

# Low-Resource Response Generation with Template Prior

Ze Yang<sup>†</sup>, Wei Wu<sup>◇</sup>, Jian Yang<sup>†</sup>, Can Xu<sup>◇</sup>, Zhoujun Li<sup>†\*</sup>

<sup>†</sup>State Key Lab of Software Development Environment, Beihang University, Beijing, China

<sup>◇</sup>Microsoft Corporation, Beijing, China

{tobey, jiaya, lizj}@buaa.edu.cn {wuwei, caxu}@microsoft.com

## Abstract

We study open domain response generation with limited message-response pairs. The problem exists in real-world applications but is less explored by the existing work. Since the paired data now is no longer enough to train a neural generation model, we consider leveraging the large scale of unpaired data that are much easier to obtain, and propose response generation with both paired and unpaired data. The generation model is defined by an encoder-decoder architecture with templates as prior, where the templates are estimated from the unpaired data as a neural hidden semi-markov model. By this means, response generation learned from the small paired data can be aided by the semantic and syntactic knowledge in the large unpaired data. To balance the effect of the prior and the input message to response generation, we propose learning the whole generation model with an adversarial approach. Empirical studies on question response generation and sentiment response generation indicate that when only a few pairs are available, our model can significantly outperform several state-of-the-art response generation models in terms of both automatic and human evaluation.

## 1 Introduction

Human-machine conversation is a long-standing goal of artificial intelligence. Early dialogue systems are designed for task completion with conversations restricted in specific domains (Young et al., 2013). Recently, thanks to the advances in deep learning techniques (Sutskever et al., 2014; Vaswani et al., 2017) and the availability of large amounts of human conversation on the internet, building an open domain dialogue system with data-driven approaches has become the new fashion in the research of conversational AI. Such dia-

logue systems can generate reasonable responses without any needs on rules, and have powered products in the industry such as Amazon Alexa (Ram et al., 2018) and Microsoft XiaoIce (Shum et al., 2018).

State-of-the-art open domain response generation models are based on the encoder-decoder architecture (Vinyals and Le, 2015; Shang et al., 2015). On the one hand, with proper extensions to the vanilla structure, existing models now are able to naturally handle conversation contexts (Serban et al., 2016; Xing et al., 2018), and synthesize responses with various styles (Wang et al., 2017), emotions (Zhou et al., 2018), and personas (Li et al., 2016a). On the other hand, all the existing success of open domain response generation builds upon an assumption that the large scale of paired data (Shao et al., 2016) or conversation sessions (Sordoni et al., 2015) are available. In this work, we challenge this assumption by arguing that one cannot always obtain enough pairs or sessions for training a neural generation model. For example, although it has been indicated by existing work (Li et al., 2016b; Wang et al., 2018) that question asking in conversation can enhance user engagement, we find that in a public dataset<sup>1</sup> with 5 million conversation sessions crawled from Weibo, only 7.3% sessions have a question response and thus can be used to learn a question generator for responding<sup>2</sup>. When we attempt to generate responses that express positive sentiment, we only get 360k pairs (18%) with positive responses from a dataset with 2 million message-response pairs crawled from Twitter. Indeed, existing big conversation data mix various intentions, styles, emotions, personas, and so on. Thus,

<sup>1</sup><http://tcci.ccf.org.cn/conference/2018/dldoc/trainingdata05.zip>

<sup>2</sup>Questions are detected with the rules in (Wang et al., 2018).

\* Corresponding Author

we have to face the data sparsity problem, as long as we attempt to create a generation model with constraints on responses.

In this work, we jump out of the paradigm of learning from large scale paired data<sup>3</sup>, and investigate how to build a response generation model with only a few pairs at hand. Aside from the paired data, we assume that there are a large number of unpaired data available. The assumption is reasonable since it is much easier to get questions or sentences with positive sentiment than to get such responses paired with messages. We formalize the problem as low-resource response generation from paired and unpaired data, which is less explored by existing work. Since the paired data are insufficient for learning the mapping from a message to a response, the challenge of the task lies in how to effectively leverage the unpaired data to enhance the learning on the paired data. Our solution to the challenge is a two-stage approach where we first distill templates from the unpaired data and then use them to guide response generation. Targeting on an unsupervised approach to template learning, we propose representing the templates as a neural hidden semi-markov model (NHSMM) estimated through maximizing the likelihood of the unpaired data. Such latent templates encode both semantics and syntax of the unpaired data and then are used as prior in an encoder-decoder architecture for modeling the paired data. With the latent templates, the whole model is end-to-end learnable and can perform response generation in an explainable manner. To ensure the relevance of responses regarding input messages and at the same time make full use of the templates, we propose learning the generation model with an adversarial approach.

Empirical studies are conducted on two tasks: question response generation and sentiment response generation. For the first task, we exploit the dataset published in (Wang et al., 2018) and augment the data with questions crawled from Zhihu<sup>4</sup>. For the second task, we build a paired dataset from Twitter by filtering responses with an off-the-shelf sentiment classifier and augment the dataset with tweets in positive sentiment extracted from a large scale tweet dataset published in (Cheng et al., 2010). Evaluation results on both auto-

<sup>3</sup>The study in this work starts from response generation for single messages. One can easily extend the proposed approach to handle conversation history.

<sup>4</sup><https://en.wikipedia.org/wiki/Zhihu>

matic metrics and human judgment indicate that with limited message-response pairs, our model can significantly outperform several state-of-the-art response generation models. The source code is available online.<sup>5</sup>

Our contributions in this work are three-folds: (1) proposal of low-resource response generation with paired and unpaired data for open domain dialogue systems; (2) proposal of encoder-decoder with template prior; and (3) empirical verification of the effectiveness of the model with two large-scale datasets.

