applied sciences

MDPI

Article

# Multi-Turn Chatbot Based on Query-Context Attentions and Dual Wasserstein Generative Adversarial Networks

**Jintae Kim [1], Shinhyeok Oh [1], Oh-Woog Kwon [2] and Harksoo Kim [1,*]**

[1] Program of Computer and Communications Engineering, Kangwon National University, Chuncheon 24341, Korea; wlsxo1119@kangwon.ac.kr (J.K.); osh7605@kangwon.ac.kr (S.O.)
[2] Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; ohwoog@etri.re.kr
* Correspondence: nlpdrkim@kangwon.ac.kr; Tel.: +82-33-250-6388

Received: 20 August 2019; Accepted: 16 September 2019; Published: 18 September 2019


**Abstract:** To generate proper responses to user queries, multi-turn chatbot models should selectively consider dialogue histories. However, previous chatbot models have simply concatenated or averaged vector representations of all previous utterances without considering contextual importance. To mitigate this problem, we propose a multi-turn chatbot model in which previous utterances participate in response generation using different weights. The proposed model calculates the contextual importance of previous utterances by using an attention mechanism. In addition, we propose a training method that uses two types of Wasserstein generative adversarial networks to improve the quality of responses. In experiments with the DailyDialog dataset, the proposed model outperformed the previous state-of-the-art models based on various performance measures.

**Keywords:** multi-turn chatbot; dialogue context encoding; WGAN-based response generation

## 1. Introduction

Chatbots are computer systems that feature natural conversation with people. Recently, generative chatbots that generate responses directly by the models have been developed with advances in deep learning. Based on the number of dialogue contexts that chatbots should consider to generate responses, chatbot models are divided into two categories: single-turn and multi-turn. Single-turn chatbots generate a response based on an immediately preceding utterance called a user's query (i.e., a user's utterance just before the response). Multi-turn chatbots generate a response based on multiple previous utterances, called a dialogue context, as well as a user's query. Table 1 shows the differences between the responses of single- and multi-turn chatbots.

**Table 1.** Examples of chitchat.

| ID | Speaker | Utterance |
|---|---|---|
| (1) | User | I like pork. |
| (2) | Chatbot | Me too. |
| (3) | User | So I like Chinese food. |
| (4) | Chatbot | I prefer Korean foods to Chinese food. |
| (5) | User | Why? |
| (6-1) | Chatbot-*SING* | No reason. |
| (6-2) | Chatbot-*MULT* | Korean food is healthier. |

In Table 1, Chatbot-*SING* and Chatbot-*MULT* are single- and multi-turn chatbots, respectively. As shown, Chatbot-*SING* generates the short and safe response, "No reason" because it cannot

look up any other previous utterances except the immediately preceding one, "Why?". Compared with Chatbot-*SING*, Chatbot-*MULT* generates the more context-aware response, "Korean foods are healthier.", because it can look up the full dialogue history. Although multi-turn chatbots are more natural and informative, implementing multi-turn chatbots is not easy because they must determine the previous utterances that are associated with a response and also the degree to which those previous utterances affect the generation of a response. To overcome this problem, we propose a multi-turn chatbot model in which previous utterances are effectively and differently considered to generate responses using an attention mechanism. In addition, the proposed model uses two types of generative adversarial network (GAN) architectures [1–4]: One maps a vector representation of a dialogue history to a vector representation of a response, and the other maps a vector representation of a generated response (i.e., a decoded response) to a vector representation of an original response (i.e., an encoded response). The first GAN plays the role of generating a response vector associated with a dialogue history, and the second GAN plays the role of generating a surface sentence (i.e., a sequence of words) to realize a response vector.

The remainder of this paper is organized as follows: In Section 2, we review the previous studies on generative chatbots, and in Section 3, we describe the proposed multi-turn chatbot model. In Section 4, we explain the experimental setup and report some of our experimental results. We provide a conclusion to our study in Section 5.

