Sequence Adversarial Training and Minimum Bayes Risk Decoding for End-to-end Neural Conversation Models

Wen Wang[1], *Yusuke Koji*[1], *Bret Harsham*[2], *Takaaki Hori*[2], *John R. Hershey*[2]

[1]Information Technology R&D Center, Mitsubishi Electric Corporation
[2]Mitsubishi Electric Research Laboratories (MERL)

{Wang.Wen@ds, Koji.Yusuke@bx}.MitsubishiElectric.co.jp, {harsham, thori, hershey}@merl.com

Abstract

We present a neural conversation system that incorporates multiple sequence-to-sequence models, sequence adversarial training, example-based response selection, and BLEU-based Minimum Bayes Risk (MBR) decoding. The system was trained and tested using the 6th Dialog System Technology Challenges (DSTC6) Twitter help-desk dialog task. Experimental results demonstrate that adversarial training and the example-based method are effective in improving human rating score while system combination with MBR decoding improves objective measures such as BLEU and METEOR scores. Moreover, we investigate extension of the reward function for sequence adversarial training in order to balance subjective and objective scores.

Index Terms: 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 proposed system has several key features including sequence adversarial training, example-based response selection, multiple sequence-to-sequence models, and minimum Bayes risk (MBR) decoding, where the multiple models are a long short-term memory (LSTM) encoder decoder, a bidirectional LSTM (BLSTM) encoder decoder, and a hierarchical recurrent encoder decoder (HRED). A system combination is performed to combine the multiple hypotheses from these models to improve BLEU score. Sequence adversarial training and the example-based method are used to obtain a high human rating score. Experimental results on the Twitter help-desk dialog task show that adversarial training and the example-based method are effective in improving human rating score while system combination improves objective measures such as BLEU and METEOR scores (We might change this part after getting human rating results). Furthermore, we investigate extension of reward functions for sequence adversarial training to balance subjective and objective scores.

2. End-to-end Conversation Modeling

This section explains the neural conversation model of [6], which is designed as a sequence-to-sequence mapping process 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 corresponds to the sequence of all previous sentences in a conversation, and Y is the system response sentence we want to generate. In our model, both X and Y are sequences of words. 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 most likely hypothesis of Y is obtained as

$$\hat{Y} = \arg\max_{Y \in \mathcal{V}^*} P(Y|X) \tag{1}$$

$$= \arg\max_{Y \in \mathcal{V}^*} \prod_{m=1}^{M} P(y_m|y_1, \ldots, y_{m-1}, X), \tag{2}$$

where $\mathcal{V}^*$ denotes a set of sequences of zero or more words in system vocabulary $\mathcal{V}$.

Let X be word sequence $x_1, \ldots, x_T$ and Y be word sequence $y_1, \ldots, y_M$. The encoder network is used to obtain hidden states h_t for $t = 1, \ldots, T$ as:

$$h_t = \text{LSTM}\left(x_t, h_{t-1}; \theta_{enc}\right), \tag{3}$$

where h_0 is initialized with a zero vector. LSTM$(\cdot)$ is a LSTM function with parameter set θ_{enc}.

The decoder network is used to compute probabilities $P(y_m|y_1, \ldots, y_{m-1}, X)$ for $m = 1, \ldots, M$ as:

$$s_0 = h_T \tag{4}$$

$$s_m = \text{LSTM}\left(y_{m-1}, s_{m-1}; \theta_{dec}\right) \tag{5}$$

$$P(\mathbf{y}|y_1, \ldots, y_{m-1}, X) = \text{softmax}(W_o s_m + b_o), \tag{6}$$

where y_0 is set to `<eos>`, a special symbol representing the end of sequence. s_m is the m-th decoder state. θ_{dec} is a set of decoder parameters, and W_o and b_o are a matrix and a vector. In this model, the initial decoder state s_0 is given by the final encoder state h_T as in Eq. (4), and the probability is estimated from each state s_m. To efficiently find $\hat{Y}$ in Eq. (2), we use a beam search technique since it is computationally intractable to consider all possible Y.

