SKYMN at the NTCIR-15 DialEval-1 Task

Junjie Wang
Waseda University, Japan
wjj1020181822@toki.waseda.jp

Yuxiang Zhang
Waseda University, Japan
joel0495@asagi.waseda.jp

Tetsuya Sakai
Waseda University, Japan
tetsuyasakai@acm.org

Hayato Yamana
Waseda University, Japan
yamana@yama.info.waseda.ac.jp

ABSTRACT

We participated in the English Dialogue Quality subtask of the NTCIR-15 DialEval-1 task. We implemented deep learning models (a convolutional neural network and bi-directional long short-term memory) and the pre-trained models (ALBERT and DistilRoBERTa), and for this task, we proposed a label-based training method to transform the problem from a special multi-label classification task into a multi-class classification task. Based on our results, the label-based training method improves the performance of DistilRoBERTa model. For the distribution-based training method, a deep learning model with bi-directional long short-term memory, bi-directional gate recurrent unit and convolutional layer outperform other models.

KEYWORDS

Dialogue system evaluation, English, deep learning, Distribution-based training, Label-based training

TEAM NAME

SKYMN

SUBTASKS

Dialogue Quality (English)

1 INTRODUCTION

When a company tries to sell a product to its users, despite the quality of the product itself, high-quality after-sales and customer services are essential. A good customer service system requires a targeted problem-solving ability and provides 24-h service. A higher sales volume represents higher hiring and training costs for the helpdesk, suggesting the importance of a high-quality intelligent helpdesk agent system. Evaluating the dialogue between the helpdesk and users of a real-world product is not only helpful for improving and judging the helpdesk quality, but also for training a reliable automatic evaluation method for an intelligent helpdesk system.

We participated in the English Dialogue Quality subtask of the NTCIR-15 DialEval-1 task. Our DistilRoBERTa model with a label-based training method achieved the highest performance among our models.

The rest of this paper is organized as follows. Related studies are described in Section 2, followed by our approaches in Section 3. Section 4 shows the experiments: In section 4.1, two different training methods we used are proposed, the evaluation metrics are explained in Section 4.2, and all the results are shown in section 4.3. Finally, we provide some concluding remarks in Section 5.

2 RELATED WORK

At NTCIR-14, some studies were conducted on building new word representations and new model architectures. Team WUST applied an attention layer after the LSTM baseline model [19]. Team CUIS proposes a two-stage method to obtain turn-level representation [6]. They utilized BERT to obtain sentence-level representations and then applied hierarchical attention networks to obtain turn- and conversation-level representations. Team WIDM applied a hierarchical CNN structure to encode sentences and then implemented LSTM and CNN models [4]. Furthermore, they applied a memory layer to capture the long-term information. Team SLSTC proposed three methods based on BiLSTM [9]. They utilized a transformer to encode the dialogue and built a BiLSTM with multi-head attention. They made a few attempts to replace the embedding layer of the LSTM baseline model with BERT. Moreover, they proposed a multi-task learning model to solve the dialogue quality and nugget detection tasks.

3 OUR APPROACHES

3.1 Word Embedding

We used Global Vectors for Word Representation (GloVe) as a word embedding method. We applied the GloVe dictionary with each word represented by a 50-dimension vector provided by Stanford University [14]. The word vectors were trained based on a combination of Gigaword5 [1] and Wikipedia2014 [2] corpora, which has 6 billion tokens in total.

In our experiments, because the lengths of each sentence are different, we pad all sentences into a uniformed length (we chose a length of 256 for the experiments using a CNN and a recurrent neural network (RNN)). In addition, we only keep the top 20,000 words (covers almost all words in English) in frequency of occurrence in the text as input. We then convert the words into word vectors. Words that do not exist in pre-trained word vectors dictionary but exist in dialogue texts of the dataset will be represented by normally distributed random vectors. Then, for the words that do exist, they are encoded as vectors of the pre-trained GloVe.

3.2 Deep learning model

3.2.1 CNN model. Our Convolutional Neural Network (CNN) model architecture is shown in Figure 1. Our CNN model architecture is modified based on Kim's model [10]. First, we encode the words of the dialogue text. We then convert the words into word vectors.