## 2. Previous Works

Most of the recent approaches on generative chatbots are primarily based on sequence-to-sequence (Seq2Seq) learning. Seq2Seq is a supervised learning algorithm in which the input and the generated output are each a sequence of words [5,6]. In general, Seq2Seq models consist of two recurrent neural networks (RNNs): An RNN for encoding inputs and an RNN for generating outputs. Previous studies have demonstrated that chatbots based on Seq2Seq models often respond with either a safe response problem (i.e., the problem returning short and general responses such as "Okay" and "I see") or a semantically erroneous response [7,8]. A variational auto-encoder (VAE) is a continuous latent variable model intended to learn a latent space using a given set of samples. The model consists of an encoder and decoder: The encoder maps inputs into latent variables, and the decoder generates outputs that are similar to the inputs based on the latent variables. As a result, VAEs represent high-level semantics of the responses and help chatbots to generate various responses [9,10]. However, VAE models tend to suffer from collapse problems, where the decoder learns to ignore the latent variable and simplifies the latent variable to a standard normal distribution [11,12]. This problem has been partially solved by learning latent variable space through adversarial learning [12]. In addition, various studies using GAN architecture have been conducted [1–4]. However, adversarial learning for discrete tokens is difficult because of non-differentiability [2,4]. To solve these problems, various studies have been conducted, including those on a hybrid model of a GAN and reinforcement learning [4,13]. These studies have problems when considering non-differentiability. Moreover, they must calculate the word probability distribution of each step of the decoder to learn a discriminant model. In this study, we propose a learning method that does not consider non-differentiability when learning using a GAN because it uses the response vector generated by the decoder. To generate natural responses in multi-turn dialogues, we propose an attention method between a query and its previous utterances that helps chatbots selectively consider the given context.

## 3. Multi-Turn Chatbot Model Based on Dual Wasserstein Generative Adversarial Networks

The proposed model consists of three sub-modules: a query encoder, a query-to-response (QR) mapper, and a response-to-response (RR) mapper, as shown in Figure 1.

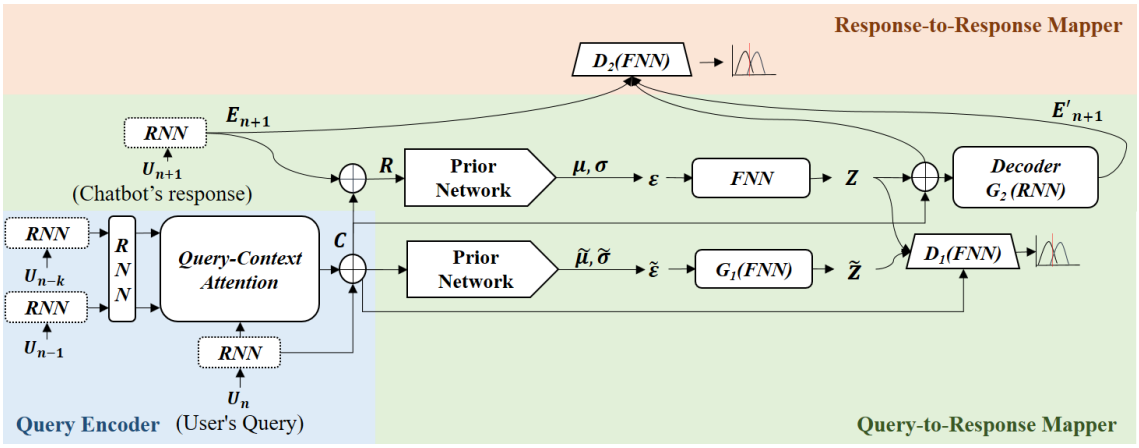


**Figure 1.** Overall architecture of the proposed chatbot. RNN: recurrent neural network; FNN: fully-connected neural network.

The query encoder returns a query vector embedding a current utterance $U_n$ (i.e., user query) and a dialogue context composed of $k$ previous utterances $U_{n-1}$, $U_{n-2}$, …, $U_{n-k}$, by using RNNs and a scaled dot product attention mechanism [14]. At training time, the QR mapper makes a query vector similar to an RNN-encoded response vector (i.e., a vector of a next utterance $U_{n+1}$; a vector of a chatbot's response) through an adversarial learning scheme. Then, it decodes an encoded response vector through an auto-encoder learning scheme. At the inference time, a query vector is input to a response decoder based on an RNN. The RR mapper makes an encoded response vector similar to a response vector decoded by the RNN through an adversarial learning scheme.