3. The MELCO/MERL System

Figure 1 shows the architecture of our DSTC6 end-to-end conversation system. In the training phase, the upper part of the figure, a sequence-to-sequence model is trained with the Cross-Entropy (CE) criterion using the training corpus, where the model can be LSTM, BLSTM or HRED. Furthermore, sequence adversarial training is optionally performed for the model to generate better sentences.

In the generation phase (the lower part of figure), we employ model-based sentence generation and example-based response selection. The generated or example-based responses are selected based on a threshold. We also apply a system combination technique to enhance the response sentence by combining multiple hypotheses generated by different models. We describe each module in the following subsections.

3.1. Conversation models

We employ three types of sequence-to-sequence models. Figure 2 (a) shows a LSTM encoder decoder described in Section 2. Figure 2 (b) shows a BLSTM encoder decoder, where the encoder has bidirectional layers. The last hidden and cell vectors of the forward layer and the first hidden and cell vectors of the backward layer are concatenated and fed to the LSTM decoder. Figure 2 (c) shows a HRED [12], which has a hierarchical structure of word-level and sentence-level propagation processes. In the word-level layer of the encoder, a sentence embedding vector is obtained at each sentence end, which is 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, which are depicted as dashed lines in the figure. An HRED model can capture sentence-level state transitions in the dialog, which is potentially effective to predict the next response when it has longer contextual information.

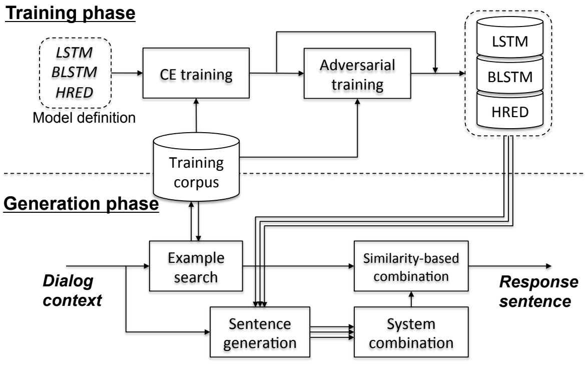
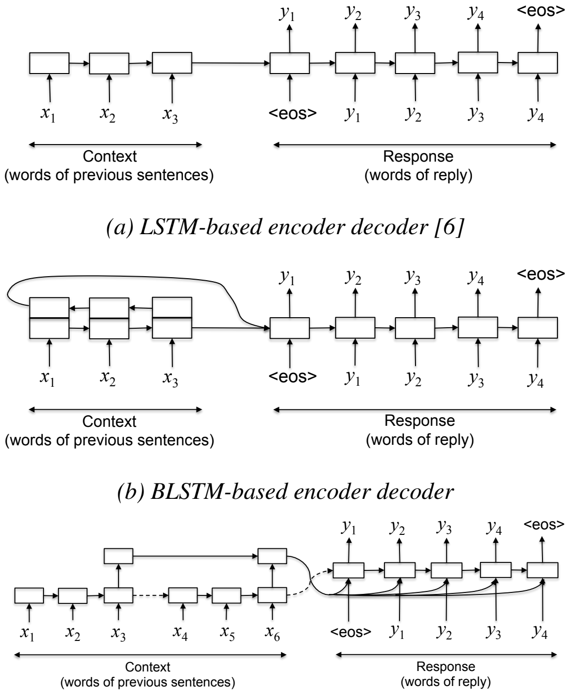


Figure 1: *The MELCO/MERL system*



(a) LSTM-based encoder decoder [6]

(b) BLSTM-based encoder decoder

(c) Hierarchical recurrent encoder decoder (HRED) [12]

Figure 2: *Sequence-to-sequence models*

3.2. Sequence adversarial training

We apply an adversarial training scheme [13] to our 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 correctly classify system generated sentences 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 [14], machine translation [15], image captioning [16], and open-domain dialog generation [17].