## 2 Related Work

Inspired by neural machine translation, early work applies the sequence-to-sequence with attention model (Shang et al., 2015) to open domain response generation, and gets promising results. Later, the basic architecture is extended to suppress generic responses (Li et al., 2015; Zhao et al., 2017; Xing et al., 2017); to model the structure of conversation contexts (Serban et al., 2016); and to incorporate different types of knowledge into generation (Li et al., 2016a; Zhou et al., 2018). In addition to model design, how to learn a generation model (Li et al., 2016c, 2017), and how to evaluate the models (Liu et al., 2016; Lowe et al., 2017; Tao et al., 2018), are drawing attention in the community of open domain dialogue generation. In this work, we study how to learn a response generation model from limited pairs, which breaks the assumption made by existing work. We propose response generation with paired and unpaired data. As far as we know, this is the first work on low-resource response generation for open domain dialogue systems.

Traditional template-based text generation (Becker, 2002; Foster and White, 2004; Gatt and Reiter, 2009) relies on handcrafted templates that are expensive to obtain. Recently, some work explores how to automatically mine templates from plain text and how to integrate the templates into neural architectures to enhance interpretability of generation. Along this line, Duan et al. (2017) mine patterns from related questions in community QA websites and leverage the patterns with a retrieval-based approach and a generation-based approach for question generation. Wiseman et al. (2018) exploit a hidden semi-markov model for joint template extraction and text generation. In

<sup>5</sup>[https://github.com/TobeyYang/S2S\\_Temp](https://github.com/TobeyYang/S2S_Temp)

addition to structured templates, raw text retrieved from indexes is also used as “soft templates” in various natural language generation tasks (Gua et al., 2018; Pandey et al., 2018; Cao et al., 2018; Peng et al., 2019). In this work, we leverage templates for open domain response generation. Our idea is inspired by (Wiseman et al., 2018), but latent templates estimated from one source are transferred to another source in order to handle the low-resource problem, and the generation model is learned by an adversarial approach rather than by maximum likelihood estimation.

Before us, the low-resource problem has been studied in tasks such as machine translation (Gu et al., 2018b,a), pos tagging (Kann et al., 2018), word embedding (Jiang et al., 2018), automatic speech recognition (Tüske et al., 2014), task-oriented dialogue systems (Tran and Nguyen, 2018; Mi et al., 2019), etc. In this work, we pay attention to low-resource open domain response generation which is untouched by existing work. We propose attacking the problem with unpaired data, which is related to the effort in low-resource machine translation with monolingual data (Gulcehre et al., 2015; Sennrich et al., 2015; Zhang and Zong, 2016). Our method is unique in that rather than using the unpaired data through multi-task learning (Zhang and Zong, 2016) or back-translation (Sennrich et al., 2015), we extract linguistic knowledge from the data as latent templates and use the templates as prior in generation.

### 3 Low-Resource Response Generation

In this section, we first formalize the setting upon which we study low-resource response generation and then elaborate the model of response generation with paired and unpaired data, including how to learn latent templates from the unpaired data, and how to perform generation with the templates.

#### 3.1 Problem Formalization

Suppose that we have a dataset  $\mathcal{D}_P = \{(X_i, Y_i)\}_{i=1}^n$ , where  $\forall i$ ,  $(X_i, Y_i)$  is a pair of message-response, and  $n$  represents the number of pairs in  $\mathcal{D}_P$ . Different from existing work, we assume that  $n$  is small (e.g., a few hundred thousands) and further assume that there is another set  $\mathcal{D}_U = \{T_i\}_{i=1}^N$  with  $T_i$  a piece of plain text sharing the same characteristics with  $\{Y_i\}_{i=1}^n$  (e.g., both are questions) and  $N > n$ . Our goal is to learn a generation probability  $P(Y|X)$  with both  $\mathcal{D}_P$  and

$\mathcal{D}_U$ . Thus, given a new message  $X$ , we can generate a response  $Y$  for  $X$  following  $P(Y|X)$ .

Since the limited resource in  $\mathcal{D}_P$  may not support accurately learning of  $P(Y|X)$ , we try to transfer the linguistic knowledge in  $\mathcal{D}_U$  to response generation. The challenges then lie in two aspects: (1) how to represent the linguistic knowledge in  $\mathcal{D}_U$ ; and (2) how to effectively leverage the knowledge extracted from  $\mathcal{D}_U$  for response generation, given that  $\mathcal{D}_U$  cannot provide any information of correspondence between a message  $X$  and a response  $Y$ . The remaining part of the section will describe our solutions to the two problems.

#### 3.2 Learning Templates from $\mathcal{D}_U$

In the representation of the knowledge in  $\mathcal{D}_U$ , we hope that both semantic information and syntactic information can be kept. Thus, we consider extracting templates from  $\mathcal{D}_U$  as the knowledge. A template segments a piece of text as a structured representation. With the templates, semantically and functionally similar text segments are grouped together. Since the templates encode the structure of language in  $\mathcal{D}_U$ , they can inform the generation model about how to express a response in a desired way (e.g., as a question or with the specific sentiment). Here, we prefer an unsupervised and parametric approach to learning templates, since “unsupervised” means that the approach is generally applicable to various tasks, and “parametric” allows us to naturally incorporate the templates into the generation model. Then, a natural choice for template learning is the neural hidden semi-markov model (NHSMM) (Dai et al., 2016; Wiseman et al., 2018).