### 3.1. Query Encoder

Given a current utterance $U_n$ (user query) and its dialogue context composed of $k$ utterances, $U_{n-k}$, …, $U_{n-1}$, the query encoder encodes each utterance by using gated recurrent unit (GRU) networks [15], as shown in the following equation:

$$E_i = GRU(U_i) \tag{1}$$

where $E_i$ is an utterance vector encoded by a GRU network (i.e., the last output vector of a GRU network). Then, the query encoder reflects contextual information to each encoded utterance by using GRU networks, as shown in the following equation:

$$\widetilde{E}_i = GRU(E_i) \tag{2}$$

where $\widetilde{E}_i$ is an output vector of the $i$th step in a GRU network. To strongly reflect the contextual associations between the current and previous utterances, the query encoder computes attention scores through the well-known scaled dot products [14] between the encoded current utterance $E_n$ and the encoded previous utterances $\widetilde{E}_{n-k}$, …, $\widetilde{E}_{n-1}$ as shown in the following equation:

$$a_i = \frac{1}{Z} exp\left(\frac{\widetilde{E}_i \circ E_n}{\sqrt{d}}\right), \; where \; i \neq n \tag{3}$$

where $Z$ and $d$ are a normalization factor and the size of an encoded vector, respectively. Then, the query encoder computes an attention vector $A$ known as a query-to-context (QC) attention, which represents a dialogue context that should be considered to generate a response, as shown in the following equation:

$$A = \sum_{i=1}^{k} a_i \widetilde{E}_i \tag{4}$$

Finally, the query encoder returns a query vector in which the encoded current utterance $E_n$ and the QC attention $A$ are concatenated, as shown in the following equation:

$$Q = E_n \oplus A. \tag{5}$$

### 3.2. Query-to-Response Mapper

The QR mapper plays the role of mapping a query vector into a response vector. To enhance the mapping performance, we adopt the wasserstein auto-encoder (WAE) model [12], in which a Wasserstein GAN (WGAN) [16] is used to optimize a generator. The reason we choose the WGAN is that it has been shown to produce good results in text generation [16]. Given a dialogue history $U_{n-k}, \ldots, U_{n-1}, U_n$ and a next utterance $U_{n+1}$ (chatbot response), the QR mapper encodes the next utterance by using gated recurrent unit (GRU) networks [15] and generates a gold response vector $R$ by concatenating the query vector $Q$ that embeds the dialogue history and the encoded next utterance $E_{n+1}$, as shown in the following equation:

$$R = Q \oplus E_{n+1}. \tag{6}$$

Then, the QR mapper adds two Gaussian noises $\varepsilon$ and $\widetilde{\varepsilon}$ to the gold response vector $R$ and the query vector $Q$ by using prior networks that are fully-connected neural networks (FNNs) resulting in means and co-variances, respectively. The Gaussian noises are transformed into two latent variables, namely $Z$, representing a gold response, and $\widetilde{Z}$ representing a query, through FNNs. The training process of the QR mapper consists of two steps. In the first stage of training, the FNN-based discriminator $D_1$ (i.e., a classifier based on a FNN), as given in Figure 1, tries to distinguish the fake vector $\widetilde{Z}$ from the real vector $Z$. Through this training process, the QR mapper makes the query vector $Q$ similar to the gold response vector $R$. In the second stage of training, the RNN-based response decoder $G_2$ (i.e., a language model based on an RNN), as given in Figure 1, tries to properly reconstruct a sequence of words in a gold response. At the inference time, the query vector $Q$ is transformed into the latent variable $\widetilde{Z}$. Then, the latent variable $\widetilde{Z}$ is input to the response decoder $G_2$.

### 3.3. Response-to-Response Mapper

The RR mapper plays the role of mapping a generated response vector into an encoded response vector in an auto-encoder model. To enhance the mapping performance, we adopt the WAE model [12] again. Through the WAE process based on WGAN, we expect that the qualities of generated responses will be enhanced because the response decoder refers outputs of the response encoder once again. Given an RNN-encoded next utterance $E_{n+1}$ (i.e., a last output vector of the response encoder; an encoded response of a chatbot) and an RNN-decoded next utterance $E'_{n+1}$ (i.e., a last output vector of the response decode; a decoded response of a chatbot), the RR mapper makes $E'_{n+1}$ similar to $E_{n+1}$ through adversarial learning. When training begins, an encoded gold response $E_{n+1}$, a gold latent variable $Z$, and a decoded gold response $E'_{n+1}$ are input to the FNN-based discriminator $D_2$ as given in Figure 1. The discriminator tries to distinguish $E'_{n+1}$ from $E_{n+1}$.