After applying the convolutional layer and the pooling layer to capture the local correlation features of the text, we use two dense layers to reduce the dimensionality to 15 for the output layer.

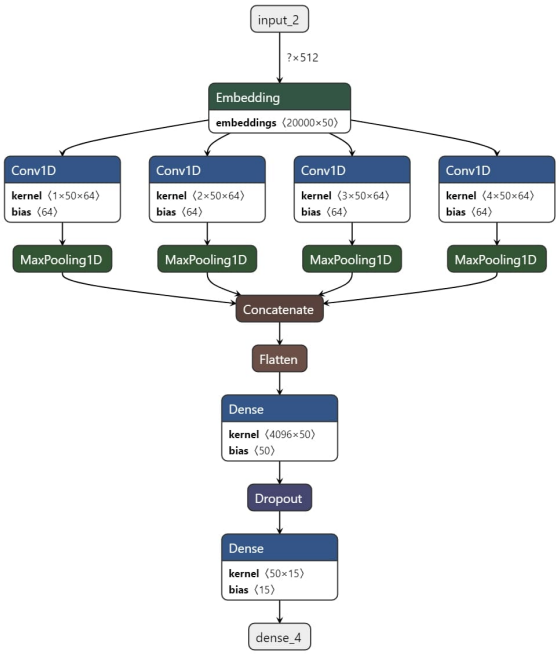

Figure 1: CNN model architecture

3.2.2 BiLSTM with attention model. Our model comprises a bi-directional long short-term memory (Bi-LSTM) network and an attention mechanism layer, as shown in Figure 2.

The most widely used RNN model is the long short-term memory (LSTM) model. This model is generally better at expressing long-term and short-term dependencies than vanilla RNNs. The structure of the LSTM is shown in Figure 3 [13].

The first step of LSTM is to decide what information should be kept or thrown away. The forget gate uses a sigmoid function for the computations. Information from the previous hidden state and information from the current state are computed using the following formula:

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

The LSTM then needs to input new information from this cell to update the cell state. The input gate handles this work by using a sigmoid function and a tanh function:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$T_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * T_t$$

After these operations, the output gate decides what the next hidden state should be. The output is obtained using the following:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)\ h_t = o_t * \tanh(C_t)$$

The new cell state and the new hidden states are computed in the next step using the same operations.

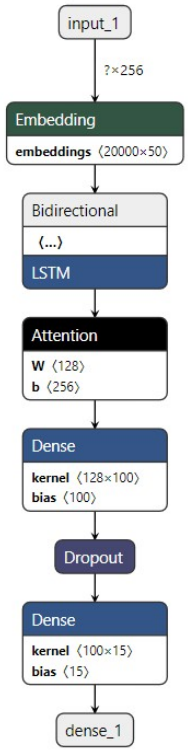

Figure 2: BiLSTM with Attention model architecture

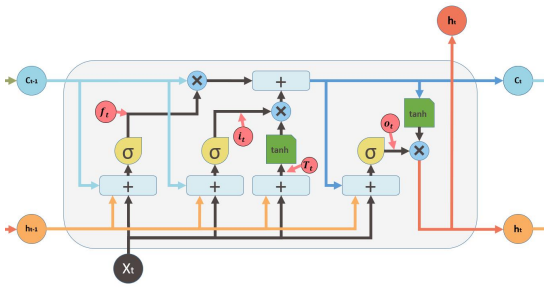

Figure 3: LSTM structure

The bidirectional LSTM structure is shown in Figure 4. Cell vector A participates in the forward calculation and cell vector A' is used in the backward calculation. So, the final output vector y depends on A and A'.

Because Bi-LSTM does not reflect the different importance of each epoch output information, we add an attention layer below. The attention layer produces a weight vector and merges word-level features into sentence-level features by multiplying the weight vector with the encoded token. We implemented the attention layer as described in Zhou *et al.* [21].

