Adversarial Training and Decoding Strategies for End-to-end Neural Conversation Models

Hori, Takaaki; Wang, Wen; Koji, Yusuke; Hori, Chiori; Harsham, Bret A.; Hershey, John

TR2018-161 December 07, 2018

Abstract

This paper presents adversarial training and decoding methods for neural conversation models that can generate natural responses given dialog contexts. In our prior work, we built several end-to-end conversation systems for the 6th Dialog System Technology Challenges (DSTC6) Twitter help-desk dialog task. These systems included novel extensions of sequence adversarial training, example-based response extraction, and Minimum Bayes-Risk based system combination. In DSTC6, our systems achieved the best performance in most objective measures such as BLEU and METEOR scores and decent performance in a subjective measure based on human rating. In this paper, we provide a complete set of our experiments for DSTC6 and further extend the training and decoding strategies more focusing on improving the subjective measure, where we combine responses of three adversarial models. Experimental results demonstrate that the extended methods improve the human rating score and outperform the best score in DSTC6.

Computer Speech and Language

Adversarial Training and Decoding Strategies for End-to-end Neural Conversation Models

Takaaki Hori[a], Wen Wang[b], Yusuke Koji[b], Chiori Hori[a],
Bret Harsham[a], John R. Hershey[a]

[a]*Mitsubishi Electric Research Laboratories, Cambridge, MA, USA*
[b]*Information Technology R&D Center, Mitsubishi Electric Corporation, Ofuna, Kamakura, Kanagawa, Japan*

Abstract

This paper presents adversarial training and decoding methods for neural conversation models that can generate natural responses given dialog contexts. In our prior work, we built several end-to-end conversation systems for the 6th Dialog System Technology Challenges (DSTC6) Twitter help-desk dialog task. These systems included novel extensions of sequence adversarial training, example-based response extraction, and Minimum Bayes-Risk based system combination. In DSTC6, our systems achieved the best performance in most objective measures such as BLEU and METEOR scores and decent performance in a subjective measure based on human rating. In this paper, we provide a complete set of our experiments for DSTC6 and further extend the training and decoding strategies more focusing on improving the subjective measure, where we combine responses of three adversarial models. Experimental results demonstrate that the extended methods improve the human rating score and outperform the best score in DSTC6.

keywords: dialog system, conversation model, sequence-to-sequence model, sentence generation

1 Introduction

Dialog system technology [1, 2, 3] has been widely used in many applications. Generally, a dialog system consists of a pipeline of data processing modules, including automatic speech recognition (ASR), spoken language understanding (SLU), dialog management (DM), sentence generation (SG), and speech synthesis. The SLU module predicts the user's intention from the user's utterance [4, 5], usually by converting text or ASR result to a semantic representation consisting of a sequence of concept tags or a set of slot-value pairs. The DM module chooses the next system action/response based on the current state and the user's intention. The SG module generates system reply sentences corresponding to the selected reply policy.

Recently, dialog systems have greatly improved because the accuracy of each module has been enhanced by machine learning techniques. However, there are still some problems with using the pipeline of modules architecture: The SLU, DM, and SG modules each require their own set of manually labeled training data. The DM and SG modules often rely on hand-crafted rules. In addition, such dialog systems are often not good at flexible interaction outside predefined scenarios, because intention labeling schemes are limited by the scenario design. For all of these reasons, conventional dialog systems are expensive to implement.

To solve these problems, end-to-end dialog systems are gathering attention in the research field. The end-to-end approach utilizes only paired input and output sentences to train the dialog model without relying on pre-designed data processing modules or intermediate internal data representations such as concept tags and slot-value pairs. End-to-end systems can be trained to directly map a user's utterance to a system response sentence and/or action. This significantly reduces the data preparation and system development cost. Recently, several types of sequence-to-sequence models have been applied to end-to-end dialog systems, and it has been shown that they can be trained in a completely data-driven manner. The end-to-end approach also has a potential to handle flexible conversation between the user and the system by training the model with large conversational data [6, 7].

In this paper, we propose an end-to-end dialog system based on several sequence-to-sequence modeling and decoding techniques, and evaluate the performance with the 6th dialog system technology challenges (DSTC6) [8] end-to-end conversation modeling track [9]. DSTC was originally a series of dialog state tracking challenges [10], where the task was to predict a set of slot-value pairs for each utterance or segment in a dialog [11]. From the 6th challenge, the focus of DSTC has been expanded to broader areas of dialog system technology. The goal of the end-to-end conversation modeling track task is to generate system sentences in response to each user input in a given context. In this task, the training and test data consists of un-annotated text dialogs which are relatively inexpensive to collect for real tasks.

Our DSTC6 system was designed to improve the response quality for both objective and subjective evaluation metrics. The reason we took this approach is not only to record good numbers in the challenge but also we thought that improving the performance in the both measures was quite important to build better dialog systems. The objective measures such as BLEU and METEOR scores focus on a similarity between the real human response and the system response while the subjective measures focus on naturalness and appropriateness of the system response based on humans' preference. Although subjective measures usually give higher scores for real human's responses, they sometimes give low scores when a human operator could not meet a user's request due to some reason. In contrast, when a system solved user's problem in the response, the score tends to be high even though it rarely happens in the real service. Objective measures sometimes give lower scores even for appropriate responses due to the lack of references covering various appropriate responses. Thus, the both measures are not perfect but complementary to each other. Hence it is reasonable to improve the system to have higher scores in the both measures.

Our proposed system has several key features to improve objective and subjective scores. We employ multiple conversation models, a long short-term memory (LSTM) encoder decoder, a bidirectional LSTM (BLSTM) encoder decoder, and a hierarchical recurrent encoder decoder (HRED). The responses given by these models are combined by a minimum Bayes risk (MBR)-based system combination technique to improve objective scores. On the other hand, sequence adversarial training and an example-based method are used to improve subjective human rating scores. Furthermore, we extend the reward function for sequence adversarial training to further improve the both scores. Experimental results on the Twitter help-desk dialog task show that the combination of these techniques effectively improves the performance in all the evaluation metrics for the end-to-end conversation modeling track of DSTC6.

This paper is an extended version of our system description paper [12] presented in the DSTC6 workshop, and includes new results that complete the evaluation of our proposed system. The contribution of the paper can be summarized as follows.

1. Thorough evaluation of three different neural conversation models using a common task,

2. Application of sequence adversarial training and extension of its objective function to improve both objective and subjective evaluation metrics,

3. MBR-based system combination of multiple neural conversation models, which has not been applied to neural conversation systems, and

4. Example-based response selection using an embedding-based context similarity.

5. Demonstrate our final system based on system combination of adversarially trained models achieves the best human rating score while keeping high objective scores.

2 System Architecture

Figure 1 shows the architecture of our DSTC6 end-to-end conversation system. In the training phase (the upper part of the figure), sequence-to-sequence models are first trained with the Cross-Entropy (CE) criterion using the training corpus, where the the system has three models LSTM, BLSTM and HRED. Furthermore, sequence adversarial training is performed for the models to generate better sentences.

In the generation phase (the lower part of figure), we employ model-based sentence generation and example-based response selection. For system response, either generated or example response is selected according to a reliability of the example. We also apply a system combination technique to enhance the response sentence by combining multiple hypotheses generated by different models.

