection, the system performs classification on each dialogue turn. There are four labels for a customer turn: Task trigger, regular nugget, goal nugget and not-a-nugget, and three labels for a helpdesk turn: regular nugget, goal nugget and not-a-nugget.

2.1 LSTM Baseline

STC3 organisers provided a LSTM baseline model[1] for both DQ and ND subtasks. In this model, each token is converted into a dense vector using an pretrained embedding matrix weight[2] $\mathbf{A} \in \mathbb{R}^{d \times V}$. To differentiate between the customer's turn and the helpdesk's turn, it converts $\mathbf{x}_i^{t\prime}$ into $\tilde{\mathbf{x}}_i^t \in \mathbb{R}^{2d \times 1}$ as: $\tilde{\mathbf{x}}_i^t = [\mathbf{x}_i^{t\prime} \cdot \mathbb{1}_{\{s^t = H\}}, \mathbf{x}_i^{t\prime} \cdot \mathbb{1}_{\{s^t = C\}}]$ where $[\cdot, \cdot]$ denotes concatenation and $\mathbb{1}$ denotes indicator function. For each turn, Bag of Words model is used to encode it into $\mathbf{e}^t$ by summing the word embedding vectors to obtain its representation $\mathbf{e}^t$: $\mathbf{e}^t = \sum_{i=1}^{m_t} \tilde{\mathbf{x}}_i^t$. Then, Bidirectional Long Short-Term Memory (BiLSTM) [3] is employed to map each post encoding $\mathbf{e}^t$ to a vector $\mathbf{h}^t$. For simplicity, we use LSTM to represent the calculation at each time step as $\mathbf{h}^t = \mathrm{LSTM}(\mathbf{h}^{t-1}, \mathbf{e}^t)$.

For the DQ subtask, the last hidden state is used as the representation of the dialogues, and then fed into a dropout layer before a fully connected layer: $\hat{y}_k = \mathbf{W}^s \mathrm{Dropout}(\mathbf{h}^T) + \mathbf{b}^s$ where $\hat{y}_k$ is the predicted score of the k-th example. We consider predicting the truth score y_k a classification problem, so Cross Entropy is used as the loss function to train the model. Since there are three scores in DQ, the model has three output layers that share the same LSTM encoder. For the ND subtask, the hidden state of each turn is used for classification and Cross Entropy is also used as the loss function.

[1] `https://github.com/sakai-lab/stc3-baseline`

[2] GloVe-840B is used for the English dataset, and a Chinese Word2Vec [5] embedding pre-trained by Baidu is used for the Chinese dataset.

3 Run Description

3.1 Run 0: BiLSTM with Multi-Head Attention

Attention mechanisms have been proved to improve the ability of natural language understanding. Transformer [6] is a deep learning model that only adopts attention mechanisms for translation tasks without a recurrent or convolutional network. Positional Encoding in Transformer is an alternative approach to RNN for capturing time-series features from the input. However, it is not appropriate in the specific environment which does not have a large amount of data. Hence, we apply BiLSTM like baseline model as the input of attention layers after word embedding and bag of words instead of Positional Encoding.

In this run, we adopt two attention mechanisms, Self-Attention and Score-Attention. Self-Attention is used to analyze the input, on the other hands Score-Attention is used to capture the individual features for each score in the DQ subtask. Self-Attention includes two parts, Scaled Dot-Product Attention and Multi-Head Attention. In Scaled Dot-Product Attention, the input consists of queries, keys and values of dimension d_v. Firstly, we compute the dot products of the query with all keys. Secondly, we divide each dot product by $\sqrt{d_v}$ for scaling, then apply a softmax function to obtain the weights matrix. Finally, all values are multiplied by the weights matrix. Each query is packed together into a matrix Q to compute the dot products simultaneously. The keys and values are also packed together into matrices K and V.

$$Attention(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_v}})V \tag{1}$$

Multi-Head Attention using single attention function h times with different enables the model to attend to information from different representation subspaces at different positions. Therefore, we utilize Scaled Dot-Product Attention after dividing the dimensionality d_{model} of input by the number of head h. We employ $h = 8$ parallel attention heads and $d_v = d_{model}/h = 32$. The output of Self-Attention is added to the input as shortcut connection [2]. After the connection, we adopt Position-wise Feed-Forward Networks (FFN). This consists of two linear transformations with a ReLU activation.