3.2.3 LGC model. The model architecture for our LSTM+GRU+CNN (LGC) model is shown in Figure 5. As a type of RNN such as LSTM,

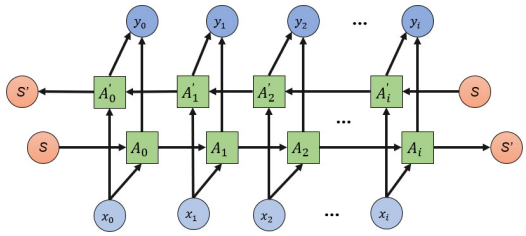

Figure 4: Bi-LSTM model structure

a GRU is also proposed to solve problems such as long-term memory and gradient vanishing during back propagation. A GRU combines the forget gate and input gate into a single "update gate." It also merges the cell state and hidden state, and makes some other changes, which makes it simpler than the standard LSTM model [5]. GRU and LSTM have almost the same actual performance in many cases. Compared with LSTM, GRU has one less "gating" inside. It has fewer parameters than LSTM, which is directly related to computing power and time cost. GRU also performs better on a small training dataset than LSTM [8]; thus, we add an extra GRU layer and expect it to perform better than the simple stacking of three LSTM layers because the original training dataset is not very large.

For the CNN part, we use a simple one-dimensional convolution layer to create a convolution kernel, and concatenate the outputs with an average pooling layer, following two dense layers to obtain the output layer in 15 dimensionalities.

3.3 Pre-trained model

A language model captures many language-related features for downstream tasks, such as long-term dependencies, hierarchical relationships, and emotional semantics. Compared with unsupervised learning tasks such as auto-encoders, pre-trained language models can perform well on tasks even with a small amount of training data. As the pre-training method of the language model, large-scale data are used with an unsupervised method. After achieving the pre-trained model, we utilize it as a feature extractor (like an embedding layer) or fine-tune it on specific tasks. Because some studies have confirmed that fine-tuning is slightly better than feature extraction [15], we chose the fine-tuning method in our experiment.

One of the most widely used pre-trained model is Pre-training of Deep Bidirectional Transformers for Language Understanding (BERT). Using masked language modeling and the next sentence prediction, BERT captures the word and sentence level representations, respectively [7]. A number of studies on improving BERT have been proposed, such as ALBERT [11], distilBERT [18] and DistilRoBERTa [3].

3.3.1 ALBERT model. A Lite BERT for Self-supervised Learning of Language Representations (ALBERT) offers an alternative for reducing parameters without massive performance loss. By utilizing factorized embedding parameterization, cross-layer parameter sharing and sentence order prediction, ALBERT$_{base}$ achieves similar performance to BERT$_{base}$ with only 89% parameters of BERT$_{base}$. For multi-sentence inputs, it utilizes a self-supervised loss that focuses on inter-sentence coherence.

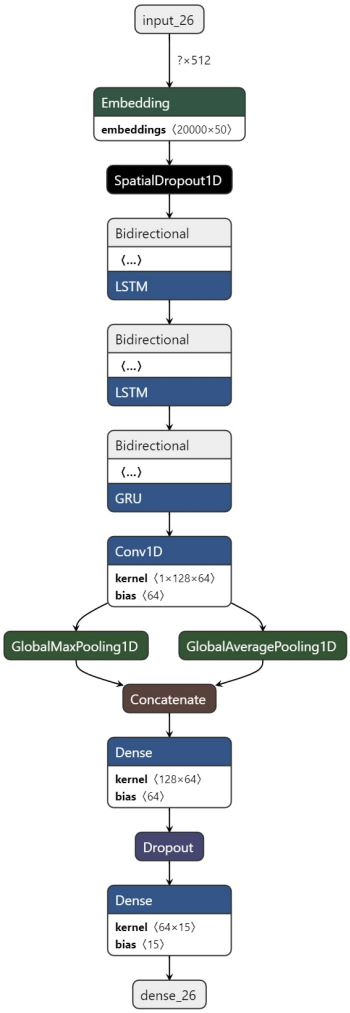

Figure 5: LGC model structure

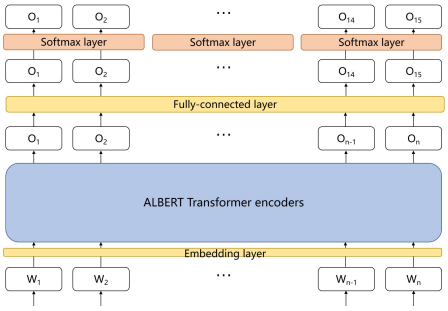

Figure 6: Fine-tuned ALBERT model structure

Figure 6 shows our fine-tuned ALBERT model architecture. We have obtained the open-sourced ALBERT pre-trained model (12 repeating layers, 128 embedding, 768-hidden, 12-heads, 11M parameters ALBERT base model) and the corresponding network

structure [1]. We take the training data in our experiments and carry out model training directly on this network to fine-tune the network parameters obtained in the pre-training step. After the output, we add three softmax layers to conduct the distributions of three labels.

3.3.2 DistilRoBERTa model. A Robustly Optimized BERT Pretraining Approach (RoBERTa) [12] is an improved BERT model by training more epochs than the BERT model with 10 times more data. A byte-level BPE vocabulary instead of the character-level vocabulary is used for encoding. Unlike BERT, RoBERTa is not trained for the next sentence prediction task and utilizes dynamic masking instead of static masking.

DistilRoBERTa is a distilled version of RoBERTa, with a 35% size reduction while achieving 95% of RoBERTa's performance on GLUE. Using the distillation loss and cosine similarity methods from DistilBERT [18], Huggingface implements DistilRoBERTa and releases the model on GitHub [2]. We obtained the open-sourced pre-trained DistilRoBERTa model (6 layers, 128 embedding, 768-hidden, 12-heads, 82M parameters) and the corresponding network structure from GitHub.

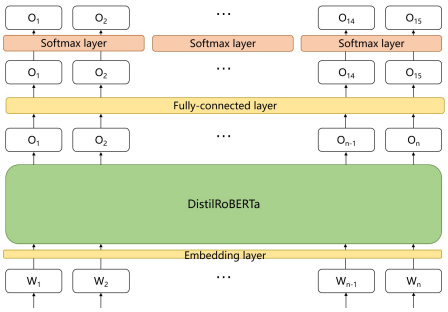

Figure 7: Fine-tuned DistilRoBERTa model structure

In this task, we used two training methods for the DistilRoBERTa model. Similar to ALBERT model, Figure 7 shows our fine-tuned DistilRoBERTa model architecture for distribution-based training in section 4.1.1. The rest of the methods are described in the Section 4.1.2.

4 EXPERIMENTS

4.1 Training methods

4.1.1 Distribution-based training. For a dialogue, an annotator chooses a label from [2, 1, 0, -1, -2] for three quality scores. Therefore, there are 15 labels for each annotation in the dialogue. We merged all A, S, and E scores from 19 annotators into a label probability distribution using the arithmetical mean. The DQ subtask can be tackled as a special multi-label text classification task. As shown in Figure 8, the model takes the label distribution as one of the inputs and results in a distribution.

[1] https://github.com/google-research/albert
[2] https://github.com/huggingface/transformers/tree/master/examples/distillation

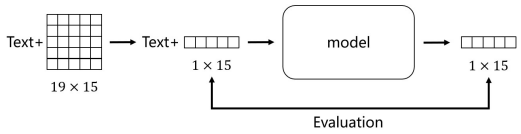

Figure 8: A dialogue input for distribution-based training

4.1.2 Label-based training. We propose a label-based training method for training the models. In detail, we built a model for each annotator using each quality score. We therefore built 57 models (3×19) to train and merge all prediction results into probability distributions. This task is tackled as a multi-class text classification task. As shown in Figure 9, the models take the labels of each quality scores and result in a label.

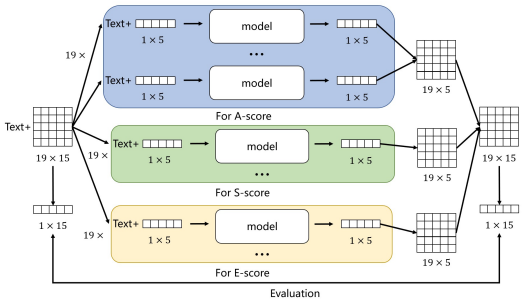

Figure 9: A dialogue input for label-based training

4.2 Evaluation metrics

We applied the evaluation metrics from the DialEval-1 Task. For the dialogue evaluation where both the ground truth data and the prediction results are represented as a distribution, two cross-bin measures exist: the Normalized Match Distance (NMD) and the Root Symmetric Normalized Order-aware Divergence (RSNOD) [16]. A lower score indicates that the performance of the model is better for the task.

4.3 Results

4.3.1 Result description. The DialEval-1 Task Organisers released three files, which is "Train set", "Dev set" and "Test set". We utilized the "Test set" to evaluate the performance of models, Table 1 and Table 2 show the evaluation scores for our best models respectively in "Test set". The terms of "A-score", "S-score" and "E-score" represent the mean evaluation scores for each dialogue. "Average" represents the average value of "A-score", "S-score" and "E-score". "BiLSTM" stands for the BiLSTM with attention model. The performance of 3 baseline models ("BL-lstm", "BL-popularity" and "BL-uniform") are officially provided by the DialEval-1 Task Organisers [20]. For the models, we only apply the label-based training method for DistilRoBERTa, which is DistilRoBERTa** in the tables and all models are trained with distribution-based training.

Our runs are the prediction results from the test set of the ensemble models. Almost all model we trained have a 10% to 20% decrease

Table 1: Results experiments for RSNOD

Model	A-score	S-score	E-score	Average
CNN	0.254	0.253	0.221	0.243
BiLSTM	0.264	0.266	0.229	0.253
LGC	**0.238**	0.226	0.184	**0.216**
ALBERT	0.271	0.281	0.243	0.264
DistilRoBERTa*	0.276	0.244	0.201	0.240
DistilRoBERTa**	0.248	**0.223**	**0.177**	**0.216**
BL-lstm	0.227	0.211	0.169	0.202
SKYMN-run2	0.241	0.218	0.178	0.213
SKYMN-run0	0.247	0.222	0.180	0.216
BL-popularity	0.247	0.229	0.261	0.246
SKYMN-run1	0.256	0.231	0.184	0.224
BL-uniform	0.271	0.281	0.243	0.265

*DistilRoBERTa is trained with distribution-based Training.
**DistilRoBERTa is trained with label-based Training.

Table 2: Results of experiments for NMD

Model	A-score	S-score	E-score	Average
CNN	0.226	0.218	0.193	0.212
BiLSTM	0.240	0.233	0.203	0.225
LGC	0.167	0.155	0.144	0.155
ALBERT	0.252	0.250	0.211	0.237
DistilRoBERTa*	0.252	0.249	0.211	0.237
DistilRoBERTa**	**0.163**	**0.147**	**0.128**	**0.146**
BL-lstm	0.159	0.141	0.125	0.142
SKYMN-run2	0.160	0.145	0.132	0.146
SKYMN-run0	0.161	0.148	0.132	0.147
BL-popularity	0.164	0.144	0.178	0.162
SKYMN-run1	0.166	0.152	0.134	0.151
BL-uniform	0.252	0.250	0.211	0.238

*DistilRoBERTa is trained with Distribution-based Training.
**DistilRoBERTa is trained with Label-based Training.

for RSNOD and NMD scores in test set than dev set. To avoid overfitting, increase robustness and get better accuracy, we apply weighted average ensemble method. For the best two models DistilRoBERTa** and LGC, we gave relatively large proportion. In run0 and run2, we also added small proportion of BiLSTM to increase the model diversity and to avoid the situation that DistilRoBERTa** and LGC model all have high weighted on wrong direction. The details of the results are presented in the Overview of the DialEval-1 Task [20].

- run0: 0.4DistilRoBERTa** + 0.4LGC + 0.2BiLSTM
- run1: 0.5DistilRoBERTa** + 0.5LGC
- run2: 0.3DistilRoBERTa** + 0.4LGC + 0.3BiLSTM

We computed randomised Tukey HSD p-values and effect sizes based on one-way ANOVA (without replication) [17]. Tables 3 to 8 show the statistical significance test results among our models (CNN, BiLSTM, LGC, ALBERT, DistilRoBERTa* and DistilRoBERTa**).

4.3.2 Results and discussion. As the results show, the performance of DistilRoBERTa** model is statistically significantly better than

Table 3: Statistical significance in terms of NMD (A-score) calculated by Randomised Tukey HSD tests

Model	significantly better than these models	
DistilRoBERTa**	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.305$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.754$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.787$)
LGC	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.229$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.678$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.711$)
CNN	ALBERT	($p < 0.0001$, $ES_{E1} = 0.564$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.596$)
BiLSTM	ALBERT	($p < 0.0001$, $ES_{E1} = 0.449$)

Table 4: Statistical significance in terms of NMD (S-score) calculated by Randomised Tukey HSD tests

Model	significantly better than these models	
DistilRoBERTa**	CNN	($p < 0.0001$, $ES_{E1} = 0.281$)
	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.349$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.991$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 1.027$)
LGC	CNN	($p < 0.0001$, $ES_{E1} = 0.249$)
	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.318$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.960$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.996$)
CNN	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.711$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.747$)
BiLSTM	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.642$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.678$)

Table 5: Statistical significance in terms of NMD (E-score) calculated by Randomised Tukey HSD tests

Model	significantly better than these models	
LGC	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.370$)
	CNN	($p < 0.0001$, $ES_{E1} = 0.406$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.790$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.791$)
DistilRoBERTa**	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.270$)
	CNN	($p < 0.0001$, $ES_{E1} = 0.305$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.690$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.691$)
BiLSTM	ALBERT	($p < 0.0001$, $ES_{E1} = 0.420$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.421$)
CNN	ALBERT	($p < 0.0001$, $ES_{E1} = 0.385$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.386$)

Table 6: Statistical significance in terms of RSNOD (A-score) calculated by Randomised Tukey HSD tests

Model	significantly better than these models	
DistilRoBERTa**	CNN	($p < 0.0001$, $ES_{E1} = 0.141$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.239$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.265$)
	LGC	($p < 0.0001$, $ES_{E1} = 0.300$)
	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.333$)
CNN	LGC	($p < 0.0001$, $ES_{E1} = 0.159$)
	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.191$)

Table 7: Statistical significance in terms of RSNOD (S-score) calculated by Randomised Tukey HSD tests

Model	significantly better than these models	
DistilRoBERTa**	LGC	($p < 0.0001$, $ES_{E1} = 0.187$)
	CNN	($p < 0.0001$, $ES_{E1} = 0.212$)
	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.386$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.563$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.630$)
LGC	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.200$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.376$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.443$)
CNN	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.175$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.351$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.418$)
BiLSTM	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.176$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.243$)

Table 8: Statistical significance in terms of RSNOD (E-score) calculated by Randomised Tukey HSD tests

Model	significantly better than these models	
DistilRoBERTa**	LGC	($p < 0.0001$, $ES_{E1} = 0.211$)
	CNN	($p < 0.0001$, $ES_{E1} = 0.306$)
	BiLSTM	($p < 0.0001$, $ES_{E1} = 0.348$)
	ALBERT	($p < 0.0001$, $ES_{E1} = 0.621$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.651$)
LGC	ALBERT	($p < 0.0001$, $ES_{E1} = 0.410$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.440$)
CNN	ALBERT	($p < 0.0001$, $ES_{E1} = 0.315$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.346$)
BiLSTM	ALBERT	($p < 0.0001$, $ES_{E1} = 0.273$)
	DistilRoBERTa*	($p < 0.0001$, $ES_{E1} = 0.303$)

other models evaluated by RSNOD. DistilRoBERTa** model outperforms CNN, BiLSTM, ALBERT and DistilRoBERTa* evaluated by NMD. According to the results of DistilRoBERTa, the performance of the model applying the label-based training method is 10.0% better in mean RSNOD and 38.4% better in mean NMD than the distribution-based training method. The DistilRoBERTa** model achieves the best scores among all models. For the distribution-based training method, the model finds the information and relationship between texts and the estimated distribution of dialogue quality ratings for the entire dialogue. For the label-based training method, a model builds an evaluation system for each annotator in each quality scale. Therefore, this method might represent the process of obtaining the data and simulating a group of annotators scoring dialogues. Furthermore, we are able to use some evaluation metrics of multi-class classification tasks to evaluate our models for analysis.

In addition, the LGC model achieved close to the best performance, which significantly outperforms other models except DistilRoBERTa** in most cases. Our LGC model has a CNN layer with a kernel size of 1 for the last layer to complete the missing word information. From our experiments, concatenating max pooling with the average pooling seems to achieve the best score compared with using them individually.

For the pre-trained models applied to distribution-based training, the performances of DistilRoBERTa* and ALBERT are similar. Moreover, the pre-trained models are worse than the baseline models and other deep learning models. For this task, the small dataset might be one of the influencing factors. Compared to a CNN and BiLSTM, the depth of the LGC model allows for more parameters and a better nonlinear expression. The performance of BiLSTM is the worst among the CNN, BiLSTM, Baseline, and LGC. Compared to the CNN, the attention layer in the BiLSTM might not capture sufficient information to form a distribution.

5 CONCLUSIONS

In this paper, we propose two training methods, a distribution-based training method, and a label-based training method, for an English learning quality subtask. The experiments and evaluation results show that the label-based training method outperforms the distribution-based training method. We also implemented different deep learning models to explore their performance levels.

The DistilRoBERTa model with the label-based training method (DistilRoBERTa**) achieved the best result among our models. However, the pre-trained models with a distribution-based training method (ALBERT and DistilRoBERTa**) achieved the worst results. For this task, the pre-trained model might be influenced by the small size of the dataset. It might be a good choice to utilize a pre-trained model for building the word embeddings. The LGC model is slightly behind the first place model. Therefore, we increased the depth of the model appropriately to improve the performance. Continuing to deepen the model seems to be a feasible direction.

Our future study will focus on applying the label-based training method to all models and exploring additional pre-trained models. Moreover, we will try to utilize the pre-trained models as word embedding encoders. Furthermore, machine learning methods and some retrieval-based methods are also worth implementing.

ACKNOWLEDGEMENT

We would like to express thanks to members in The Real Sakai Laboratory[3], Waseda University for giving us suggestions, especially Wanqi Wu, Zhaohao Zeng and Sosuke Kato.

REFERENCES

[1] 2011. English Gigaword Fifth Edition - Linguistic Data Consortium. https://catalog.ldc.upenn.edu/LDC2011T07. (Accessed on 08/15/2020).

[2] 2014. Wikimedia Downloads. https://dumps.wikimedia.org/. (Accessed on 08/15/2020).

[3] 2020. GitHub - google-research/albert: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. https://github.com/google-research/albert. (Accessed on 09/09/2020).

[4] Hsiang-En Cherng and Chia-Hui Chang. 2019. Dialogue quality and nugget detection for short text conversation (STC-3) based on hierarchical multi-stack model with memory enhance structure. *NTCIR14* (2019), 362–375.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[6] Kai Cong and Wai Lam. 2019. CUIS at the NTCIR-14 STC-3 DQ Subtask. *NTCIR14* (2019), 390–398.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[8] Łukasz Kaiser and Ilya Sutskever. 2015. Neural gpus learn algorithms. *arXiv preprint arXiv:1511.08228* (2015).

[9] Sosuke Kato, Rikiya Suzuki, Zhaohao Zeng, and Tetsuya Sakai. 2019. SLSTC at the NTCIR-14 STC-3 dialogue quality and nugget detection subtasks. *NTCIR14* (2019), 355–361.

[10] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[11] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv:cs.CL/1909.11942

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[13] Christopher Olah. 2015. Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[14] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[15] Matthew Peters, Sebastian Ruder, and Noah A Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. *arXiv preprint arXiv:1903.05987* (2019).

[16] Tetsuya Sakai. 2018. Comparing two binned probability distributions for information access evaluation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1073–1076.

[17] Tetsuya Sakai. 2018. *Laboratory experiments in information retrieval: Sample Sizes, Effect Sizes, and Statistical Power*. Springer. https://link.springer.com/book/10.1007/978-981-13-1199-4.

[18] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*.

[19] Ming Yan, Maofu Liu, and Junyi Xiang. 2019. WUST at the NTCIR-14 STC-3 Dialogue Quality and Nugget Detection Subtask. *NTCIR14* (2019), 376–382.

[20] Zhaohao Zeng, Sosuke Kato, Tetsuya Sakai, and Inho Kang. 2020. Overview of the NTCIR-15 Dialogue Evaluation (DialEval-1) Task. In *Proceedings of the NTCIR-15 Conference*.

[21] 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)*, Vol. 2. 207–212.

[3] http://sakailab.com/english/