### 3.4. Implementation and Training

The proposed model was implemented using TensorFlow 1.4 [17]. The RNNs in the query encoder were bidirectional GRU networks [18] with 300 hidden units in each direction. The RNNs in the QR mapper were GRU networks with 1421 hidden units. The dimensions of QC attentions were 600. The discriminators (i.e., $D_1$ in the QR mapper and $D_2$ in the RR mapper) were three-layer FNNs with rectified linear unit activation [19]. The vocabulary size was set to 16,925, and all the out-of-vocabulary words were defined to the special token "UNK" meaning an unknown word. The word-embedding size was 200, and word-embedding vectors were initialized using pre-trained Glove vectors [20]. The size of the dialogue context was set to 10 with a maximum utterance length of 41. The response decoder used a greedy decoding algorithm. In the training step, the mini-batch size was set to 32. The training process of the entire model consisted of three steps. First, the WGAN in the QR mapper was trained through adversarial learning. Then, the entire model, except the RR mapper, was trained through multi-task learning to maximize the log probability for the response decoder to generate correct words. Finally, the WGAN in the RR mapper was trained through adversarial learning. All adversarial learning that was employed reduced the Wasserstein distance [16]. In addition, the gradient penalty was used when training the discriminant models [21], and its hyper-parameter $\lambda$ was set to 10. To maximize the log-probability, a cross-entropy function was used. In the inference step, $\widetilde{Z}$ was used instead of Z as a latent variable.

## 4. Evaluation

### 4.1. Data Sets and Experimental Settings

We evaluated our model on a DailyDialog dataset [22], which has been widely used in recent studies. DailyDialog has 13,118 daily conversations for English learners. The datasets are separated into training, validation, and testing as 8:1:1. Bilingual evaluation understudy (BLEU) [23,24], bag-of-words (BOW) embedding [25], and Distinct [7] were used as performance measures. BLEU measures the number of generated responses that contain word $n$-gram overlaps with gold responses, as shown in the following equation:

$$\text{BLEU} = \min(1, \frac{\text{length of a generated sentence}}{\text{length of a gold sentence}})(\prod_{i=1}^{n} \text{precison}_i)^{\frac{1}{n}} \quad (7)$$

where $n$ is the maximum number length of $n$-grams considered and is commonly set to 4, and $precison_i$ is a word $i$-gram precision (i.e., the number of correct word $i$-grams divided by the number of word $i$-grams in a generated sentence). Precision of BLEU is an average score of BLEUs of 10 generated sentences per query, and Recall of BLEU is a maximum score among BLEUs of 10 generated sentences per query. The BOW embedding metric is the cosine similarity of BOW embedding between generated and gold responses. The BOW embedding metric consists of three metrics: Greedy [26], Average [27], and Extrema [28]. In our test, the maximum BOW embedding score among the 10 sampled responses was reported. The distinct score, such as Intra-dist or Inter-dist, computes the diversity of the generated responses. Dist-$n$ is defined as the ratio of unique word $n$-grams over all word $n$-grams in the generated responses. Intra-dist is defined as the average of distinct values within each sampled response, and Inter-dist as the distinct value among all sampled responses.

### 4.2. Experimental Results

Our first experiment involved evaluating the effectiveness of the proposed model at the architecture level, as shown in Table 2.

In Table 2, WAE is a conditional WAE model [12] that is the baseline model because it has similar architecture to that of the proposed model. WAE + query encoder (QE) is a modified WAE in which the encoding module of a dialog context is changed into the proposed query encoder. WAE + QE +

RR is a modified WAE + QE to which the RR mapper (i.e., WGAN for response generation) is added. As shown in Table 1, WAE + QE showed better BLEU-R and BLEU-F1 than did WAE. This means that the proposed query encoder can provide some assistance in generating words in gold responses by selectively looking up dialogue contexts. The final model, WAE + QE + RR, showed better performance than did WAE + QE at all measures except Intra-dist. This means that WGAN for response generation improves the overall quality of responses.

**Table 2.** Performance comparison based on change in architecture.

| Model | BLEU | | | BOW Embedding | | | Intra-dist | | Inter-dist | |
|---|---|---|---|---|---|---|---|---|---|---|
| | R | P | F1 | A | E | G | Dist-1 | Dist-2 | Dist-1 | Dist-2 |
| WAE | 0.341 | 0.278 | 0.306 | 0.948 | 0.578 | 0.846 | 0.830 | 0.940 | 0.327 | 0.583 |
| WAE + QE | 0.442 | 0.268 | 0.334 | 0.947 | 0.680 | 0.845 | **0.913** | **0.995** | 0.322 | 0.475 |
| WAE + QE + RR | **0.463** | **0.283** | **0.351** | **0.949** | **0.688** | **0.851** | 0.902 | 0.993 | **0.371** | **0.585** |

P: Precision, R: Recall, F1: F1 score, A: Average, E: Extrema, G: Greedy. BLEU: bilingual evaluation understudy; BOW: bag-of-words; WAE: wasserstein auto-encoder; QE: query encoder; RR: response-to-response. The bolds indicate the highest scores in each evaluation measure.

In the second experiment, we compared the performance of the proposed model with those of the previous state-of-the-art models, as shown in Table 3.

**Table 3.** Performance comparison between proposed and previous models.

| Model | BLEU | | | BOW Embedding | | | Intra-dist | | Inter-dist | |
|---|---|---|---|---|---|---|---|---|---|---|
| | R | P | F1 | A | E | G | Dist-1 | Dist-2 | Dist-1 | Dist-2 |
| HRED | 0.232 | 0.232 | 0.232 | 0.915 | 0.511 | 0.798 | 0.935 | 0.969 | 0.093 | 0.097 |
| CVAE | 0.265 | 0.222 | 0.242 | 0.923 | 0.543 | 0.811 | 0.938 | 0.973 | 0.177 | 0.222 |
| CVAE-BOW | 0.256 | 0.224 | 0.239 | 0.923 | 0.540 | 0.812 | **0.947** | 0.976 | 0.165 | 0.206 |
| CVAE-CO | 0.259 | 0.244 | 0.251 | 0.914 | 0.530 | 0.818 | 0.821 | 0.911 | 0.106 | 0.126 |
| WAE | 0.341 | 0.278 | 0.306 | 0.948 | 0.578 | 0.846 | 0.830 | 0.940 | 0.327 | 0.583 |
| WAE + QE + RR | **0.463** | **0.283** | **0.351** | **0.949** | **0.688** | **0.851** | 0.902 | **0.993** | **0.371** | **0.585** |

The bolds indicate the highest scores in each evaluation measure.

In Table 3, HRED is a generalized Seq2Seq model with a hierarchical RNN encoder [29]. CVAE is a conditional VAE model with KL-annealing [9]. CVAE-BOW is a conditional VAE model with a BOW loss [9]. CVAE-CO is a collaborative conditional VAE model [10]. WAE is a conditional WAE model [12]. As shown in Table 3, the proposed model, WAE + QE + RR, outperformed the comparison models at all measures except Dist-1.

## 5. Conclusions

We proposed a generative, multi-turn chatbot model. To generate responses that consider dialogue histories, the proposed model used the query-context attention mechanism in the query encoding step. Furthermore, to improve the quality of responses, the proposed model used two types of WGAN: A WGAN for dialogue modeling and a WGAN for response generation. In experiments with DailyDialog datasets, the proposed model outperformed the previous state-of-the-art models and generated responses with higher quality. The proposed chatbot model has the lack of a consistent personality because it is typically trained using many dialogues from different speakers. To alleviate this problem, we will study how to have a chatbot that maintains a consistent persona. In addition, we will study how a chatbot can search and use outer knowledge for open-domain dialogue.

**Author Contributions:** Conceptualization, H.K. and J.K.; methodology, J.K.; software, J.K.; validation, J.K. and S.O.; formal analysis, J.K.; investigation, J.K.; resources, S.O.; data curation, S.O.; writing—original draft preparation, J.K.; writing—review and editing, H.K.; visualization, H.K.; supervision, H.K. and O.-W.K.; project administration, H.K. and O.-W.K.; funding acquisition, H.K. and O.-W.K.

**Funding:** This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP), grant funded by the Korean government (MSIT) (2019-0-0004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners), and was partially supported by the National Research Foundation of Korea (NRF), grant funded by the Korean government (MSIP) (No.2016R1A2B4007732). This work was also partially supported by the Hyundai motor group.

**Acknowledgments:** We would especially like to thank the members of NLP laboratory in Kangwon National University for their technical support.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 2672–2680.
2. Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; Jurafsky, D. Adversarial Learning for Neural Dialogue Generation. *arXiv* **2017**, arXiv:1701.06547.
3. Xu, Z.; Liu, B.; Wang, B.; Chengjie, S.U.; Wang, X.; Wang, Z.; Qi, C. Neural Response Generation via GAN with an Approximate Embedding Layer. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9 September 2017; pp. 617–626.
4. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. Seqgan: Sequence Generative Adversarial Nets with Policy Gradient. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 13 February 2017.
5. Vinyals, O.; Le, Q. A neural conversational model. *arXiv* **2015**, arXiv:1506.05869.
6. Shang, L.; Lu, Z.; Li, H. Neural Responding Machine for Short-Text Conversation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 1577–1586.
7. Li, J.; Galley, M.; Brockett, C.; Gao, J.; Dolan, B. A Diversity-Promoting Objective Function for Neural Conversation Models. In Proceedings of the NAACL-HLT, San Diego, CA, USA, 1–26 July 2016; pp. 110–119.
8. Sato, S.; Yoshinaga, N.; Toyoda, M.; Kitsuregawa, M. Modeling Situations in Neural Chatbots. In Proceedings of the ACL 2017, Student Research Workshop, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 120–127.
9. Zhao, T.; Zhao, R.; Eskenazi, M. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 654–664.
10. Shen, X.; Su, H.; Niu, S.; Demberg, V. Improving Variational Encoder-Decoders in Dialogue Generation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, Lyon, France, 23–27 April 2018.
11. Goyal, P.; Hu, Z.; Liang, X.; Wang, C.; Xing, E.P. Nonparametric Variational Auto-Encoders for Hierarchical Representation Learning. In Proceedings of the IEEE International Conference on Computer Vision, Sinaia, Romania, 19–21 October 2017; pp. 5094–5102.
12. Gu, X.; Cho, K.; Ha, J.W.; Kim, S. DialogWAE: Multimodal Response Generation with Conditional Wasserstein Auto-Encoder. *arXiv* **2018**, arXiv:1805.12352.
13. Shen, T.; Lei, T.; Barzilay, R.; Jaakkola, T. Style Transfer from Non-Parallel Text by Cross-Alignment. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 6830–6841.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
15. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.

16. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
17. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A System for Large-Scale Machine Learning. In Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
18. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
19. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 July 2010; pp. 807–814.
20. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
21. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5767–5777.
22. Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; Niu, S. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In Proceedings of the Eighth International Joint Conference on Natural Language Processing, Taipei, Taiwan, 27 November–1 December 2017; Volume 1, pp. 986–995.
23. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 6 July 2002; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 311–318.
24. Chen, B.; Cherry, C. A Systematic Comparison of Smoothing Techniques for Sentence-Level Bleu. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, MD, USA, 26–27 June 2014; pp. 362–367.
25. Liu, C.W.; Lowe, R.; Serban, I.; Noseworthy, M.; Charlin, L.; Pineau, J. How not to Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 2122–2132.
26. Rus, V.; Lintean, M. A Comparison of Greedy and Optimal Assessment of Natural Language Student Input using Word-to-Word Similarity Metrics. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, Montréal, QC, Canada, 7 June 2012; Association for Computational Linguistics: Stroudsburg, PA, USA, 2012; pp. 157–162.
27. Mitchell, J.; Lapata, M. Vector-Based Models of Semantic Composition. In Proceedings of the ACL-08: HLT, Columbus, OH, USA, 15–20 June 2008; pp. 236–244.
28. Forgues, G.; Pineau, J.; Larchevêque, J.M.; Tremblay, R. Bootstrapping Dialog Systems with Word Embeddings. In Proceedings of the Nips, Modern Machine Learning and Natural Language Processing Workshop, Geneva, Switzerland, 12 December 2014.
29. Serban, I.V.; Sordoni, A.; Bengio, Y.; Courville, A.; Pineau, J. Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 3776–3783.