NHSMM is an HSMM parameterized with neural networks. HSMM (Murphy, 2002) extends HMM by allowing a hidden state to emit a sequence of observations and thus can segment a piece of text with the latent variables and group similar segments by the variables. Formally, given an observed sequence  $Y = (y_1, \dots, y_S)$ , the joint distribution of  $Y$  and its segmentation is

$$\prod_{t=1}^{S'} P(z_{t+1}, l_{t+1} | z_t, l_t) \prod_{t=1}^{S'} P(y_{i(t-1)+1:i(t)} | z_t, l_t),$$

where  $z_t \in \{1, \dots, K\}$  is the hidden state for step  $t$ ,  $l_t \in \{1, \dots, D\}$  is the duration variable for  $z_t$  that represents the number of tokens emitted by  $z_t$ ,  $i(t) = \sum_{j=1}^t l_j$  with  $i(0) = 0$  and  $i(S') = S$ , and  $y_{i(t-1)+1:i(t)}$  is the sequence

of  $(y_{i(t-1)+1}, \dots, y_{i(t)})$ .  $P(z_{t+1}, l_{t+1} | z_t, l_t)$  is factorized as  $P(z_{t+1} | z_t) \times P(l_{t+1} | z_{t+1})$  where  $P(l_{t+1} | z_{t+1})$  is a uniform distribution on  $\{1, \dots, D\}$  and  $P(z_{t+1} | z_t)$  can be viewed as a transition matrix  $[A(i, j)]_{K \times K}$  which is defined by

$$A(i, j) \triangleq P(z_{t+1} = j | z_t = i) = \frac{\exp(e_j^\top e_i + b_{i,j})}{\sum_{o=1}^K \exp(e_o^\top e_i + b_{i,o})},$$

where  $e_i, e_j, e_o \in \mathbb{R}^{d_1}$  are embeddings of state  $i, j, o$  respectively, and  $b_{i,j}, b_{i,o}$  are scalar bias terms. In practice, we set  $b_{i,j} = -\infty \Leftrightarrow i = j$  to disable self-transition, because the adjacent states play different syntactic or semantic roles in a desired template. The emission distribution  $P(y_{i(t-1)+1:i(t)} | z_t, l_t)$  is defined by

$$P(y_{i(t-1)+1:i(t)} | z_t, l_t) = P(y_{i(t-1)+1} | z_t, l_t) \times \prod_{j=2}^{l_t} P(y_{i(t-1)+j} | y_{i(t-1)+j-1}, z_t, l_t),$$

and parameterized with a recurrent neural network with gated recurrent unit (GRU) (Cho et al., 2014). The hidden vector for position  $j$  is formulated as

$$\begin{aligned} o_j^t &= \text{GRU}_H(o_{j-1}^t, [e_{z_t}; e_{y_{i(t-1)+j-1}}]) \\ v_j^t &= g_{z_t} \odot o_j^t, \end{aligned} \quad (1)$$

where  $o_j^t \in \mathbb{R}^{d_2}$ ,  $e_{y_{i(t-1)+j-1}} \in \mathbb{R}^{d_3}$  is the embedding of word  $y_{i(t-1)+j-1}$ ,  $[\cdot; \cdot]$  is a concatenation operator,  $\odot$  refers to element-wise multiplication, and  $g_{z_t} \in \mathbb{R}^{d_2}$  is a gate (in total, there are  $K$  gate vectors as parameters).  $P(y_{i(t-1)+j} | y_{i(t-1)+j-1}, z_t, l_t)$  is then defined by

$$\text{softmax}(W_1 v_j^t + b_1),$$

where  $W_1 \in \mathbb{R}^{V \times d_2}$  and  $b_1 \in \mathbb{R}^{d_2}$  are parameters with  $V$  the vocabulary size.

Following Murphy (2002), the marginal distribution of  $Y$  can be obtained by the backward algorithm which is formulated as

$$\begin{aligned} \beta_t(i) &\triangleq P(y_{t+1:S} | q_t = i) = \sum_{j=1}^K \beta_t^*(j) A(i, j) \\ \beta_t^*(j) &\triangleq P(y_{t+1:S} | q_{t+1} = j) \\ &= \sum_{d=1}^D \left[ \beta_{t+d}^*(j) P(d|j) P(y_{t+1:t+d} | j, d) \right] \\ P(Y) &= \sum_{j=1}^K \beta_0^*(j) P(q_1 = j). \end{aligned} \quad (2)$$

where  $q_t$  is the hidden state of the  $t$ -th word in  $Y$ , and the base cases  $\beta_S(i) = 1, \forall i \in \{1, \dots, K\}$ . Specifically, to learn more reasonable segmentations, we parsed every sentence by stanford parser (Manning et al., 2014) and forced NHSMM not to break syntactic elements such as VP and NP, etc. The parameters of the NHSMM are estimated by maximizing the log-likelihood of  $\mathcal{D}_U$  through backpropagation.

### 3.3 Response Generation with Template Prior

We propose incorporating the templates parameterized by the NHSMM learned from  $\mathcal{D}_U$  into response generation as prior. Figure 1 illustrates the architecture of the generation model. In a nutshell, the model first samples a chain of states with duration as a template. The template specifies a segmentation of the response to generate. Then, the hidden representations of the segments defined by Equation (1) are fed to an encoder-decoder architecture for response generation, where the hidden states of the decoder are calculated with both attention over the hidden states of the input message given by the encoder and the hidden representations of the segments given by the template prior. The template prior acts as a base and assists the encoder-decoder in response generation regarding to an input message, when paired information is insufficient for learning the correspondence between a message and a response. Note that similar to the conditional variational autoencoder (CVAE) (Zhao et al., 2017), our model also exploits hidden variables for response generation. The difference is that the hidden variables in our model are structured and learned from extra resources, and thus encode more semantic and syntactic information.