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{2}$$

The dimensionality of input and output is $d_{model} = 256$, and the inner-layer has dimensionality $d_{ff} = 1024$.

In the DQ subtask, the model must output the distribution of each score separately, so we apply Score Attention. This attention function includes multiple linear or non-linear functions for mapping the input to the subspaces of score or scale class.

$$ScoreAttention(x) = \mathrm{softmax}(\tanh{(xW_1 + b)}W_2)^T xW_3 \tag{3}$$

We employ $d_{score} = 3$ and $d_{scale} = 5$. In this case, where the projections are the parameter matrices $W_1 \in \mathbb{R}^{d_{model} \times d_v}$, $W_2 \in \mathbb{R}^{d_v \times d_{score}}$, $W_3 \in \mathbb{R}^{d_{model} \times d_{scale}}$, and the parameter vector $b \in \mathbb{R}^{d_v}$.

3.2 Run 1: BiLSTM with multi-task learning

We applied the two following changes to the baseline model mentioned in section 2.1,

- a loss based on comparing adjacent probability bins;
- parameters shared for both subtasks.

A loss based on comparing adjacent probability bins First, we devised a loss which considers ordinal probability bins as follows;

$$L_{\text{diff}} = \frac{1}{B-1} \sum_{i}^{B-1} \left\{ (\hat{y}(i+1) - \hat{y}(i)) - (y(i+1) - y(i)) \right\}^2 , \tag{4}$$

where $y(i)$ and $\hat{y}(i)$ denotes the gold and predicted probability of a label i, and B denotes the number of label types, i.e., $5 = |\{-2, -1, 0, 1, 2\}|$, then we use the sum of the loss used in the baseline model and the above loss.

Parameters shared for both subtasks Next, we devised parameters that are shared between multiple tasks referencing [4]. Similarily, we tried to share parameters of part of LSTMs. However, for the DQ and ND subtasks, a model that shares all parameters of LSTMs obtained better scores than a model that shares parameters of part of LSTMs with validation dataset. Therefore, the model used in Run 1 shared all parameters of LSTMs for ND and DQ subtasks and we trained it by multitask learning.

3.3 Run 2: BiLSTM Baseline with BERT Embedding

Instead of GloVe or Word2Vec embedding, we attempt to utilise Bidirectional Encoder Representations from Transformers (BERT) [1] as the embedding layer of the LSTM baseline. Different from the original BERT, we simply use the top four layers of it as a feature extractor, and we do not fine-tune it during training because our cross-validation results suggest that fine-tuning may not provide any improvement in this task.

4 Official Results and Discussions

According to the official results, the improvements of the submitted runs are not statistically significant compared to the LSTM baseline in both Chinese and English dataset. In addition to the mean evaluation scores, we count how many times each run obtained the best evaluation score among all STC runs for one dialogue, and we denote it by BC (Best Count) in this paper. In the ND subtask, the evaluation score is calculated for each turn, therefore, we use the final evaluation score described in 3.2 subsection of [7], i.e., $0.5S_C + 0.5S_H$ where S_C and S_H denotes the average measure score for customers turns and helpdesks turns respectively. We show the official evaluation scores and BCs of our runs and the LSTM baseline model and underline the top scores in Tables 1 to 8.

On the Chinese dataset, all our submitted runs outperform the baseline in both ND and DQ subtasks, while none of the improvements is statistically significant. BiLSTM with Multi-Head Attention (Run 0) obtains the worst results in all model of SLSTC. However, BCs are relatively larger than others (e.g. Run1 for DQ subtask), which may because that the hyper-parameters of this model were densely tuned on a subset of test data so the model may overfit the dataset. BiLSTM with Multi-Task Learning (Run 1) obtains the best results in terms of E-score and S-score but BCs are not large. It appears that Run 1 tends to perform well for dialogues that are hard for others to predict distributions.

Table 1. Chinese Dialogue Quality (A-score) Results and Best Counts

Run	Mean RSNOD	BC(RSNOD)	Mean NMD	BC(NMD)
SLSTC-run0	0.1306	68	0.0831	69
SLSTC-run1	0.1235	60	0.0819	50
SLSTC-run2	0.1249	59	0.0843	60
BL-lstm	0.1263	32	0.0863	35

Table 2. Chinese Dialogue Quality (S-score) Results and Best Counts

Run	Mean RSNOD	BC(RSNOD)	Mean NMD	BC(NMD)
SLSTC-run0	0.1290	52	0.0787	58
SLSTC-run1	0.1243	46	0.0772	38
SLSTC-run2	0.1175	69	0.0731	68
BL-lstm	0.1245	37	0.0800	33

Table 3. Chinese Dialogue Quality (E-score) Results and Best Counts

Run	Mean RSNOD	BC(RSNOD)	Mean NMD	BC(NMD)
SLSTC-run0	0.1238	58	0.0790	60
SLSTC-run1	0.1159	49	0.0754	47
SLSTC-run2	0.1178	68	0.0779	61
BL-lstm	0.1182	24	0.0794	30

Table 4. Chinese Nugget Detection Results and Best Counts

Run	Mean JSD	BC(JSD)	Mean RNSS	BC(RNSS)
SLSTC-run0	0.0241	58	0.0946	50
SLSTC-run1	0.0225	62	0.0913	54
SLSTC-run2	0.0217	78	0.0876	78
BL-lstm	0.0220	45	0.0899	47

BiLSTM with BERT (Run 2) achieves the largest BCs in most of the results, but BCs of the LSTM baseline more than a half of ones of Run 2. To improve the results of our runs more, we need to investigate what kind of dialogues are hard to predict by each of our runs.

Table 5. English Dialogue Quality (A-score) Results and Best Counts

Run	Mean RSNOD	BC(RSNOD)	Mean NMD	BC(NMD)
SLSTC-run0	0.1493	50	0.1017	42
SLSTC-run1	0.1391	58	0.0908	62
SLSTC-run2	0.1370	54	0.0933	59
BL-lstm	0.1320	55	0.0896	52

Table 6. English Dialogue Quality (S-score) Results and Best Counts

Run	Mean RSNOD	BC(RSNOD)	Mean NMD	BC(NMD)
SLSTC-run0	0.1423	45	0.0907	59
SLSTC-run1	0.1340	53	0.0820	47
SLSTC-run2	0.1306	68	0.0822	67
BL-lstm	0.1310	36	0.0838	40

Table 7. English Dialogue Quality (E-score) Results and Best Counts

Run	Mean RSNOD	BC(RSNOD)	Mean NMD	BC(NMD)
SLSTC-run0	0.1404	59	0.0938	45
SLSTC-run1	0.1321	68	0.0859	66
SLSTC-run2	0.1219	76	0.0828	88
BL-lstm	0.1220	49	0.0824	45

Table 8. English Nugget Detection Results and Best Counts

Run	Mean JSD	BC(JSD)	Mean RNSS	BC(RNSS)
SLSTC-run0	0.0289	58	0.1037	50
SLSTC-run1	0.0252	62	0.0973	54
SLSTC-run2	0.0263	78	0.0979	78
BL-lstm	0.0248	45	0.0952	47

On the English dataset, most participant runs do not outperform the baseline BiLSTM system, which may be because the English dataset is much smaller than the Chinese one. However, BiLSTM with Multi-Task Learning (Run 1) and BiLSTM with BERT (Run 2) are the only two runs which are better than the baseline in some cases (e.g., S-score and E-Score), which suggests that pre-trained BERT embedding and multi-tasking learning are helpful for small training data.

5 Conclusions and Future Work

We participated in the STC-3 DQ and ND subtasks and submitted three runs, i.e., BiLSTM with Multi-Head Attention (Run 0), BiLSTM with Multi-Task Learning (Run 1) and BiLSTM with BERT (Run 2). On the Chinese dataset, our runs outperformed the baseline but not statistically significantly. On the English dataset, Run 1 and Run 2 were better than the baseline in some cases. While our preliminary analysis suggests the pre-trained BERT embedding and multi-task learning are helpful for both Chinese and English datasets, more detailed analyses are required.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2016)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
4. Liu, P., Qiu, X., Huang, X.: Adversarial multi-task learning for text classification. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1–10 (2017)
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is All you Need. In: Advances in Neural Information Processing Systems 30, pp. 5998–6008 (2017)
7. Zeng, Z., Kato, S., Sakai, T.: Overview of the NTCIR-14 Short Text Conversation Task: Dialogue Quality and Nugget Detection Subtasks. In: Proceedings of The NTCIR-14 Conference (2019)