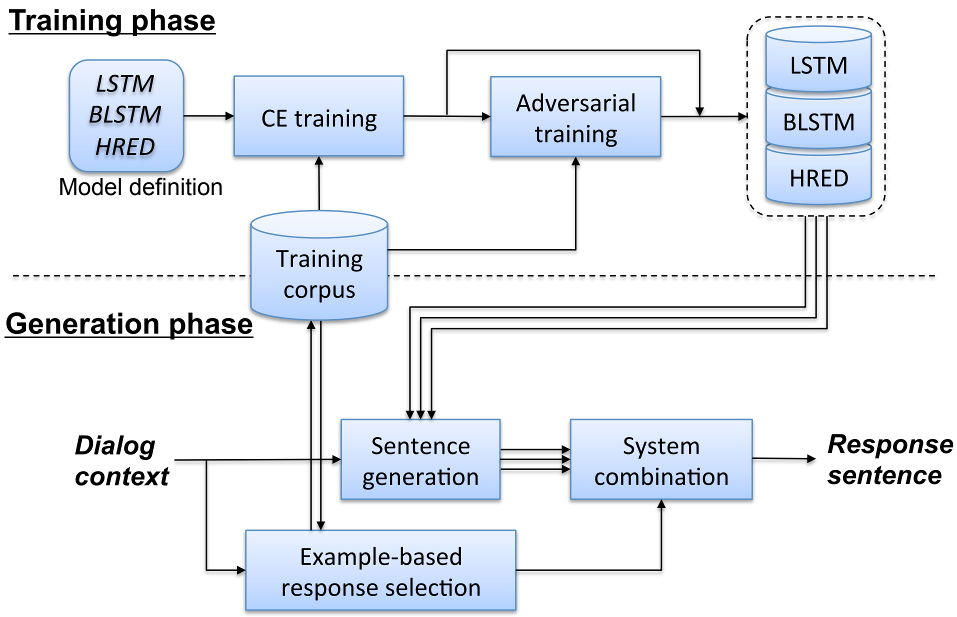

Figure 1: System architecture. The upper part corresponds to the model training phase, where we first define the model structure of LSTM, BLSTM, and HRED, then apply cross entropy (CE) training to each of the models using the training corpus, optionally we further apply adversarial training, and finally obtain the trained models. The lower part corresponds to the generation phase, where we generate response sentences from the trained models and apply system combination. We also use example-based response selection if a similar context is found in the training corpus.

Details of each module are described in the following sections. Section 3 presents conversation models and training algorithms used in the training phase of the system. Section 4 explains response generation techniques used in the generation phase.

3 Conversation Model Training

3.1 Neural conversation models

The neural conversation model [6] is designed as an encoder decoder network using recurrent neural networks (RNNs). Let X and Y be input and output sequences, respectively. The model is used to compute posterior probability distribution $P(Y|X)$. For conversation modeling, X is word sequence $x_1, \dots, x_T$ representing all previous sentences in a conversation, and Y is word sequence $y_1, \dots, y_M$ corresponding to a system response sentence. X contains all of the previous turns of the conversation, concatenated in sequence, separated by markers that indicate to the model not only that a new turn has started, but which speaker said that sentence.

The encoder decoder network is used to compute $P(Y|X)$ [6], where the encoder

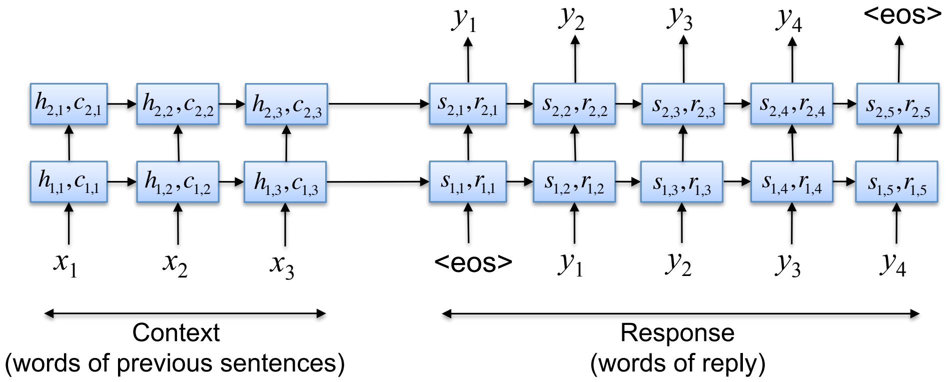

Figure 2: LSTM-based encoder decoder [6]. The encoder (on the left-hand side) embeds contextual information, i.e., words of previous sentences, into state vectors, and the decoder (on the right-hand side) predicts the response, i.e., words of reply, from the encoder's last state. In this example, the encoder and the decoder have two LSTM layers, respectively.

first converts X to H a set of hidden vectors representing a contextual information, and the decoder generates system response Y word by word referring to H, i.e.,

$$H = \text{Encoder}(X) \tag{1}$$

$$y_m \sim \text{Decoder}(y_1, \ldots, y_{m-1}, H). \tag{2}$$

Since the decoder network provides a probability distribution of the next label as

$$P(\cdot|y_1, \ldots, y_{m-1}, X) \triangleq \text{Decoder}(y_1, \ldots, y_{m-1}, H), \tag{3}$$

we can compute $P(Y|X)$ using the probabilistic chain rule:

$$P(Y|X) = \prod_{m=1}^{M} P(y_m|y_1, \ldots, y_{m-1}, X). \tag{4}$$

We employ three types of encoder decoder networks, a LSTM encoder decoder (Fig. 2), a BLSTM encoder decoder (Fig. 3), and a HRED (Fig. 4). When using an encoder decoder network, each word of X and Y is converted to a word vector using word embedding layers included in the network. In this work, we put a linear layer before recurrent layers in each of the encoder and decoder networks, and a word is converted as

$$x'_t = \text{Linear}(x_t; \theta^E_{\text{enc}}) \tag{5}$$

$$y'_m = \text{Linear}(y_m; \theta^E_{\text{dec}}), \tag{6}$$

where $\text{Linear}(\cdot; \theta^E)$ denotes a linear transformation with a set of parameters θ^E for the encoder or the decoder, and we assume x_t and y_m are represented as one-hot vectors. The word embedding layers are jointly trained with the encoder decoder network. Note that these word embedding layers are not explicitly depicted in Figs. 2–4).

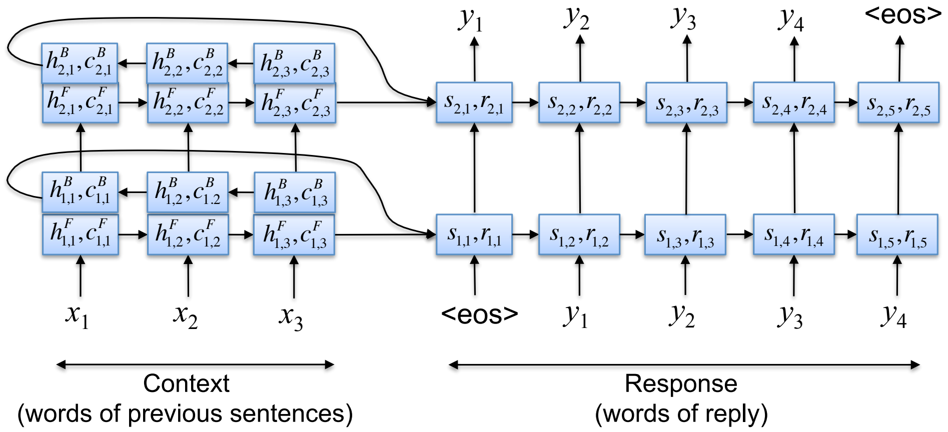

Figure 3: BLSTM-based encoder decoder. The encoder (on the left-hand side) embeds contextual information using forward and backward LSTM layers, where the rightmost states of the forward layers and the leftmost states of the backward layers are concatenated and fed to the decoder (on the right-hand side). In this example, the encoder and the decoder have two LSTM layers, respectively.

3.1.1 LSTM encoder decoder

Given a LSTM encoder decoder of L layers, the encoding process (on the left hand side of Fig. 2) outputs hidden state $h_{l,t}$ and cell state $c_{l,t}$ for $l = 1, \ldots, L$ and $t = 1, \ldots, T$ as:

$$h_{l,t}, c_{l,t} = \mathrm{LSTM}\left(h_{l-1,t}, h_{l,t-1}, c_{l,t-1}; \theta_{\mathrm{enc},l}\right) \tag{7}$$

where $\mathrm{LSTM}(\cdot; \theta_{\mathrm{enc},l})$ is a LSTM function with a set of parameters $\theta_{\mathrm{enc},l}$ for the l-th LSTM layer of the encoder. We initialize activation vectors such that $h_{0,t} = x'_t$, $h_{l,0} = \mathbf{0}$ and $c_{l,0} = \mathbf{0}$. Since the last hidden and cell states are given to the decoder network, the encoder function is defined as

$$H = \mathrm{Encoder}(X) \triangleq \{h_{l,T}, c_{l,T} | l = 1, \ldots, L\}. \tag{8}$$

The decoding process (right hand side in Fig. 2) computes hidden state $s_{l,m}$ and cell state $r_{l,m}$ for $l = 1, \ldots, L$ and $m = 1, \ldots, M$ of the decoder as:

$$s_{l,m}, r_{l,m} = \mathrm{LSTM}\left(s_{l-1,m}, s_{l,m-1}, r_{l,m-1}; \theta_{\mathrm{dec},l}\right), \tag{9}$$

where $\theta_{\mathrm{dec},l}$ is a set of decoder parameters for the l-th LSTM layer of the decoder. The decoder states at $m = 0$ are initialized with H such that

$$s_{l,0} = h_{l,T}, \quad r_{l,0} = c_{l,T} \quad \text{for } h_{l,T}, c_{l,T} \in H,\ l = 1, \ldots, L. \tag{10}$$

We also initialize activation vectors such that $s_{0,m} = y'_{m-1}$, $y'_0 = y'_M = \mathrm{Linear}(\texttt{<eos>}; \theta^E_{\mathrm{dec}})$, where `<eos>` is a special symbol representing the end of sequence.

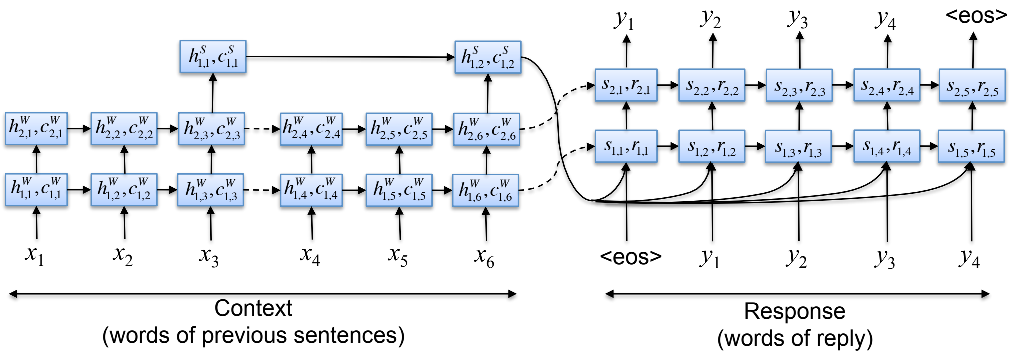

Figure 4: Hierarchical recurrent encoder decoder (HRED) [13]. The hierarchical encoder (on the left-hand side) has word-level and sentence-level layers. In the word-level layers, sentence embedding vectors are obtained at each sentence end, which are then fed to the sentence-level layer. The last hidden state of the sentence-level layer is fed to all the decoder states (on the right-hand side). Note that connections represented in dashed lines are not included in the original HRED.

The decoder function that produces the word probability distribution is defined as

$$\text{Decoder}(y_1, \ldots, y_{m-1}, H) \triangleq \text{Softmax}\left(\text{Linear}(s_{L,m}; \theta^O_{\text{dec}})\right), \tag{11}$$

where each hidden state $s_{L,m}$ is converted to the distribution using the linear transformation followed by the softmax function. θ^O_{dec} denotes a set of parameters for the linear transformation including a transformation matrix and a bias vector.

3.1.2 BLSTM encoder decoder

A BLSTM encoder decoder has bidirectional LSTM layers in the encoder and unidirectional LSTM layers in the decoder, where the encoder and the decoder have L layers, respectively. As shown in Fig. 3, the last hidden and cell states of the forward layers and the first hidden and cell states of the backward layers are concatenated and fed to a LSTM decoder. The BLSTM encoder is used to obtain hidden and cell states as:

$$h^F_{l,t}, c^F_{l,t} = \text{LSTM}\left(h_{l-1,t}, h^F_{l,t-1}, c^F_{l,t-1}; \theta^F_{\text{enc},l}\right) \tag{12}$$

$$h^B_{l,t}, c^B_{l,t} = \text{LSTM}\left(h_{l-1,t}, h^B_{l,t+1}, c^B_{l,t+1}; \theta^B_{\text{enc},l}\right) \tag{13}$$

$$h_{l,t} = \left(h^F_{l,t}, h^B_{l,t}\right), \tag{14}$$

where $h^F_{l,0}$, $c^F_{l,0}$, $h^B_{l,T+1}$ and $c^B_{l,T+1}$ are initialized with zero vectors. $\theta^F_{\text{enc},l}$ and $\theta^B_{\text{enc},l}$ are sets of parameters of the l-th forward and backward LSTM layers in the BLSTM encoder, respectively. At each layer, the forward and backward state vectors are concatenated by Eq. (14) and fed to the upper layer.

Since the last forward states and the first backward states are given to the decoder network, the BLSTM encoder function is defined as

$$H = \text{Encoder}(X) \triangleq \left\{h^F_{l,T}, c^F_{l,T}, h^B_{l,1}, c^B_{l,1} | l = 1, \ldots, L\right\}. \tag{15}$$

The decoder states at $m = 0$ are initialized with H such that

$$s_{l,0} = \left(h^F_{l,T}, h^B_{l,1}\right),\ r_{l,0} = \left(c^F_{l,T}, c^B_{l,1}\right) \quad \text{for } h^F_{l,T}, c^F_{l,T}, h^B_{l,1}, c^B_{l,1} \in H,\ l = 1, \ldots, L. \tag{16}$$

Given the initial states, the decoder can predict Y in the same way as the LSTM encoder decoder according to Eqs. (9) and (11).

3.1.3 Hierarchical recurrent encoder decoder

A HRED [13] has a hierarchical structure of word-level and sentence-level propagation processes as shown in Fig. 4. In the word-level layer of the hierarchical encoder, a sentence embedding vector is obtained at each sentence end, which is then fed to the sentence-level layer. The last hidden state of the sentence-level layer is fed to all the decoder states as an entire contextual information. In our system, the initial encoder state of each word-level layer is also given from the last state of the previous sentence, and the initial decoder state is given from the last encoder state of the word-level layer. These connections are depicted with dashed lines in the figure, because they are not included in the original HRED [13]. But, in our preliminary experiments, these connections yielded slightly better BLEU and METEOR scores. An HRED network can capture sentence-level state transitions in the dialog, which is potentially effective to predict the next response when it has longer contextual information.

Suppose a HRED has an L-layer word-level encoder, a K-layer sentence-level encoder, and an L-layer decoder, and input word sequence X consists of U sentences. Sentence chunks can be obtained automatically using a set of predefined sentence-end markers such as independent period, question, exclamation, and carriage return characters.

With the word-level encoder, X is encoded to a sequence of word-level hidden states $h^W_{l,t}$ and cell states $c^W_{l,t}$ as:

$$h^W_{l,t}, c^W_{l,t} = \text{LSTM}\left(h^W_{l-1,t}, h^W_{l,t-1}, c^W_{l,t-1}; \theta^W_{\text{enc},l}\right) \tag{17}$$

where $\theta^W_{\text{enc},l}$ is a set of parameters for the l-th LSTM layer. We initialize activation vectors such that $h^W_{0,t} = x'_t$, $h^W_{l,0} = \mathbf{0}$ and $c^W_{l,0} = \mathbf{0}$. The hidden state sequence on top of word-level layers, $h_{L,t}$, is then fed to the sentence-level encoder to obtain sentence-level hidden states $h^S_{k,\tau}$ and cell states $c^S_{k,\tau}$ for $k = 1, \ldots, K$ and $\tau = 1, \ldots, U$ as

$$h^S_{k,\tau}, c^S_{k,\tau} = \text{LSTM}\left(h^S_{k-1,\tau}, h^S_{k,\tau-1}, c^S_{k,\tau-1}; \theta^S_{\text{enc},k}\right) \tag{18}$$

where $\theta^S_{\text{enc},k}$ is a set of parameters for the k-th LSTM layer of the sentence-level encoder. We initialize activation vectors such that $h^S_{0,\tau} = h^W_{L,\text{widx}(\tau)}$, $h^S_{k,0} = \mathbf{0}$ and $c^S_{k,0} = \mathbf{0}$, where $\text{widx}(\tau)$ converts sentence position index τ to the word position index corresponding to the end of the τ-th sentence. In our HRED, the last hidden and cell states of the word-level layers and the last hidden state on top of the sentence-level layer are given to the decoder network. Thus, the encoder function is defined as

$$H = \text{Encoder}(X) \triangleq \left\{h^W_{l,T}, c^W_{l,T}, h^S_{K,U} | l = 1, \ldots, L\right\}. \tag{19}$$

The decoder states at $m = 0$ are initialized with H such that

$$s_{l,0} = h^W_{l,T}, \quad r_{l,0} = c^W_{l,T} \quad \text{for } h^W_{l,T}, c^W_{l,T} \in H, \; l = 1, \ldots, L \tag{20}$$

and sentence-level hidden vector $h^S_{K,U}$ is fed to the decoder network together with input vectors y'_m, i.e., we assume $s_{0,m} = (y'_m, h^S_{K,U})$ for $m = 1, \ldots, M$ in Eq. (9).

3.2 Model training strategies

Neural conversation models are usually trained to minimize cross entropy (CE) loss

$$\mathcal{L}_{CE}(\theta) = -\log P(Y|X; \theta) \tag{21}$$

for a set of paired context and response sentences (X, Y), where θ denotes the set of model parameters.

In our system, we also apply adversarial training [14] to the conversation models to generate more human-like sentences. In the adversarial training, a generative model and a discriminator are jointly trained, where the discriminator is trained to classify system-generated and human-generated sentences as a binary classification problem, and the generative model is trained to generate sentences so that they are judged as human-generated sentences by the discriminator.

Adversarial training was originally proposed for image generation tasks. It has also been applied to text generation tasks such as sentence generation [15], machine translation [16], image captioning [17], and open-domain dialog generation [18].

To train the models, we use a policy gradient optimization based on the REINFORCE algorithm [19]. First, the generative model, i.e., conversation model, is trained with the cross entropy criterion. The discriminator is also trained using human-generated (positive) samples and machine-generated (negative) samples.

In the REINFORCE algorithm, the reward is given as the probability that the sentence is generated by human, which is computed by the discriminator. The generative model is trained to generate sentences to obtain higher rewards, which means that generated sentences will become more human-like sentences. The objective function for training the generative model is

$$J_{ADV}(\theta) = E_{Y \sim P_G(Y|X;\theta)}[P_D(+1|\{X, Y\})], \tag{22}$$

and its gradient is computed as

$$\begin{aligned}\nabla J_{ADV}(\theta) \approx & [P_D(+1|\{X, Y\}) - b(\{X, Y\})] \\ & \nabla \sum_t \log P_G(y_t | X, y_1, \ldots, y_{t-1}; \theta),\end{aligned} \tag{23}$$

where $P_G(Y|X; \theta)$ is the probability distribution on Y given X computed by the generative model, and $P_D(+1|\{X, Y\})$ is the probability that Y is generated by a human (rather than by a machine) in response to X, which is given by the discriminator.

Algorithm 1 Sequence Adversarial Training

```
procedure SequenceAdversarialTraining($TrainCorpus$, $\theta_G$, $\theta_D$)
    for $i = 1, \ldots, N\_Iterations$ do
        for $j = 1, \ldots, N\_DSteps$ do
            Sample $(X, Y')$ from $TrainCorpus$
            Sample $Y \sim P_G(\cdot|X; \theta_G)$
            Update $\theta_D$ using $(X, Y')$ as positive examples and $(X, Y)$ as negative examples
        end for
        for $k = 1, \ldots, N\_GSteps$ do
            Sample $(X, Y')$ from $TrainCorpus$
            Sample $Y \sim P_G(\cdot|X; \theta_G)$
            Update $\theta_G$ to increase $P_D(+1|\{X, Y\}; \theta_D) + \lambda \mathrm{Sim}(Y, Y')$
        end for
        Update $\theta_G$ using $(X, Y')$ for teacher forcing
    end for
end procedure
```

$b(\{X, Y\})$ is the baseline value [19]. The generative model and the discriminator are alternately updated through the training iterations. We also added a teacher forcing step, i.e., updating with the cross-entropy criterion for the generative model as in [18].

Moreover, we extend the reward function to regularize the generative model as

$$J_{ADVS}(\theta) = E_{Y \sim P_G(Y|X;\theta)} \left[P_D(+1|\{X, Y\}) + \lambda \mathrm{Sim}(Y, Y') \right], \tag{24}$$

where we incorporate a similarity measure between the generated sentence Y and the reference (ground truth) sentence Y' in the reward function. We use a similarity function $\mathrm{Sim}(Y, Y')$ with scaling factor λ, which is a cosine similarity between average word embedding vectors of the sentences. We used the same embedding model as the example-based method in Section 4.3.

The similarity term hopefully avoids generating semantically mismatched sentences even though they are characteristic of human generated sentences. The teacher forcing step may afford a similar regularization effect to the model. However, it relies on the cross entropy loss, i.e., the model is strongly affected by the distribution of the training corpus, and the generated sentences become more likely to be machine generated. Accordingly, the similarity term can improve the sentence quality in a different way from the teacher forcing.

We summarize the training procedure in Algorithm 1, where θ_G and θ_D are sets of parameters of a generative model and a discriminator, which have been pretrained using the cross entropy criterion before adversarial training. $N\_Iterations$ is the number of training iterations. N_DSteps and N_GSteps are the numbers of generation and discrimination steps in each iteration.

4 Response Generation

4.1 Basic method

Given a conversation model $P(Y|X;\theta)$ and dialog context X, the system response is obtained as the most likely hypothesis $\hat{Y}$:

$$\hat{Y} = \arg\max_{Y \in \mathcal{V}^+} P(Y|X;\theta) \tag{25}$$

$$= \arg\max_{Y \in \mathcal{V}^+} \prod_{m=1}^{M_Y} P(y_m|y_1, \ldots, y_{m-1}, X;\theta), \tag{26}$$

where $\mathcal{V}^+$ denotes a set of sequences of one or more words in system vocabulary $\mathcal{V}$, and M_Y indicates the length of Y. To find $\hat{Y}$ efficiently, a beam search technique is usually used, since Eq. (26) is computationally intractable to consider all possible Y. The beam search method can also generate n-best hypotheses. The generated hypotheses are used for system combination described in the next section (Section 4.2).

4.2 System combination

System combination is a technique to combine multiple hypotheses. Each component system generates sentence hypotheses based on a single model, and the hypotheses of multiple systems are combined to generate a better response.

To perform system combination, we apply a minimum Bayes-risk (MBR) decoding [20, 21], which can improve the sentence quality by focusing on a specific evaluation metric.

In MBR decoding, the decoding objective is defined as

$$\hat{Y} = \arg\max_{Y \in \mathcal{V}^*} \sum_{Y' \in \mathcal{V}^*} P(Y'|X)E(Y', Y), \tag{27}$$

where $E(Y', Y)$ denotes an evaluation metric assuming Y' is a reference (ground-truth) and Y is a hypothesis (generated description). The summation on the right-hand side calculates the expected value of the evaluation metric over $P(Y'|X)$, and the MBR decoder finds hypothesis $\hat{Y}$ that maximizes (or minimizes) the expected evaluation metric. Since it is intractable to enumerate all possible word sequences in vocabulary $\mathcal{V}$, we usually limit them to the n-best hypotheses generated by a standard decoder. Although in theory the probability distribution $P(Y'|X)$ should be the true distribution, we instead compute it using the encoder-decoder model since the true distribution is unknown.

In this approach, any evaluation metric can be used. If we use BLEU [22] score, the metric can be computed as

$$E(Y', Y) = \exp\left(\sum_{n=1}^{N} \log \frac{p_n(Y', Y)}{N}\right) \times \gamma(Y', Y), \tag{28}$$

where N is the order of the BLEU score (usually $N = 4$), and $p_n(Y', Y)$ is the precision of n-grams in hypothesis Y. The penalty term, $\gamma(Y', Y) = 1$ if $\mathrm{len}(Y') < \mathrm{len}(Y)$ and $\exp(1 - \mathrm{len}(Y')/\mathrm{len}(Y))$ otherwise, penalizes hypotheses Y that are shorter than reference Y'.

For system combination, multiple n-best lists obtained by the different systems are first concatenated into one list, where the posterior probability of each hypothesis is rescaled so that the sum of the probabilities equals one in the concatenated list. Then, the MBR decoding is performed to select the best hypothesis according to Eq. (27). Since the evaluation metric usually indicates a similarity between two sentences, hypotheses that are similar to each other receive higher expected scores and one with the highest score is selected as the final output.

4.3 Example-based response selection

We also use an example-based method. When the system finds a similar context in a training corpus, it outputs the response corresponding to the context. Suppose dialogs in the training corpus are represented as following format:

$$(X'_i, Y'_i), i = 1, ..., N \tag{29}$$

where X'_i is the sequence of all previous sentences in dialog i, Y'_i is the system response, and N is the total number of dialogs in the corpus. Given previous sentences X as an input, the similarity between X and X'_i is computed for each training dialog, where a cosine similarity is used. Then reference Y'_i corresponding to the highest similarity is regarded as system output $\hat{Y}$, i.e.,

$$\hat{Y} = Y'_{\hat{i}} \tag{30}$$

$$\hat{i} = \arg\max_{i=1,\ldots,N} \mathrm{Sim}(X, X'_i). \tag{31}$$

When computing the similarity, word vectors obtained by word2vec [23] is applied to feature extraction. Firstly a training corpus is used to obtain a word2vec model. Secondly, word vectors in the input sequence are averaged to obtain the final feature vector. The similarity is computed as

$$\mathrm{Sim}(X, X') = \mathrm{CosineSimilarity}(\mathrm{Embed}(X), \mathrm{Embed}(X')), \tag{32}$$

where

$$\mathrm{Embed}(X) = \frac{\sum_{x \in X} \mathrm{word2vec}(x)}{\left|\sum_{x \in X} \mathrm{word2vec}(x)\right|}, \tag{33}$$

and word2vec$(\cdot)$ converts word x to its vector representation.

The example-based response is combined with other sentence generated sentences as shown in Figure 1. If the highest similarity score in Eq. (31) is larger than a predefined threshold, $Y'_{\hat{i}}$ is used for the system output, otherwise the generated sentence is used.

5 Related Work

There is a lot of prior work done for end-to-end conversation modeling and training [6, 13, 24, 18, 25]. A neural conversation model was proposed by [6], where LSTM-based sequence-to-sequence models were trained with a large amount of conversational text corpus. Since the examples of generated sentences looked reasonable, this approach gained attention in the research field. However, this model is basically trained with cross entropy loss, i.e., maximum likelihood criterion, and therefore the system responses tend to be very common sentences in the corpus, which often degrade subjective evaluation scores. To solve this problem, diversity-promoting objective functions have been proposed [24, 18, 25]. Our system employs an adversarial training approach of [18] since it outperformed the maximum mutual information (MMI) approach [25] in pair-wise human judgment evaluation [18]. However, these diversity-promoting functions only focus on improving subjective evaluation scores. In this paper, we extend the objective function to improve both subjective and objective evaluation scores by incorporating a semantic similarity measure between the reference and system responses as a regularization term.

Our system also has system combination module. Although the system combination technique has previously been applied to speech recognition [26, 27] and machine translation [28], it has not yet been used for dialog response generation using multiple neural conversation models (to the best of our knowledge). To perform system combination, we apply a minimum Bayes-risk (MBR) decoding [20, 21], which can improve the sentence quality by focusing on a specific objective evaluation metric. Furthermore, system combination of adversarially trained models is a novel approach, where we aim at selecting a response with a high objective score from human-like diversified responses generated by the trained models for improving the both subjective and objective scores.

The example-based methods are also popular in dialog systems [29, 30] since the system can respond with a real example of natural human response in a dialog corpus. However, it is quite expensive to prepare a sufficient number of examples, which cover various users' utterances and the corresponding responses. Our system takes the example-based response only when a very similar context is found in the corpus. Otherwise, it uses the response from the neural conversation model. This architecture is not seen in other systems.

6 Experiments

6.1 Conditions

We evaluated our proposed system with the DSTC6 Twitter dialog task. Training, development and test sets were collected from Twitter sites related to customer services. Table 1 shows the size of each data set.

In order to be able to predict responses occurring partway through a dialog, we expanded the training and development sets by truncating each dialog after each system

Table 1: Twitter data

		train	dev.	test
#dialog		888,201	107,506	2,000
#turn		2,157,389	262,228	5,266
#word		40,073,697	4,900,743	99,389
#dialog	(expanded)	1,043,640	126,643	-
#turn	(expanded)	2,592,255	317,146	-
#word	(expanded)	50,106,092	6,182,080	-

Table 2: Model size

	encoder			decoder	
	#layer	#sent-layer	#cell	#layer	#cell
LSTM	2	-	128	2	128
BLSTM	2	-	128	2	256
HRED	2	1	128	2	128

response, and adding the truncated dialogs to the data sets. In each dialog, all turns except the last response were concatenated into one sequence to form input sequence X, with meta symbols `<U>` and `<S>` inserted at the beginning of each turn to explicitly utilize turn switching information. The last response was used as output sequence Y.

We built three types of models, LSTM, BLSTM, and HRED for response generation using the expanded training set. We employed an ADAM optimizer [31] with the cross-entropy criterion and iterated the training process up to 20 epochs. For each of the encoder-decoder model types, we selected the model with the lowest perplexity on the expanded development set. We also decided the model size based on the BLEU score for the development set, which resulted in Table 2.

We further applied adversarial training for each model, where we built a discriminator as an LSTM-based sequence classifier, which takes input sequence $\{X, Y\}$ and returns probability $P_D(+1|\{X, Y\})$. We applied a linear layer on top of the final hidden state of the LSTM, and the single output value is converted to the probability using a sigmoid function. The discriminator had two layers and 128 hidden units (cells) in each layer. After pretraining, one generative model update and five discriminator updates were alternately performed as in [18]. In preliminary experiments, adversarial training was unstable for LSTM encoder decoder and HRED with the LSTM discriminator. We only show the results for the BLSTM encoder decoder.

For example-based response selection, we trained a word2vec model using the expanded training set. The dimension of word vectors was 200. The similarity threshold to use the examples instead of the model-based responses was set to 0.9. We chose this threshold so that the BLEU score did not degrade on the development set.

For system combination, we combined three system outputs from the LSTM, BLSTM, and HRED networks. Each system generated 20-best results. The networks we used here were trained with the cross entropy criterion. We did not use the networks trained

Table 3: Evaluation results with word-overlap based metrics for 11 references.

Methods	BLEU4	METEOR	ROUGE_L	CIDEr
Baseline	0.1619	0.2041	0.3598	0.0825
LSTM	0.2166	0.2147	0.3928	0.1069
BLSTM	0.2051	0.2139	0.3876	0.1077
HRED	0.1978	0.2106	0.3892	0.1035
3-System Combination	**0.2205**	**0.2210**	**0.4102**	**0.1279**
LSTM+EG	0.2118	0.2140	0.3953	0.1060
BLSTM/ADV	0.1532	0.1833	0.3469	0.0800
BLSTM/ADV+EG	0.1504	0.1826	0.3446	0.0803
BLSTM/ADV+CSR+EG	0.1851	0.2040	0.3748	0.0965

with the adversarial method or the example-based method, since the aim of system combination was to improve objective scores.

6.2 Results submitted to DSTC6

The evaluation results of our models, training and decoding methods are summarized in Tables 3, 4 and 5. Table 3 shows the objective scores measured by word-overlap based metrics, BLEU4, METEOR, ROUGE_L, and CIDEr, while Table 4 shows the objective scores measured by word-embedding based metrics, SkipThoughts Cosine Similarity, Embedding Average Cosine Similarity, Vector Extrema Cosine Similarity, and Greedy Matching scores. We used `nlg-eval`[1] [32] to compute the objective scores. We prepared 11 references consisted of one true response and human-generated 10 responses for each dialog context. All the references were provided by the challenge organizers. The word-embedding-based scores were computed using the embedding models trained with the BookCorpus dataset [33]. Table 5 shows the subjective evaluation results based on human rating conducted by the challenge organizers, where each response was rated by 10 human subjects given the dialog context.

Since the evaluation was done using a crowd-sourcing service, Amazon Mechanical Turk (AMT), the 10 human subjects (Turkers) could be different for each response. The human ratings were collected for each system response and the reference using 5 point Likert scale, where the subjects rated each response by 5 level scores, *Very good* (5), *Good* (4), *Acceptable* (3), *Poor* (2), and *Very poor* (1). Before rating, the subjects were instructed to consider naturalness, informativeness, and appropriateness of the response for the given context. The details are described in [8].

The baseline results were obtained with an LSTM-based encoder decoder in [34], but this is a simplified version of [6], in which back-propagation is performed only up to the previous turn from the current turn, although the state information is taken over throughout the dialog. We used the default parameters, i.e., #layer=2 and #cell=512 for the baseline system. 'EG' and 'ADV' denote example-based response selection and

[1] `https://github.com/Maluuba/nlg-eval`

Table 4: Evaluation results with embedding based metrics for 11 references.

Methods	Skip Thought	Embedding Average	Vector Extrema	Greedy Matching
Baseline	0.6380	0.9132	0.6073	0.7590
LSTM	0.6824	0.9187	0.6343	0.7719
BLSTM	0.6757	0.9185	0.6268	0.7700
HRED	0.6859	0.9221	0.6315	0.7729
3-System Combination	0.6636	0.9251	**0.6449**	**0.7802**
LSTM+EG	**0.7075**	**0.9271**	0.6371	0.7747
BLSTM/ADV	0.6463	0.9077	0.5999	0.7544
BLSTM/ADV+EG	0.6451	0.9070	0.5990	0.7534
BLSTM/ADV+CSR+EG	0.6706	0.9116	0.6155	0.7613

Table 5: Evaluation results with 5-level human ratings.

Methods	Human Rating
Baseline	3.3638
3-System Combination	3.4332
LSTM+EG	3.3894
BLSTM/ADV	3.4381
BLSTM/ADV+EG	3.4453
BLSTM/ADV+CSR+EG	**3.4777**
Reference	3.7245

adversarial training. ‘CSR’ means we used the cosine similarity reward in addition to the discriminator scores as in Eq. (24).

The results demonstrate substantial improvement by using system combination in most objective measures, where we used the BLEU1 metric for MBR decoding[2]. These objective scores were also better than other systems’ scores officially submitted to DSTC6.

On the other hand, such objective scores degraded slightly by example-based response selection and significantly by adversarial training. Since our aim of using these techniques was to improve the subjective measure rather than the objective measures, we expected these results to some extent. However, if we add the cosine similarity to the reward function, we can mitigate the degradation of objective scores by adversarial training.

Regarding the subjective evaluation, as we expected, the example-based response selection and adversarial training improved the human rating score. Table 5 shows a slight improvement from ‘BLSTM/ADV’ (3.4381) to ‘BLSTM /AVG+EG’ (3.4453) by the example-based method and a further gain by adding the cosine similarity reward for ad-

[2] We conducted preliminary experiments to compare BLEU1 to BLEU4, METEOR, and Embedding Average Cosine Similarity for MBR decoding, and BLEU1 was the best for most objective measures for the development set.

versarial training, which achieved the best performance with BLSTM/ADV+CSR+EG (3.4777) using our official DSTC6 system.

Although we should also investigate the impact of CSR without EG, it is difficult to do the experiments again with the same human subjects. However, the impact of EG is very limited, because we used 0.9 as the threshold for the similarity, and the examples were used only for 6% of dialogs in the test set. Therefore, we think the comparison is still effective. We set the threshold as small as possible within the range where the objective scores do not decrease.

System combination also improved the human rating score, which can be observed by comparing the scores of 'LSTM+EG' (3.3894) and '3-System Combination' (3.4332) even though the models for system combination were not trained by adversarial training or combined with the example-based method. Accordingly, an approach to further improve the human rating score is to combine the responses from different adversarial models. The next section will investigate this possibility.

6.3 System combination results on adversarial models

We further conducted system combination of LSTM, BLSTM, and HRED trained with the adversarial method. In adversarial training, a LSTM-based discriminator consistently used, but it was downsized to the half, i.e., 64 cells for each layer in LSTM and HRED since the training procedure was unstable when we used the 128 cells. This could be because the discriminator was too strong against the generative models, which had almost the same complexity as the generative ones. The LSTM, BLSTM and HRED models were retrained with the adversarial plus cosine similarity objective in Eq. (24).

Tables 6 and 7 show objective scores based on word-overlapping and word-embedding metrics. New results on LSTM indicate that adversarial training does not necessarily degrade the objective scores as comparing the results of 'LSTM+EG' with those of 'LSTM/ADV+CSR+EG'. This could be due to the regularization effect by the similarity term. Similar to the results in Tables 3 and 4, system combination of three adversarial models yielded a certain gain even for the objective measures. For reference, we also show the objective scores of [25]'s system, which achieved the best human rating score in DSTC6, and the best score of the other systems in each metric [8]. Since [25]'s system was designed to improve subjective scores, it did not provide high objective scores.

Table 8 shows human rating scores for the baseline and system combination, where the system combination of adversarially trained models improved the rating score as well as the objective scores. We also compare our system with the best system in terms of human rating score in the official evaluation of DSTC6 [25]. Finally, our system achieved a higher score than that of the best system. Note that since we conducted the subjective evaluation with different human raters for these new experiments, the scores are not the same as those reported in the official results [8].

Table 6: Evaluation results on adversarial system combination with word-overlap based metrics for 11 references.

Methods	BLEU4	METEOR	ROUGE_L	CIDEr
Baseline	0.1619	0.2041	0.3598	0.0825
3-System Combination	**0.2205**	**0.2210**	**0.4102**	**0.1279**
LSTM+EG	0.2118	0.2140	0.3953	0.1060
LSTM/ADV+CSR+EG*	0.2190	0.2026	0.3888	0.1187
BLSTM/ADV+CSR+EG	0.1851	0.2040	0.3748	0.0965
HRED/ADV+CSR+EG*	0.1821	0.2003	0.3794	0.0945
3-System Comb./ADV+CSR+EG*	0.2199	0.2207	0.3982	0.1204
[25]	0.1575	0.1918	0.3658	0.1112
Best of other systems	0.1779	0.2085	0.3829	0.1112

*New results after DSTC6 workshop.

Table 7: Evaluation results on adversarial system combination with embedding based metrics for 11 references.

Methods	Skip Thought	Embedding Average	Vector Extrema	Greedy Matching
Baseline	0.6380	0.9132	0.6073	0.7590
3-System Combination	0.6636	0.9251	**0.6449**	**0.7802**
LSTM+EG	**0.7075**	0.9271	0.6371	0.7747
LSTM/ADV+CSR+EG*	0.6782	0.9043	0.6363	0.7673
BLSTM/ADV+CSR+EG	0.6706	0.9116	0.6155	0.7613
HRED/ADV+CSR+EG*	0.6815	0.9211	0.6355	0.7681
3-System Comb./ADV+CSR+EG*	0.7046	**0.9285**	0.6433	0.7797
[25]	0.6457	0.9076	0.6075	0.7528
Best of other systems	0.6529	0.9132	0.6106	0.7683

*New results after DSTC6 workshop.

6.4 Verification of evaluation results

We conducted statistical tests to verify the objective and subjective evaluation results obtained in the above experiments. We used a two-sample z-test for the difference of objective scores and Welch's t-test for the difference of human rating scores.

Table 9 compares BLEU4 and METEOR scores before and after adding the cosine similarity reward (CSR) in adversarial training. p-values for the difference of the scores were nearly equal to zero. This means that the improvements are statistically significant, and demonstrates the effectiveness of CSR to retain high objective scores in adversarial training. We also had significant improvements in the other objective measures as shown in Tables 3 and 4.

Table 10 shows human rating scores and p-values for the baseline and BLSTM/ADV systems. Compared with the baseline, adversarial training and CSR provided significant improvements with $p < 0.05$. We can also see that CSR further improved the human

Table 8: Evaluation results on adversarial system combination with 5-level human ratings. Note that rating scores do not match those in Table 5 because human raters are different from those who performed the rating for the official DSTC6 evaluation.

Methods	Human Rating
Baseline	3.2145
3-System Combination	3.3169
3-System Comb./ADV+CSR+EG*	**3.5126**
[25]	3.4871
Reference	3.6929

*New results after DSTC6 workshop.

Table 9: Statistical test p-values on the BLEU4 and METEOR improvements from the BLSTM/ADV+EG system. Bold numbers indicate $p < 0.05$.

Methods	BLEU4	p-value	METEOR	p-value
BLSTM/ADV+EG	0.1504	-	0.1826	-
BLSTM/ADV+CSR+EG	0.1851	$\approx \mathbf{0}$	0.2040	$\approx \mathbf{0}$

rating score although the difference is not significant ($p = 0.097$).

Table 11 verifies the efficacy of system combination in objective measures (BLEU4 and METEOR). We can see that the system combination yielded a significant improvement ($p = 0.0354$ for BLEU4 and $p = 0.0007$ for METEOR) from our best single system (LSTM). We also confirmed that system combination of adversarially trained models had high objective scores competitive to or better than the best single system ($p = 0.0749$ for BLEU4 and $p = 0.0012$ for METEOR).

Furthermore, we show p-values on human rating scores of system combination in Table 12. According to the p-values, our final system is significantly better than the original system combination, although the score gain from [25] is not statistically significant ($p = 0.2152 > 0.05$). Accordingly, we can say that our final system achieved top-level performance in both subjective and objective evaluation metrics for this neural conversation task.

6.5 Response examples

We show some examples of responses that our system generated. The first example in Table 13 shows the impact of adversarial training and the cosine similarity-based reward (CSR). The LSTM-based model generated irrelevant phrase "`please dm us your order number`", which is observed very frequently in the training data but it is inappropriate in this context. When using adversarial training (BLSTM/ADV+EG), the response became more relevant to the context[3]. Although the human rating score increased from 2.8 to 4.0, the BLEU4 score degraded from 0.5 to 0.22. But it was

[3] Since the BLSTM without adversarial training generated the same response as the LSTM for this context, the performance difference would not cause the difference of model architecture.

Table 10: Statistical test p-values on human rating scores in Table 5. The p-values were computed on the improvements from the baseline and BLSTM/ADV. Bold numbers indicate $p < 0.05$.

Methods	Human Rating	p-value for baseline	p-value for BLSTM/ADV
LSTM+EG	3.3638	0.3195	-
BLSTM/ADV	3.4381	**0.0028**	-
BLSTM/ADV+EG	3.4453	**0.0011**	0.7636
BLSTM/ADV+CSR+EG	3.4777	$\approx \mathbf{0}$	0.0970

Table 11: Statistical test p-values on the BLEU4 and METEOR improvements from the best single system. Bold numbers indicate $p < 0.05$.

Methods	BLEU4	p-value	METEOR	p-value
Best single system (LSTM)	0.2166	-	0.2147	-
3-System Combination	0.2205	**0.0354**	0.2210	**0.0007**
3-System Comb./ADV+CSR+EG	0.2199	0.0749	0.2207	**0.0012**

recovered to 0.51 by adding the CSR, where the responses with and without CSR are similar, but more common phrases are used by CSR. The adversarial training with CSR provided higher scores in the both metrics.

The second example in Table 14 shows similar effects on the response quality. BLSTM/ADV+EG generated a fancy response with "♡" marks, while BLSTM/ADV+CSR+EG generated more conservative response even though their meanings are not different. Adversarial training actually increases this kind of responses, but CSR can usually suppress them.

The third example in Table 15 shows the impact of system combination. With system combination, the objective score, e.g. BLEU4, can be improved. But it is difficult to correct irrelevant responses if all the models generated irrelevant responses that are similar to each other. In Table 15, the response of 3-System Combination is almost the same as that of LSTM+EG although the BLEU4 score slightly improved. Finally, system combination of adversarially trained models, i.e., 3-System Combination/ADV+CSR+EG, generated a better response with 3.4 human rating score while retaining the BLEU4 score of standard system combination.

7 Conclusion

We proposed a neural conversation system for the 6th Dialog System Technology Challenge (DSTC6). Our system consists of several conversation models such as a long short-term memory (LSTM), a bidirectional LSTM (BLSTM) encoder decoder, and hierarchical recurrent encoder decoder (HRED), advanced training modules such as sequence adversarial training, and decoding techniques based on system combination by minimum Bayes Risk (MBR) decoding and example-based response selection. In our experimental results on a Twitter help-desk dialog task, adversarial training and

Table 12: Statistical test p-values on human rating scores of system combination in Table 8.

Methods	Human Rating	p-value for 3-System Combination	p-value for [25]
3-System Combination	3.3169	-	-
3-System Comb./ADV+CSR+EG	3.5126	$\approx \mathbf{0}$	0.2152
[25]	3.4871	$\approx \mathbf{0}$	-

Table 13: Example responses (1).