Specifically, we segment responses in  $\mathcal{D}_P$  with Viterbi algorithm (Zucchini et al., 2016), collect all chains of states as a pool and sample a chain from the pool uniformly. We do not sample states according to the transition matrix  $[A(i, j)]_{K \times K}$ , since it is difficult to determine the end of a chain. Suppose that the sampled chain is  $(z_1, \dots, z_S)$ , then  $\forall 1 \leq t \leq S$ , we sample an  $l_t$  for  $z_t$  according to  $P(l_t | z_t)$ , and finally form a latent template  $(\langle z_1, l_1 \rangle, \dots, \langle z_S, l_S \rangle)$ . Given a message  $X = (x_1, \dots, x_L)$ , the encoder exploits a GRU to transform  $X$  into a hidden sequence  $H_X = (h_{X,1}, \dots, h_{X,L})$  with the  $i$ -th hidden state

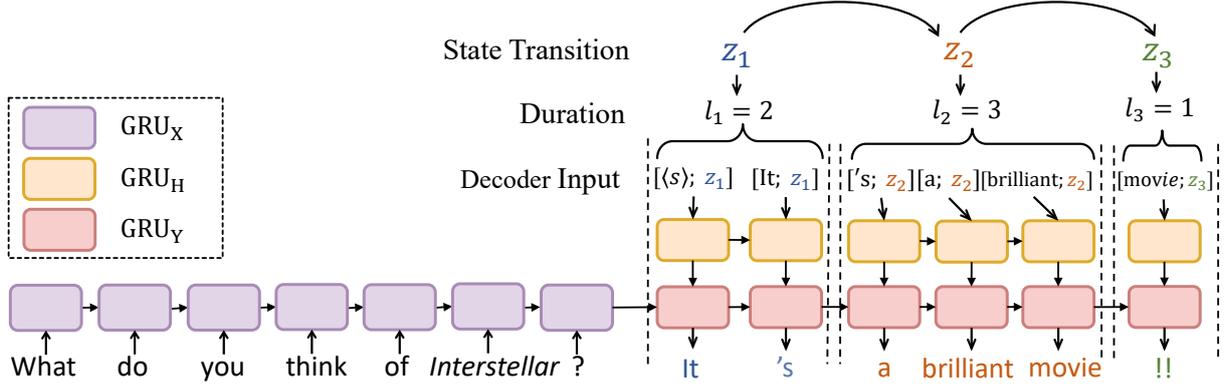


Figure 1: The architecture of the generation model.

$h_{X,i} \in \mathbb{R}^{d_2}$  given by

$$h_{X,i} = \text{GRU}_X(h_{X,i-1}, e_{x_i}),$$

where  $e_{x_i} \in \mathbb{R}^{d_3}$  is the embedding of word  $x_i$  and  $h_{X,0} = 0$ . Then when predicting the  $t$ -th word of the response, the decoder calculates the probability  $P(y_t|y_{1:t-1}, X, \mathcal{T})$  via

$$P(y_t|y_{1:t-1}, X, \mathcal{T}) = \text{softmax}(W_2[s_t; c_t] + b_2)$$

$$s_t = \text{GRU}_Y(s_{t-1}, v_k^m).$$

with parameters  $W_2 \in \mathbb{R}^{V \times 2d_2}$  and  $b_2 \in \mathbb{R}^V$ ,  $s_t \in \mathbb{R}^{d_2}$  and  $s_{t-1} \in \mathbb{R}^{d_2}$  are the hidden states of the decoder for step  $t$  and step  $t-1$  respectively,  $v_k^m$  is defined by Equation (1) where  $m$  satisfies  $i(m-1) < t \leq i(m)$ ,  $k = t - i(m-1)$ , and  $o_k^m = \text{GRU}_H(o_{k-1}^m, [e_{z_m}; e_{y_{t-1}}])$ , and  $c_t \in \mathbb{R}^{d_2}$  is a context vector of  $X$  obtained via attention over  $H_X$  (Bahdanau et al., 2015):

$$c_t = \sum_{i=1}^L \alpha_{t,i} h_{X,i},$$

$$\alpha_{t,i} = \frac{\exp(s_{t,i})}{\sum_{j=1}^L \exp(s_{t,j})}$$

$$s_{t,i} = v^\top \tanh(Ws_t + Uh_i + b),$$

where  $v, b \in \mathbb{R}^{d_2}$ ,  $W, U \in \mathbb{R}^{d_2 \times d_2}$  are parameters.

#### 4 Learning Approach

Intuitively, we can estimate the parameters of the encoder-decoder and fine-tune the parameters of NHSM by maximizing the likelihood of  $\mathcal{D}_P$  (i.e., MLE). However, since  $\mathcal{D}_P$  only contains a few pairs, the MLE approach may suffer from a dilemma: (1) if we stop training early, then both the template prior and the encoder-decoder are not

sufficiently supervised by the pairs. In that case, the linguistic knowledge in  $\mathcal{D}_U$  will play a more important role in response generation and result in irrelevant responses regarding to messages; or (2) if we let the training go deep, then the template prior will be overwhelmed by the pairs in  $\mathcal{D}_P$ . As a result, the generation model will lose the knowledge obtained from  $\mathcal{D}_U$ . Since response generation starts from a latent template, we consider learning the model with an adversarial approach (Goodfellow et al., 2014) that can well balance the effect of the latent template and the input message. The learning involves a generator  $G$  described in Section 3 and a discriminator  $D$ .  $G$  is updated with REINFORCE algorithm (Williams, 1992) with rewards defined by  $D$ , and  $D$  is updated to distinguish human responses in  $\mathcal{D}_P$  from responses generated by  $G$ .