To train our models, we use a policy gradient optimization based on the reinforce algorithm [18]. 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

$$J(\theta) = E_{Y\sim P_G(Y|X;\theta)}[P_D(+1|\{X,Y\})], \tag{7}$$

and its gradient is computed as

$$\begin{aligned}\nabla J(\theta) \approx & [P_D(+1|\{X,Y\}) - b(\{X,Y\})] \\ & \nabla \sum_t \log P_G(y_t|X, y_1, \ldots, y_{t-1};\theta),\end{aligned} \tag{8}$$

where θ is the set of parameters of the generative model, $P_G(Y|X;\theta)$ is the probability distribution on Y given X, 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. $b(\{X,Y\})$ is the baseline value [18]. 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 [17].

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

$$J(\theta) = E_{Y\sim P_G(Y|X;\theta)}[P_D(+1|\{X,Y\}) + \lambda \mathrm{Sim}(Y,Y')], \tag{9}$$

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 3.3.

3.3. Example-based response selection

We use an example-based method to generate system response when we find a similar context to the input in a training corpus. Suppose dialogs in the training corpus are represented as following format:

$$(X_i', Y_i'), i = 1, ..., N \tag{10}$$

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 $Y_{\hat{i}}'$ corresponding to the highest similarity is regarded as system output $\hat{Y}$, i.e.,

$$\hat{Y} = Y_{\hat{i}}' \tag{11}$$

$$\hat{i} = \arg \max_{i=1,\ldots,N} \mathrm{Sim}(X, X_i'). \tag{12}$$

When computing the similarity, word vectors obtained by word2vec [19] is applied to feature extraction. Firstly a training corpus is used to obtain a word2vec model. Secondly, vector of each word in input sentences is summed as final feature vector.

Example-based response selection is combined with other sentence generation methods as shown in Figure 2. If the similarity score of the best sentence(s) from the example-based method is larger than a predefined threshold, example-base sentences will be used as the final system response output, otherwise generated sentences will be used.

3.4. 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. Although system combination has previously been applied to speech recognition [20, 21] and machine translation [22], it has not yet been used for dialog response generation (to the best of our knowledge).

To perform system combination, we apply a minimum Bayes-risk (MBR) decoding [23, 24], which can improve the sentence quality by focusing on a specific evaluation metric. Here we use BLEU score [25].

In MBR decoding, the decoding objective is defined as

$$\hat{Y} = \arg \max_{Y\in\mathcal{V}^*} \sum_{Y'\in\mathcal{V}^*} P(Y'|X)E(Y',Y), \tag{13}$$

where $E(Y',Y)$ denotes an evaluation metric assuming Y' is a reference (ground-truth) and Y is a hypothesis (generated description). For the BLEU [25] score, the evaluation metric can be computed as

$$E(Y',Y) = \exp\left(\sum_{n=1}^{N} \log \frac{p_n(Y',Y)}{N}\right) \times \gamma(Y',Y), \tag{14}$$

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'.

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 the system. Although in theory the distribution $P(Y'|X)$ should be the true distribution, we instead estimate it using the encoder-decoder model.

4. Experiments

4.1. 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 2 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 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.

Table 1: *Evaluation results with objective measures based on 11 references and a subjective measure based on 5-level ratings*

Methods	BLEU4	METEOR	ROUGE_L	CIDEr	Skip Thought	Embedding Average	Vector Extrema	Greedy Matching	Human Rating
Baseline [26]	0.1619	0.2041	0.3598	0.0825	0.6380	0.9132	0.6073	0.7590	3.3638
LSTM	0.2166	0.2147	0.3928	0.1069	0.6824	0.9187	0.6343	0.7719	-
BLSTM	0.2051	0.2139	0.3876	0.1077	0.6757	0.9185	0.6268	0.7700	-
HRED	0.1978	0.2106	0.3892	0.1035	0.6859	0.9221	0.6315	0.7729	-
3-System Combination	**0.2205**	**0.2210**	**0.4102**	**0.1279**	0.6636	0.9251	**0.6449**	**0.7802**	3.4332
LSTM+EG	0.2118	0.2140	0.3953	0.1060	**0.7075**	**0.9271**	0.6371	0.7747	3.3894
BLSTM/ADV	0.1532	0.1833	0.3469	0.0800	0.6463	0.9077	0.5999	0.7544	3.4381
BLSTM/ADV+EG	0.1504	0.1826	0.3446	0.0803	0.6451	0.9070	0.5990	0.7534	3.4453
BLSTM/ADV+CS+EG	0.1851	0.2040	0.3748	0.0965	0.6706	0.9116	0.6155	0.7613	**3.4777**

Table 2: *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 3: *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

We built three types of models, LSTM, BLSTM, and HRED for response generation using the expanded training set. We employed an ADAM optimizer [27] 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 3.