Context	U: starting our holiday with a 3hr delay isn 't gr8 .. hope you make sure there are some refreshments at barcelo fuertaventura		
Systems	Response	BLEU4	Human Rating
LSTM+EG	hi <USER> , i 'm sorry to hear this . please dm us your order number so we can look into this for you . thank you .	0.50	2.8
BLSTM /ADV+EG	sorry to hear about this <USER> , can you dm us with further details so that we can look into this for you ?	0.22	4.0
BLSTM /ADV+CSR+EG	hi <USER> , i 'm sorry to hear this . can you please dm us your details so we can look into this for you ?	0.51	3.9
Reference	hi <USER> , i am sorry to hear this . our team are hard at work to ensure this departs as quickly as possible	1.00	4.0

example-based response selection improved human rating score while system combination with MBR decoding improved both objective and subjective measures including BLEU, METEOR and human rating scores. Furthermore, our system yielded the best scores in most objective evaluation metrics and a better score than that of the best system in terms of human rating evaluation. Although the difference of human rating scores between the best system and our final system was not statistically significant, we demonstrated that our final system achieved top-level performance in both subjective and objective evaluation metrics by applying the proposed techniques.

References

[1] M. F. McTear, Spoken dialogue technology: enabling the conversational user interface, ACM Computing Surveys (CSUR) 34 (1) (2002) 90–169.

[2] S. J. Young, Probabilistic methods in spoken–dialogue systems, Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 358 (1769) (2000) 1389–1402.

Table 14: Example responses (2).

Context	U: happy birthday ! ! ! it 's her first time at the q ! ! ! @monstershockey @theqassist @theqarena <URL>		
Systems	Response	BLEU4	Human Rating
LSTM+EG	thank you !	0.23	3.8
BLSTM /ADV+EG	happy birthday , <USER> ! ! ! ! ! ! ! ♡ ♡ ♡ ♡ ♡ ♡ ♡ ♡ ♡ ♡	0.04	4.2
BLSTM /ADV+CSR+EG	happy birthday !	0.43	4.7
Reference	saw your tweet - - we 'd love to give her a commemorative birthday lanyard . stop by guest services at section 100 .	1.00	4.1

[3] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, L. Hetherington, Juplter: a telephone-based conversational interface for weather information, IEEE Transactions on speech and audio processing 8 (1) (2000) 85–96.

[4] D. Jurafsky, J. H. Martin, Speech & Language Processing, Pearson Education, 2000.

[5] R. De Mori, Spoken language understanding: a survey., in: ASRU2007, 2007, pp. 365–376.

[6] O. Vinyals, Q. Le, A neural conversational model, arXiv preprint arXiv:1506.05869.

[7] R. Lowe, N. Pow, I. Serban, J. Pineau, The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems, arXiv preprint arXiv:1506.08909.

[8] C. Hori, J. Perez, R. Higasinaka, T. Hori, Y.-L. Boureau, M. Inaba, Y. Tsunomori, T. Takahashi, K. Yoshio, S. Kim, Overview of the sixth dialog system technology challenge: DSTC6, submitted to Computer Speech & Language.

[9] C. Hori, T. Hori, End-to-end conversation modeling track in DSTC6, arXiv preprint arXiv:1706.07440.

[10] J. Williams, A. Raux, D. Ramachandran, A. Black, The dialog state tracking challenge, in: Proceedings of the SIGDIAL 2013 Conference, 2013, pp. 404–413.

[11] T. Hori, H. Wang, C. Hori, S. Watanabe, B. Harsham, J. Le Roux, J. R. Hershey, Y. Koji, Y. Jing, Z. Zhu, et al., Dialog state tracking with attention-based sequence-to-sequence learning, in: Spoken Language Technology Workshop (SLT), 2016 IEEE, IEEE, 2016, pp. 552–558.

Table 15: Example responses (3).

Context	U: when is @ikeacanada opening in dartmouth crossing ? ! ? ! anyone know ? ? #shopping @peady @kerra_aucoin @adolomount @imashleymi		
Systems	Response	BLEU4	Human Rating
LSTM+EG	hi <USER> , we 're sorry to hear this . please dm us your contact info so we can look into this for you . thank you .	0.11	1.7
3-System Combination	hi <USER> , we 're sorry to hear this . we 'll be sure to share your feedback with our team . thanks	0.15	1.9
3-System Combination /ADV+CSR+EG	hi <USER> , you can find out more here : <URL>	0.15	3.4
Reference	thanks for asking <USER> . we anticipate the ikea store to open sometime in late fall 2017 . we hope this helps !	1.00	4.2

[12] W. Wang, Y. Koji, B. Harsham, J. R. Hershey, Sequence adversarial training and minimum bayes risk decoding for end-to-end neural conversation models, in: Proceedings of the 6th Dialog System Technology Challenges (DSTC6) Workshop, 2017.

[13] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, J. Pineau, Building end-to-end dialogue systems using generative hierarchical neural network models., in: AAAI, 2016, pp. 3776–3784.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[15] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: Sequence generative adversarial nets with policy gradient., in: AAAI, 2017, pp. 2852–2858.

[16] Z. Yang, W. Chen, F. Wang, B. Xu, Improving neural machine translation with conditional sequence generative adversarial nets, arXiv preprint arXiv:1703.04887.

[17] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, B. Schiele, Speaking the same language: Matching machine to human captions by adversarial training, arXiv preprint arXiv:1703.10476.

[18] J. Li, W. Monroe, T. Shi, A. Ritter, D. Jurafsky, Adversarial learning for neural dialogue generation, arXiv preprint arXiv:1701.06547.

[19] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning 8 (3-4) (1992) 229–256.

[20] A. Stolcke, Y. Konig, M. Weintraub, Explicit word error minimization in n-best list rescoring., in: Eurospeech, Vol. 97, 1997, pp. 163–166.

[21] S. Kumar, W. Byrne, Minimum bayes-risk decoding for statistical machine translation, Tech. rep., JOHNS HOPKINS UNIV BALTIMORE MD CENTER FOR LANGUAGE AND SPEECH PROCESSING (CLSP) (2004).

[22] K. Papineni, S. Roukos, T. Ward, W. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA., 2002, pp. 311–318.
URL `http://www.aclweb.org/anthology/P02-1040.pdf`

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[24] J. Li, M. Galley, C. Brockett, J. Gao, B. Dolan, A diversity-promoting objective function for neural conversation models, arXiv preprint arXiv:1510.03055.

[25] M. Galley, C. Brockett, The MSR-NLP system at dialog system technology challenges 6, in: Proceedings of the 6th Dialog System Technology Challenges (DSTC6) Workshop, 2017.

[26] J. G. Fiscus, A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover), in: Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on, IEEE, 1997, pp. 347–354.

[27] G. Evermann, P. Woodland, Posterior probability decoding, confidence estimation and system combination, in: Proc. Speech Transcription Workshop, Vol. 27, Baltimore, 2000, p. 78.

[28] K. C. Sim, W. J. Byrne, M. J. Gales, H. Sahbi, P. C. Woodland, Consensus network decoding for statistical machine translation system combination, in: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, Vol. 4, IEEE, 2007, pp. IV–105.

[29] H. Murao, N. Kawaguchi, S. Matsubara, Y. Yamaguchi, Y. Inagaki, Example-based spoken dialogue system using WOZ system log, in: Proceedings of the Fourth SIGdial Workshop of Discourse and Dialogue, 2003.

[30] C. Lee, S. Jung, S. Kim, G. G. Lee, Example-based dialog modeling for practical multi-domain dialog system, Speech Communication 51 (5) (2009) 466–484.

[31] D. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[32] S. Sharma, L. El Asri, H. Schulz, J. Zumer, Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation, CoRR abs/1706.09799.
URL http://arxiv.org/abs/1706.09799

[33] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 19–27.

[34] T. Hori, DSTC6 end-to-end conversation modeling track: tools and baseline system, https://github.com/dialogtekgeek/DSTC6-End-to-End-Conversation-Modeling (2017).