**Generator Pre-training.** To improve the stability of adversarial learning, we first pre-train  $G$  with MLE on  $\mathcal{D}_P$ .  $\forall (X_i, Y_i) \in \mathcal{D}_P$ , the template prior  $\mathcal{T}_i$  is obtained by running Viterbi algorithm (Zucchini et al., 2016) on  $Y_i$  rather than by sampling. Let  $Y_i = (y_{i,1}, \dots, y_{i,S_i})$ , then the objective of pre-training is given by

$$\frac{1}{n} \sum_{(X_i, Y_i) \in \mathcal{D}_P} \sum_{t=1}^{S_i} \log P(y_{i,t} | y_{i,1:t-1}, X_i, \mathcal{T}_i). \quad (3)$$

**Discriminator Update.** The discriminator  $D$  is defined by a convolutional neural network (CNN) based binary classifier (Kim, 2014).  $D$  takes a message-response pair as input and outputs a score that indicates how likely the response is from humans. In the model, the message and the response are separately embedded as vectors by CNNs, and then the concatenation of the two vectors are fed to

a 2-layer MLP to calculate the score. Let  $\hat{Y}_i$  be the response generated by  $G$  for  $X_i$ , then  $D$  is updated by maximizing the following objective:

$$\sum_{(X_i, Y_i) \in \mathcal{D}_P} \log D(X_i, Y_i) + \log(1 - D(X_i, \hat{Y}_i)) \quad (4)$$

**Generator Update.** The generator  $G$  is updated by the policy gradient method (Yu et al., 2017; Li et al., 2017). Let  $\hat{y}_{1:t}$  be a partial response generated by  $G$  from beam search for message  $X$  until step  $t$ , then we adopt the Monte Carlo (MC) search method and sample  $N$  paths that supplement  $\hat{y}_{1:t}$  as responses  $\{\hat{Y}_i\}_{i=1}^N$ . The intermediate reward for  $\hat{y}_{1:t}$  is defined as  $R_t = \frac{1}{N} \sum_{i=1}^N D(X, \hat{Y}_i)$ . The gradient for updating  $G$  is given by

$$\nabla J(\theta) \approx \sum_{t=1}^S [\nabla \log P(\hat{y}_t | \hat{y}_{1:t-1}, X, \mathcal{T}) \cdot R_t], \quad (5)$$

where  $\theta$  represents the parameters of  $G$ , and  $\mathcal{T}$  is a sampled template. To control the quality of MC search, we sample from top 50 most probable words at each step.

The learning algorithm is summarized in Algorithm 1. Note that in learning of the generation model from  $\mathcal{D}_P$ , we freeze the embedding of states (i.e.,  $e_{z_t}$  in Equation (1)) and the embedding of words given by the NHSMM, and update all other parameters in generator pre-training and the following adversarial learning.

## 5 Experiments

We test the proposed approach on two tasks: question response generation and sentiment response generation. The first task requires a model to generate a question as a response to a given message; while in the second task, as a showcase, responses should express the positive sentiment.

### 5.1 Experiment Setup

**Datasets.** For the question response generation task, we choose the data published in (Wang et al., 2018) as the paired dataset. The data are obtained by filtering 9 million message-response pairs mined from Weibo with 20 handcrafted question templates and are split as a training set, a validation set, and a test set with 481k, 5k, and 5k pairs respectively. In addition to the paired data, we crawl 776k questions from Zhihu, a Chinese community QA website featured by high-quality

---

**Algorithm 1** Learning a generation model with paired and unpaired data.

---

**Require:** NHSMM  $H$ , generator  $G$ , discriminator  $D$ ,  $\mathcal{D}_U$ , and  $\mathcal{D}_P$ .

- 1: Initialize  $H, G, D$ .
  - 2: Learn  $H$  from  $\mathcal{D}_U$  according to Equation (2).
  - 3: Pre-train  $G$  using MLE on  $\mathcal{D}_P$ .
  - 4: Pre-train  $D$  using  $G$  according to Equation (4).
  - 5: **repeat**
  - 6:   **for** g-steps **do**
  - 7:     Sample  $(X, Y)$  from  $\mathcal{D}_P$ .
  - 8:     Sample a template  $\mathcal{T}$  for  $(X, Y)$  with  $H$ .
  - 9:     Generate  $\hat{Y} = (\hat{y}_1, \dots, \hat{y}_S) \sim G(\cdot | X, \mathcal{T})$ .
  - 10:    **for**  $t$  in  $1 : S$  **do**
  - 11:     MC search and compute reward  $R_t$  using  $D$ .
  - 12:    **end for**
  - 13:    Update  $G$  on  $(X, \hat{Y})$  via Equation (5).
  - 14:   **end for**
  - 15:   **for** d-steps **do**
  - 16:     Sample  $(X, Y)$  from  $\mathcal{D}_P$ .
  - 17:     Sample a template  $\mathcal{T}$  for  $(X, Y)$  with  $H$ .
  - 18:     Generate  $\hat{Y} \sim G(\cdot | X, \mathcal{T})$  and pair  $(X, \hat{Y})$  with  $(X, Y)$ .
  - 19:     Update  $D$  by Equation (4).
  - 20:   **end for**
  - 21: **until** convergence
- 