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 $P_D(+|\{X, Y\})$. 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 [17]. 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 models. Each system generated 20-best results. The models we used here were trained with the cross entropy criterion. We did not use the models trained with the adversarial method or the example-based method, since the aim of system combination was to improve objective scores.

4.2. Results

Table 1 shows the performance of our models, training and decoding methods using objective measures, BLEU4, METEOR, ROUGE_L, CIDEr, SkipThought Cosine Similarity, Embedding Average Cosine Similarity, Vector Extrema Cosine Similarity, and Greedy Matching scores, which were computed with `nlg-eval`[1] [28]. The table also includes subjective evaluation results based on human rating conducted by the challenge organizer, where each response was rated with score 1 to 5 by 10 human subjects given the dialog context, and the average score for each system is shown in the table.

The baseline results were obtained with an LSTM-based encoder decoder in [26], 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 #cells=512 for the baseline system. 'EG' and 'ADV' denote example-based response selection and adversarial training. 'CSR' means we used the cosine similarity reward in addition to the discriminator scores as in Eq. (9).

The results show some improvement using system combination in most objective measures, although we used the BLEU metric for MBR decoding. 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. 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. Finally, the objective function based on adversarial training and the cosine similarity achieved the best human rating score 3.4777.

5. Conclusion

We proposed a neural conversation system for the 6th Dialog System Technology Challenge (DSTC6). In our experimental results on a Twitter help-desk dialog task, adversarial training and example-based response selection improved human rating score while system combination with MBR decoding improved objective measures such as BLEU and METEOR scores.

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

6. References

[1] M. F. McTear, "Spoken dialogue technology: enabling the conversational user interface," *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 90–169, 2002.

[2] S. J. Young, "Probabilistic methods in spoken–dialogue systems," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 358, no. 1769, pp. 1389–1402, 2000.

[3] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington, "Juplter: a telephone-based conversational interface for weather information," *IEEE Transactions on speech and audio processing*, vol. 8, no. 1, pp. 85–96, 2000.

[4] D. Jurafsky and 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 and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[7] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems," *arXiv preprint arXiv:1506.08909*, 2015.

[8] "DSTC6 Dialog System Technology Challenges," http://workshop.colips.org/dstc6/, accessed: 2017-06-01.

[9] C. Hori and T. Hori, "End-to-end conversation modeling track in dstc6," *arXiv preprint arXiv:1706.07440*, 2017.

[10] J. Williams, A. Raux, D. Ramachandran, and 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.

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

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

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

[15] Z. Yang, W. Chen, F. Wang, and B. Xu, "Improving neural machine translation with conditional sequence generative adversarial nets," *arXiv preprint arXiv:1703.04887*, 2017.

[16] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, and B. Schiele, "Speaking the same language: Matching machine to human captions by adversarial training," *arXiv preprint arXiv:1703.10476*, 2017.

[17] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *arXiv preprint arXiv:1701.06547*, 2017.

[18] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

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

[20] 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.

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

[22] K. C. Sim, W. J. Byrne, M. J. Gales, H. Sahbi, and 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.

[23] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in n-best list rescoring." in *Eurospeech*, vol. 97, 1997, pp. 163–166.

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

[25] K. Papineni, S. Roukos, T. Ward, and 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. [Online]. Available: http://www.aclweb.org/anthology/P02-1040.pdf

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

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] S. Sharma, L. El Asri, H. Schulz, and J. Zumer, "Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation," *CoRR*, vol. abs/1706.09799, 2017. [Online]. Available: http://arxiv.org/abs/1706.09799