content, as an unpaired dataset. Both datasets are tokenized by Stanford Chinese word segmenter<sup>6</sup>. We keep 20,000 most frequent words in the two data as a vocabulary for the encoder, the decoder, and the NHSMM. The vocabulary covers 95.8% words appearing in the messages, in the responses, and in the questions. Other words are replaced with “UNK”. For the sentiment response generation task, we mine 2 million message-response pairs from Twitter FireHose, filter responses with the positive sentiment using Stanford Sentiment Annotator toolkit (Socher et al., 2013), and obtain 360k pairs as a paired dataset. As pre-processing, we remove URLs and usernames, and transform each word to its lower case. After that, the data is split as a training set, a validation set, and a test set with 350k, 5k, and 5k pairs respectively. Besides, we extract 1 million tweets with positive sentiment from a public corpus (Cheng et al., 2010) as an unpaired dataset. Top 20,000 most frequent words in the two data are kept as a vocabulary that covers 99.3% words. Words excluded from the vocabulary are treated as “UNK”. In both tasks, human responses in the test sets are taken as ground truth for automatic metric calculation. From each test set, we randomly sample 500 distinct messages and recruit human annotators to judge the quality of responses generated for these messages.

<sup>6</sup><https://stanfordnlp.github.io/CoreNLP>

**Evaluation Metrics.** We conduct evaluation with both automatic metrics and human judgments. For automatic evaluation, besides BLEU-1 (Papineni et al., 2002) and Rouge-L (Lin, 2004), we follow (Serban et al., 2017) and employ Embedding Average (Average), Embedding Extrema (Extrema), and Embedding Greedy (Greedy) as metrics. All these metrics are computed by a popular NLG evaluation project available at <https://github.com/Maluuba/nlg-eval>. In terms of human evaluation, for each task, we recruit 3 well-educated native speakers as annotators, and let them compare our model and each of the baselines. Every time, we show an annotator a message (in total 500) and two responses, one from our model and the other from a baseline model. Both responses are top 1 results in beam search, and the two responses are presented in random order. The annotator then compare the two responses from three aspects: (1) **Fluency**: if the response is fluent without grammatical error; (2) **Relevance**: if the response is relevant to the given message; and (3) **Richness**: if the response contains informative and interesting content, and thus may keep conversation going. For each aspect, if the annotator cannot tell which response is better, he/she is asked to label a “tie”. Each pair of responses receive 3 labels on each of the three aspects, and agreements among the annotators are measured by Fleiss’ kappa (Fleiss and Cohen, 1973).

## 5.2 Baselines

We compare our model with the following baselines: (1) **Seq2Seq**: the basic sequence-to-sequence with attention architecture (Bahdanau et al., 2015). (2) **CVAE**: the conditional variational autoencoder that represents the relationship between messages and responses with latent variables (Zhao et al., 2017). We use the code published at <https://github.com/snakeztc/NeuralDialog-CVAE>. (3) **HTD**: the hard typed decoder model proposed in (Wang et al., 2018) that exhibits the best performance on the dataset selected by this work for question response generation. The model estimates distributions over three types of words (i.e., interrogative, topic, and ordinary) and modulates the final distribution during generation. Since our experiments are conducted on the same data as those in (Wang et al., 2018), we run the code shared

	BLEU-1	ROUGE-L	AVERAGE	EXTREME	GREEDY
Seq2Seq	0.037	0.111	0.656	0.438	0.456
CVAE	0.094	0.088	0.685	0.414	0.422
HTD	0.073	0.103	0.647	0.425	0.439
S2S-Temp-MLE	0.097	0.119	0.699	0.438	0.457
S2S-Temp-None	0.069	0.092	0.677	0.429	0.416
S2S-Temp-50%	0.091	0.113	0.702	0.442	0.461
S2S-Temp	<b>0.102</b>	<b>0.128</b>	<b>0.710</b>	<b>0.451</b>	<b>0.469</b>

Table 1: Automatic evaluation results for the task of question response generation. Numbers in bold mean that the improvement over the best performing baseline is statistically significant (t-test with  $p$ -value < 0.01).

	BLEU-1	ROUGE-L	AVERAGE	EXTREME	GREEDY
Seq2Seq	0.065	0.118	0.726	0.474	0.582
CVAE	0.088	0.081	0.727	0.408	0.563
ECM	0.051	0.102	0.708	0.462	0.559
S2S-Temp-MLE	0.103	0.124	0.732	0.458	0.593
S2S-Temp-None	0.078	0.089	0.687	0.479	0.501
S2S-Temp-50%	0.102	0.121	0.691	0.491	0.586
S2S-Temp	<b>0.106</b>	<b>0.130</b>	<b>0.738</b>	<b>0.492</b>	<b>0.603</b>

Table 2: Automatic evaluation results for the task of sentiment response generation. Numbers in bold mean that the improvement over the best performing baseline is statistically significant (t-test, with  $p$ -value < 0.01).

at [https://github.com/victorywys/Learning2Ask\\_TypedDecoder](https://github.com/victorywys/Learning2Ask_TypedDecoder) with the default setting. (4) **ECM**: emotional chatting machine proposed in (Zhou et al., 2018). We implement the model with the code published at <https://github.com/tuxchow/ecm>. Since the model can handle various emotions, we train the model with the entire 2 million Twitter message-response pairs labeled with a positive, negative, and neutral sentiment. Thus, when we only focus on responses with positive sentiment, ECM actually performs multi-task learning for response generation. In the test, we set the sentiment label as “positive”.

We name our model **S2S-Temp**. Besides the full model, we also examine three variants in order to understand the effect of unpaired data and the role of adversarial learning: (1) S2S-Temp-None. The proposed model is trained only with the paired data, where the NHSM is estimated from responses in the paired data; (2) S2S-Temp-50%. The proposed model is trained with 50% unpaired data; and (3) S2S-Temp-MLE. The pre-trained generator described in Section 4. These variants are only involved in automatic evaluation.

## 5.3 Implementation Details

In both tasks, we set the number of states (i.e.,  $K$ ) and the maximal number of emissions (i.e.,  $D$ ) in NHSM as 50 and 4 respectively.  $d_1$ ,  $d_2$  and

Models	Fluency			Relevance			Richness			Kappa
	W(%)	L(%)	T(%)	W(%)	L(%)	T(%)	W(%)	L(%)	T(%)	
S2S-Temp vs. Seq2Seq	20.8	18.3	60.9	30.8	22.5	46.7	42.5	19.2	38.3	0.63
S2S-Temp vs. CVAE	41.7	5.7	52.6	50.8	12.5	36.7	37.5	15.8	46.7	0.71
S2S-Temp vs. HTD	35.1	19.2	45.8	30.8	25.1	44.1	37.5	30.8	31.7	0.64
S2S-Temp vs. Seq2Seq	15.6	11.5	72.9	34.4	17.2	48.4	31.9	7.3	60.8	0.68
S2S-Temp vs. CVAE	48.4	9.0	42.6	31.9	5.7	62.4	31.4	8.2	60.4	0.69
S2S-Temp vs. ECM	27.1	12.3	60.6	36.9	13.9	49.2	27.9	10.6	61.5	0.78

Table 3: Human annotation results. W, L, and T refer to Win, Lose and Tie respectively. The first three rows are results on question response generation, and the last three rows are results on sentiment response generation. The ratios are calculated by combining labels from the three judges.

$d_3$  are set as 600, 300, and 300 respectively. In adversarial learning, we use three types of filters with window sizes 1, 2 and 3 in the discriminator. The number of filters is 128 for each type. The number of samples obtained from MC search (i.e.,  $N$ ) at each step is 5. We learn all models using Adam algorithm (Kingma and Ba, 2015), monitor perplexity on the validation sets, and terminate training when perplexity gets stable. In our model, learning rates for NHSMM, the generator, and the discriminator are set as  $1 \times 10^{-3}$ ,  $1 \times 10^{-5}$ , and  $1 \times 10^{-3}$  respectively.

#### 5.4 Evaluation Results

Table 1 and Table 2 report the results of automatic evaluation on the two tasks. We can see that on both tasks, S2S-Temp outperforms all baseline models in terms of all metrics, and the improvements are statistically significant (t-test with  $p\text{-value} < 0.01$ ). The results demonstrate that when only limited pairs are available, S2S-Temp can effectively leverage unpaired data to enhance the quality of response generation. Although lacking fine-grained check, from the comparison among S2S-Temp-None, S2S-Temp-50%, and S2S-Temp, we can conclude that the performance of S2S-Temp improves with more unpaired data. Moreover, without unpaired data, our model is even worse than CVAE since the structured templates cannot be accurately estimated from such a few data, and as long as half of the unpaired data are available, the model outperforms the baseline models on most metrics. The results further verified the important role the unpaired data plays in learning of a response generation model from low resources. S2S-Temp is better than S2S-Temp-MLE, indicating that the adversarial learning approach can indeed enhance the relevance of re-

sponses regarding to messages.

Table 3 shows the results of human evaluation. In terms of all the three aspects, S2S-Temp is better than all the baseline models. The values of kappa are all above 0.6, indicating substantial agreement among the annotators. When the size of paired data is small, the basic Seq2Seq model tends to generate more generic responses. That is why the gap between S2S-Temp and Seq2Seq is much smaller on fluency than those on the other two aspects. With the latent variables, CVAE brings both content and noise into responses. Therefore, the gap between S2S-Temp and CVAE is more significant on fluency and relevance than that on richness. HTD can greatly enrich the content of responses, which is consistent with the results in (Wang et al., 2018), although sometimes the responses might be irrelevant to messages or ill-formed. ECM does not perform well on both automatic evaluation and human judgement.

#### 5.5 Case Study

To further understand how S2S-Temp leverages templates for response generation, we show two examples with the test data, one for question response generation in Table 4 and the other for sentiment response generation in Table 5, where subscripts refer to states of the NHSMMs. First, we can see that a template defines a structure for a response. By varying templates, we can have responses with different syntax and semantics for a message. Second, some states may have consistent functions across responses. For example, state 36 in question response generation may refer to pronouns, and “I’m” and “it was” correspond to the same state 23 in sentiment response generation. Finally, some templates provide strong syntactic signals to response generation. For example, the

<b>Message:</b>	真的假的? 我瘦了16斤 Really? I lost 17.6 pounds
<b>Responses:</b>	[你] <sub>36</sub> [瘦了] <sub>31</sub> [吗?] <sub>13</sub> [You] <sub>36</sub> [lost weight] <sub>31</sub> [?] <sub>13</sub>  [你 是 怎么] <sub>14</sub> [做到 的] <sub>42</sub> [?] <sub>26</sub> [How do you] <sub>14</sub> [make it] <sub>42</sub> [?] <sub>26</sub>  [真的 吗?] <sub>48</sub> [我] <sub>36</sub> [不 信] <sub>32</sub> [Really?] <sub>48</sub> [I] <sub>36</sub> [don't believe it] <sub>32</sub>

Table 4: Question response generation with various templates.

<b>Message:</b>	One of my favorite Eddie Murphy movies!
<b>Responses:</b>	[it 's] <sub>29</sub> [a brilliant] <sub>35</sub> [movie] <sub>37</sub> [i screamed] <sub>29</sub> [! !] <sub>1</sub> [it was] <sub>23</sub> [so underrated] <sub>14</sub> [!] <sub>31</sub> [honestly .] <sub>30</sub> [i 'm] <sub>23</sub> [so pumped] <sub>14</sub> [to watch it] <sub>31</sub> [yeah .] <sub>47</sub> [i was watching] <sub>48</sub>

Table 5: Sentiment response generation with various templates.

segmentation of “Really? I don’t believe it” given by the template (48, 36, 32) matches the parsing result “FRAG + LS + VP” given by stanford syntactic parser.

## 6 Conclusions

We study low-response response generation for open domain dialogue systems by assuming that paired data are insufficient for modeling the relationship between messages and responses. To augment the paired data, we consider transferring knowledge from unpaired data to response generation through latent templates parameterized as a hidden semi-markov model, and take the templates as prior in generation. Evaluation results on question response generation and sentiment response generation indicate that when limited pairs are available, our model can significantly outperform several state-of-the-art response generation models.

## Acknowledgement

We appreciate the valuable comments provided by the anonymous reviewers. This work is supported in part by the National Natural Science Foundation of China (Grand Nos. U1636211, 61672081, 61370126), and the National Key R&D Program of China (No. 2016QY04W0802).

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.
- Tilman Becker. 2002. Practical, template-based natural language generation with tag. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+ 6)*, pages 80–83.
- Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161.
- Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 759–768. ACM.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. [Learning phrase representations using RNN encoder-decoder for statistical machine translation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Hanjun Dai, Bo Dai, Yan-Ming Zhang, Shuang Li, and Le Song. 2016. Recurrent hidden semi-markov model.
- Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. [Question generation for question answering](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.
- Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.
- Mary Ellen Foster and Michael White. 2004. Techniques for text planning with xslt. In *Proceedings of the Workshop on NLP and XML (NLPXML-2004): RDF/RDFS and OWL in Language Technology*.
- Albert Gatt and Ehud Reiter. 2009. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron

- Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018a. [Universal neural machine translation for extremely low resource languages](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.
- Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018b. [Meta-learning for low-resource neural machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.
- Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Hui-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.
- Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association of Computational Linguistics*, 6:437–450.
- Chao Jiang, Hsiang-Fu Yu, Cho-Jui Hsieh, and Kai-Wei Chang. 2018. Learning word embeddings for low-resource languages by pu learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1.
- Katharina Kann, Johannes Bjerva, Isabelle Augenstein, Barbara Plank, and Anders Søgaard. 2018. Character-level supervision for low-resource pos tagging. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 1–11.
- Yoon Kim. 2014. [Convolutional neural networks for sentence classification](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.
- Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 994–1003.
- Jiwei Li, Alexander H Miller, Sumit Chopra, Marc’Aurelio Ranzato, and Jason Weston. 2016b. Learning through dialogue interactions by asking questions. *arXiv preprint arXiv:1612.04936*.
- Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016c. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.
- Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.
- Chin-Yew Lin. 2004. [Rouge: A package for automatic evaluation of summaries](#). *Text Summarization Branches Out*.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. 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*, pages 2122–2132.
- Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. [The Stanford CoreNLP natural language processing toolkit](#). In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. 2019. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. *arXiv preprint arXiv:1905.05644*.
- Kevin P Murphy. 2002. Hidden semi-markov models (hsmms). *unpublished notes*, 2.
- Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar encoder-decoder for neural conversation generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1329–1338.

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.
- Hao Peng, Ankur P Parikh, Manaal Faruqui, Bhuwan Dhingra, and Dipanjan Das. 2019. Text generation with exemplar-based adaptive decoding. *arXiv preprint arXiv:1904.04428*.
- Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.
- Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. End-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.
- Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301.
- Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *ACL*, pages 1577–1586.
- Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2016. Generating long and diverse responses with neural conversation models.
- Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. [From eliza to xiaoice: Challenges and opportunities with social chatbots](#). *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Van-Khanh Tran and Le-Minh Nguyen. 2018. Dual latent variable model for low-resource natural language generation in dialogue systems. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 21–30.
- Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney. 2014. Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages. In *Fifteenth Annual Conference of the International Speech Communication Association*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.
- Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017. Steering output style and topic in neural response generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2140–2150.
- Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2193–2203.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187.
- Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *AAAI*, pages 3351–3357.
- Chen Xing, Wei Wu, Yu Wu, Ming Zhou, Yalou Huang, and Wei-Ying Ma. 2018. Hierarchical recurrent attention network for response generation. In *AAAI*, pages 5610–5617.

- Stephanie Young, Milica Gasic, Blaise Thomson, and John D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.
- Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.
- Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. 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 (Volume 1: Long Papers)*, volume 1, pages 654–664.
- Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *AAAI*, pages 730–738.
- Walter Zucchini, Iain L MacDonald, and Roland Langrock. 2016. *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC.