World Wide Web (2019) 22:1427–1446
https://doi.org/10.1007/s11280-018-0598-6

Neural personalized response generation as domain adaptation

Wei-Nan Zhang[1] · Qingfu Zhu[1] · Yifa Wang[1] · Yanyan Zhao[1] · Ting Liu[1]

Received: 22 November 2017 / Revised: 26 April 2018 / Accepted: 24 May 2018 /
Published online: 18 June 2018

Abstract One of the most crucial problem on training personalized response generation models for conversational robots is the lack of large scale personal conversation data. To address the problem, we propose a two-phase approach, namely ***initialization then adaptation***, to first pre-train an optimized RNN encoder-decoder model (**LTS** model) in a large scale conversational data for general response generation and then fine-tune the model in a small scale personal conversation data to generate personalized responses. For evaluation, we propose a novel human aided method, which can be seen as a quasi-Turing test, to evaluate the performance of the personalized response generation models. Experimental results show that the proposed personalized response generation model outperforms the state-of-the-art approaches to language model personalization and persona-based neural conversation generation on the automatic evaluation, offline human judgment and the quasi-Turing test.

Keywords Personalized response generation · Conversation generation · Sequence to sequence learning · Domain adaptation

✉ Wei-Nan Zhang
wnzhang@ir.hit.edu.cn

Qingfu Zhu
qfzhu@ir.hit.edu.cn

Yifa Wang
yfwang@ir.hit.edu.cn

Yanyan Zhao
yyzhao@ir.hit.edu.cn

Ting Liu
tliu@ir.hit.edu.cn

[1] Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin City, Heilongjiang, China

1 Introduction

Conversational robot, which is also called conversational system, virtual agent or chatbot, etc, is an interesting and challenging research of artificial intelligence. It can be applied to a large number of scenarios of human-computer interaction, such as question answering, negotiation, e-commence, tutoring, etc. Conversational robot usually plays the role of virtual companion or assistant of human [6]. For example, the virtual assistant on mobile phone is one of the most popular application of conversational robot, such as, Apple Siri,[1] Microsoft Cortana,[2] Facebook Messenger,[3] Google Assistant,[4] etc. Recently, a Twitter bot, which is called DeepDrumpf,[5] can mimic to post tweets[6] and reply the comments from other users in Twitter using the Donald Trump-like language style. It is trained by using a recurrent neural network (RNN) model on the large-scale data of speech transcripts, tweets, and debate remarks from Donald Trump and thus can be seen as his personalized model for posting tweets and replying comments.

For a same input message, responses with different personalities may lead to different topic evolution and, in some cases, user experiences in conversations. Table 1 shows an example of the responses of different personality to a same input message. From Table 1, we can see that the Response 1 is a briefly definite response to the input message. The Response 2 is full of emotion and the Response 3 provides another suggestion on dressing. Obviously, Response 2 and 3 are more likely to sustain the conversation, whereas response 1 may lead to an early close. Moreover, the conversational robots which are learnt from the conversation data like response 2 or 3 may bring a better experience to users. In addition, besides the conversation generation, capturing human's personality is also important in personalized recommendation [8, 28, 29, 47, 50].

However, one of the most crucial problem for training a personalized response generation model for a conversational robot is the lack of large scale personal conversation data. To address the problem, in this paper, we proposed a two-phase approach, namely ***initialization then adaptation***, to generate personalized response. Concretely, the proposed model is first pre-trained on a large scale data of general single-turn conversations and then fine-tuned on a small scale personal conversation data. Moreover, to address the problem of generating generic, vague or non-committal responses, such as "*I don't know*", "*Me, too*", etc., of the vanilla RNN based encoder-decoder model [1], we proposed a responding quality optimization scheme, which is called **Learning to Start (LTS)** model, to generate relevant and diverse responses. The contributions of this paper are three-fold:

- We proposed a two-phase approach, namely *initialization then adaptation*, to learn to generate personalized responses for conversational robots.
- We proposed a quasi-Turing test method to evaluate the personalized response generation of conversational robots.
- The proposed approach outperforms the state-of-the-art approaches of language model personalization and persona-based neural conversation generation.

[1] https://en.wikipedia.org/wiki/Siri

[2] https://en.wikipedia.org/wiki/Cortana_(software)

[3] https://www.messenger.com/

[4] https://assistant.google.com/

[5] https://twitter.com/deepdrumpf

[6] Here, a tweet is a message sent using Twitter.

Table 1 An example of the responses of different personality to a given input message

Input	Is it a proper dress for the first date?
Response 1	Yep.
Response 2	Honey, it is very suitable!
Response 3	It is better to wearing a silk scarf.

2 Related work

In this paper, we focus on the use of neural network approach for personalized response generation in open domain conversation systems. The related work includes three parts.

2.1 Open domain conversation generation

Open domain conversation is also called non-task-oriented dialogue or chitchat etc. [30] proposed an unsupervised approach to modeling dialogue response by clustering the raw utterances. They then presented an end-to-end dialogue response generator by using a phrase-based statistical machine translation model [31]. Xing et al. [3] introduced a search-based system, namely IRIS, to generate dialogues using vector space model and then released the experimental corpus for research and development [2]. Recently, benefit from the advantages of the neural sequence to sequence learning framework with neural networks [34, 37] and [36] had drawn inspiration from the neural machine translation [1, 10] and proposed an RNN encoder-decoder based approach to generate dialogue by considering the last one sentence and a larger range of context respectively. [33] presented a hierarchical neural network, which is inspired by [35], to build an end-to-end dialogue system. Fleiss [16] focused on resolving the generating of safe, commonplace, high frequency responses on the neural sequence to sequence model. Luan et al. [19] proposed to integrate role-based information and global topic context into an RNN (LSTM unit) based conversational model. Recently, Li et al. [18] captured the advantages of the RNN encoder-decoder on response generation and the deep reinforcement learning on the future rewarding to generate context-aware dialogues. Mei et al. [24] proposed a dynamic attention mechanism based language model with topic reranking for conversation generation.

2.2 Task-oriented dialogue generation

As concluded by [26], previous research on task-oriented dialogue generation mainly focused on defining the generation decision space with the handcrafted features or statistical models. However, they often failed to scale dialogue generation to new domains. To address the domain transferring problem, the learning based approaches are proposed. Mairesse et al. [22] proposed a statistical language generator which used a dynamic Bayesian networks to generate responses in dialogue. Mairesse and Young [21] learned to generate paraphrases in dialogue through a factored language model that was training from the data collected by crowdsourcing. Both of them are data-driven approaches and thus easy to transfer the application domains. Neural network approaches show amazing results on dialogue generation, Wen et al. [42] proposed a statistical dialogue generator based on a joint recurrent and convolutional neural network, which can directly learn from the data without any semantic alignment or handcrafted rules. Further, Wen et al. [43, 44] proposed a

semantically conditioned LSTM to generate dialogue response and then compared it with an RNN encoder-decoder generator on multi-domain data to verify the ability of domain adaptation of the two generators. Recently, Marjan et al. [23] proposed an end-to-end framework with grounded knowledge base for generating task-oriented conversations without slot filling.

2.3 Personalized response generation

The personalized response generation can be applied to either the task-oriented dialogue systems or the open domain conversation systems. Kim et al. [15] utilized a personal knowledge base and explored user interests to rank the responses in dialogue system. Bang et al. [4] proposed an example based approach to extend the input message and utilized a personal knowledge base for responses ranking in open domain conversation systems. Casanueva et al. [9] proposed an approach to automatically gathering dialogue data from similar speakers to improve the performance of personalized dialogue policy learning. Genevay and Laroche [12] presented a source selection approach and a transition selection approach to overcome the cold start problem for the new coming users of spoken dialogue systems. Mo et al. [26] proposed a personalized POMDP [48] model using transfer learning for policy optimization of task-oriented dialogue systems.

Recently, Li et al. [17] proposed a persona-based neural conversation model, which is the state-of-the-art model on neural personalized conversation generation. Luan et al. [20] took the seq2seq model and autoencoder model for response generation as two tasks and proposed a multi-task learning framework for speaker role adaptation. Wang et al. [40] proposed to use small scale style data and a topic embedding model to restrict the style and topic of generated responses. Yang et al. [46] presented a similar framework with our approach, but proposed a new adaptation mechanism by using reinforcement learning. In this paper, we take these models as our baselines for personalized response generation.

3 The proposed approach

The RNN based sequence to sequence (Seq2Seq) model is widely used to automatically generate responses for conversational robots [16–18, 32–34, 36–38, 42–45]. It usually consist of two parts, namely the encoder and decoder. The encoder is to convert the input message into a vector which represents the semantic information of the input message. The decoder then generates a response according to the encoding vector.

In the proposed approach, the RNN based Seq2Seq model with an optimized first token decoding scheme is chosen as the basic response generation unit. We then proposed a two-phase approach to generate personalized responses. As a general view, Figure 1 is the framework of the proposed approach. As can be seen, the proposed approach consists of two components, namely *initialization then adaptation*, the first one is used to pre-train the response generation model on large scale general training data and the second one fine-tunes the model on a small scale of personalized training data.

3.1 Initialization

Typically, the encoder and decoder are implemented by the GRU [10] or LSTM [14] based RNN. The encoder reads the input sentence word by word and outputs the hidden state

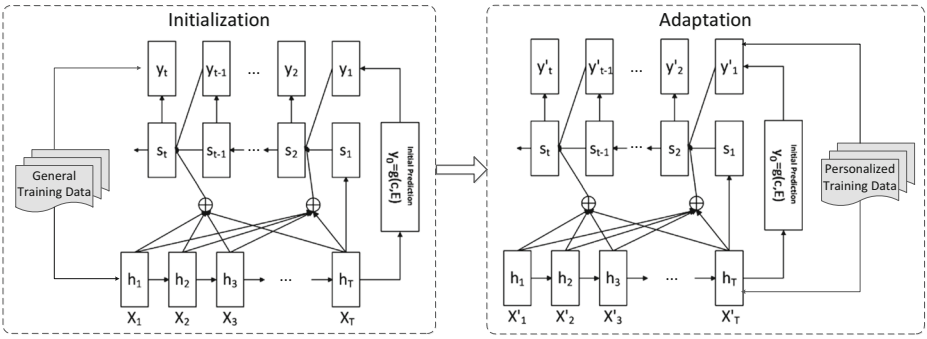


Figure 1 The framework of the proposed approach

of each word. These states are denoted as H which is also called annotations. Here, h_i represents the hidden state at time i and it is computed by its last hidden state h_{i-1} and the input word at time i, X_i. Therefore, the hidden state at time t can be denoted as:

$$h_t = f(h_{t-1}, X_t); \quad H = \{h_1, h_2, ..., h_T\} \tag{1}$$

Here, T equals to the length (the number of words) of the input sentence and f is a non-linear function which can be implemented as LSTM [14] or GRU [10].

The encoder then converts these hidden states to a context vector c as a summary of the semantic information of the input sentence.

$$c = q(\{h_1, h_2, ..., h_T\}) \tag{2}$$

Where, c can be implemented in many ways, for instance [37] set $c = h_T$.

For the decoding process, s_i denotes the hidden state at time i. It is also computed by a non-linear function f, of which the variables are the output y_{i-1} and the hidden state s_{i-1} at last time. The hidden state of the decoder at time t is computed as:

$$s_t = f(s_{t-1}, y_{t-1}) \tag{3}$$

Note that the context vector c, which is generated from the encoder, is also used to initialize the first hidden state [37] or all of the hidden states [1] of the decoder to make sure that the decoder can be conditioned by the encoder. Therefore, the hidden state of the decoder at time t is updated as:

$$s_t = f(s_{t-1}, y_{t-1}, c) \tag{4}$$

The output of the decoder at the state s_t is to map to a distribution over the vocabulary by using the maxout activation function [13]

In this paper, we utilize a weighted sum scheme [1] to dynamically compute the c_i for each state in the encoding process as:

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j \tag{5}$$

The weight α_{ij} of each hidden state h_j is computed as:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})} \tag{6}$$

Where, $e_{ij} = a(s_{i-1}, h_i)$ is a feedforward neural network, which can be called as the alignment model or attention model [1, 7].

3.2 Responding quality optimization (LTS model)

Through observing the responses generated by the RNN encoder-decoder model, we found another problem that when the first token is decoded to a high frequency word in the vocabulary, such as *"We", "I", "Yes"*, etc, it is tend to generate vague or non-committal responses. This problem is caused by the intrinsic generation scheme of the RNN encoder-decoder model, as it uses a special character "</s>" to generate the first word in decoding process. However, "</s>" could not provide any learnable information for the decoding process.

To address the above problem, we proposed a learning scheme to generate the first token in decoding process, namely Learning to Start (LTS) model. Unlike the classic RNN encoder-decoder model, the LTS model is an independent feedforward neural network that is proposed to specially predict the first token using the context vector that is generated from the encoding process. The LTS model can be represented as follows:

$$y_0 = \sigma((\sigma(W_i c) + b_i)E + b_e) \tag{7}$$

Here, c is the context vector which is computed by (5). E represents the word embedding matrix of the decoder, b_i and b_e are bias items. W_i is a learnable matrix that is trained to model the conditional dependence of the context vector c and the first word in decoding process.

By ignoring the bias items, the (7) can be transformed as follows:

$$y_0 = g(c, E) \tag{8}$$

We thus found that the LTS is to model the relation between the context vector c and the embedding matrix E of the decoder. According to the distribution of the generation probability over the decoding vocabulary, LTS predicts the first token for the decoder and the decoding process goes on until the finish of generating a response.

3.3 Adaptation

Due to the lack of personal conversation data for training personalized response generation model, we first train the neural response generation model in a large scale general conversation data, which is collected from Chinese online forums and totally includes 1,154,268 one-to-one post (input message) and response pairs.[7] 1.15 million one-to-one post and response pairs are used for the general training and the vocabulary contains 35 thousand tokens. We then fine-tune the general response generation model by using a small scale of personal conversation data to make the pre-trained model adapt to generate personalized responses. For adaptation, we invited 5 volunteers, each of which shared 2,000 messages of their chatting history from the use of instant messaging service without any privacy information. Towards the size of general training data, the size of the personal conversation data is extremely small. Therefore, in the adaptation phase, all the initial parameters of the personalized response generation model are shared from the "*Initialization*" (Section 3.1) phase. Moreover, different vocabularies are used for encoding and decoding respectively to generate personalized responses. Here, taking the general training data as the source domain and the personalized training data as the target domains, the personalized response generation thus can be seen as a domain adaptation process.

[7]Here, one-to-one means one post is only corresponded to one response.

4 Experiments and analysis

4.1 Data

The 1.15 million post (input messages) and response pairs is used for training the proposed LTS model as a basic response generation unit. The rest 4,268 post and response pairs are used for the sampling of test set. As the proposed personalized response generation approach includes two phases, there are two separate training data sets, namely general training data and personalized training data (See Figure 1). We collected 2,000 single-turn conversation pairs from each volunteer. After training, we obtained 5 personalized responding models that are corresponding to the 5 volunteers, respectively, for the test. Note that the personal data is collected from the 5 volunteers for training the personalized responding models and they are also the corresponding volunteers in testing the performance of the personalized responding models. There is only one tester, who is familiar with the 5 volunteers and does not participate in collecting the training data, is asked to judge whether the responses are coming from the volunteers or not.

4.2 Parameter setting

The parameter settings in the response generation model are as follow: The dimension of the hidden layer of the RNN encoder and decoder model equals to 1,024. The dimension of the word embedding, which is obtained by using the word2vec toolkit [25], is tuned to 500. Here, the word2vec is trained on the SogouCS&CA corpus (2008 version),[8] which is widely used for Chinese text analysis [39, 49]. The size of SogouCS&CA dataset is 8.7GB. It contains 1,520,842,220 tokens and the vocabulary size is 1,354,247. The LTP[9] toolkit is used to Chinese word segmentation for all the data. The encoder-decoder framework is implemented by using Theano toolkit [5]. The batch size is set to 128. The iteration times are set to 10 and 8 for the general training and personalized training respectively.

4.3 Baselines

We choose 6 baselines for the empirical comparisons. The first 4 baselines are for personalized response generation, the last 2 baselines are for response generation.

- $\mathbb{LMP}$: the state-of-the-art approach for language model personalization, which is proposed by [41].
- $\mathbb{PCM}$: the state-of-the-art approach for persona-based neural conversation model, which is proposed by [17].
- $\mathbb{STM}$: the state-of-the art approach for style and topic based response generation, which is proposed by [40].
- $\mathbb{NPM}$: a neural personalized model with domain adaptation for conversation generation, which is the most relevant work proposed by [46].
- $\mathbb{NRM}$: the first neural responding machine for short-text conversation generation, which is proposed by [34].
- $\mathbb{DRL}$: the first deep reinforcement learning based approach for open domain dialogue generation, which is proposed by [18].

[8] http://www.sogou.com/labs/dl/cs.html

[9] http://www.ltp-cloud.com/

4.4 Evaluation

Automatic evaluation of response generation is still an open problem [34]. The BLEU score [27], which is widely used in machine translation, is not a suitable evaluation metric for response generation. As the responses to the same post may share less common words, it is impossible to construct a reference set with adequate coverage. Meanwhile, the Perplexity, which is an evaluation metric for language modeling, is also not reasonable for evaluating the relevance between post and response.

To address the above issues, we design a novel human aided quasi-Turing test method for evaluation. The diagram of the evaluation method is shown in Figure 2. The evaluation method includes a volunteer, a tester and a chatbot. The volunteer and the tester are communicating through an instant messaging service. Here, the tester is told to chitchat with a volunteer through the instant messaging service. Meanwhile, the tester do not know the existence of the chatbot in all the chatting. In a conversation, each message from the tester is sent to a volunteer and his/her chatbot simultaneously. The question mark "**?**" denotes that the volunteer needs to randomly decide whether to respond by himself/herself or let the chatbot sends its response. The *Shelter* in Figure 2 represents that the volunteer could not see the response that is generated by the chatbot before it is sent to the tester. We aim to reduce the preference of the volunteer to the response of the chatbot. When a conversation is finished, the tester is asked to judge whether each response is from the volunteer or someone else. We proposed the ***imitation rate***, r_{imi} to evaluate the personality of responses generated by the chatbots. Here, we use n_{imi} to denote the number of responses that are judged to be from a volunteer, but are generated by his/her chatbot in testing. n_{gr} is the total number of responses that are generated by the chatbot in testing. The *imitation rate* is thus defined as:

$$r_{imi} = \frac{n_{imi}}{n_{gr}} \tag{9}$$

We can obviously see from (9) that the *imitation rate* can reflect the ability of the chatbot on imitating the personalized responding/language style of the volunteers. The larger the *imitation rate*, the better a chatbot imitates its corresponding volunteer.

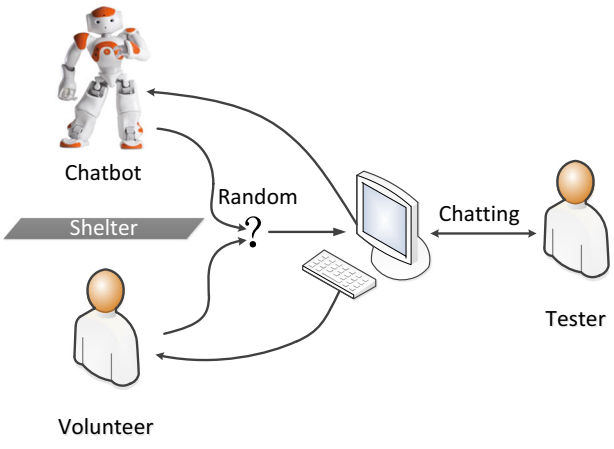


Figure 2 The quasi-Turing test method for evaluating personalized response generation. Note that the chatbot denotes to the corresponding personalized response generation model of the volunteer

4.5 Experimental results

4.5.1 Offline human judgment result

The offline human judgment is to evaluate the ability of the response generation models on imitating the personalized responding/language style of the volunteers. First, the tester provides 50 input messages for testing. Second, the messages are then respectively sent to the $\mathbb{LMP}$, $\mathbb{PCM}$, $\mathbb{STM}$, $\mathbb{NPM}$ and $\mathbb{OURS}$ to collect personalized responses. Note that for each approach, there are 5 personalized responding models, namely, $\mathbb{LMP}1 \sim 5$, $\mathbb{PCM}1 \sim 5$, $\mathbb{STM}1 \sim 5$, $\mathbb{NPM}1 \sim 5$ and $\mathbb{OURS}1 \sim 5$. Therefore, given the 50 input messages, for each volunteer, there are 5 groups of imitated (personalized) responses and each group contains 50 responses. For example, for volunteer #1, $\mathbb{LMP}1$, $\mathbb{PCM}1$, $\mathbb{STM}1$, $\mathbb{NPM}1$ and $\mathbb{OURS}1$ respectively generate 50 responses to imitate the responding/language style of the volunteer. Third, for each 5 groups of imitated responses, we ask the tester to judge whether a response is from the volunteer or someone else. Table 2 shows the offline judgment results. As can be seen, the proposed personalized responding models ($\mathbb{OURS}$) outperform the four baselines. It illustrates the generated responses by our proposed approach are more similar to the volunteers than the baseline approaches. Meanwhile, besides the imitation rate, we also ask 3 annotators to judge the quality of the generated responses by scoring them from 0 to 2. The average quality score of each model is shown in Table 2. We can see that although the imitation rates are quite different among these models, the average quality score is very close. It also reveals the average quality of neural generative conversation models based on the sequence to sequence framework.

4.5.2 Response similarity between volunteers and models

To verify the ability of the personalized response generation models on imitating the personalized responding style of the volunteers, we calculate the **cosine similarity**[10] of the responses generated by $\mathbb{LMP}1 \sim 5$, $\mathbb{PCM}1 \sim 5$, $\mathbb{STM}1 \sim 5$, $\mathbb{NPM}1 \sim 5$ and $\mathbb{OURS}1 \sim 5$ with the responses given by volunteers(V)1$\sim$5, respectively. For calculation, the 5 volunteers are also asked to provide their responses of the 50 input messages given by the tester. The generated responses by the $\mathbb{LMP}1 \sim 5$, $\mathbb{PCM}1 \sim 5$, $\mathbb{STM}1 \sim 5$, $\mathbb{NPM}1 \sim 5$ and $\mathbb{OURS}1 \sim 5$ are then used in this section.

Formally, the response similarity can be represented as $\cos(v_{\mathbb{LMP}i}, v_{\mathrm{Vi}})$, $\cos(v_{\mathbb{PCM}i}, v_{\mathrm{Vi}})$, $\cos(v_{\mathbb{STM}i}, v_{\mathrm{Vi}})$, $\cos(v_{\mathbb{NPM}i}, v_{\mathrm{Vi}})$ and $\cos(v_{\mathbb{OURS}i}, v_{\mathrm{Vi}})$, where $v_{\mathbb{LMP}i}$, $v_{\mathbb{PCM}i}$, $v_{\mathbb{STM}i}$, $v_{\mathbb{NPM}i}$ and v_{Vi} denote the vector representations of the responses generated by $\mathbb{LMP}$i, $\mathbb{PCM}$i, $\mathbb{STM}$i, $\mathbb{NPM}$i, $\mathbb{OURS}$i and Vi, respectively. Here, Vi indicates the i-th volunteer. Concretely, the each element of $v_{\mathbb{LMP}i}$, $v_{\mathbb{PCM}i}$, $v_{\mathbb{STM}i}$, $v_{\mathbb{NPM}i}$ and v_{Vi} equals to the frequencies of unigram or bigram that are counted from the corresponding responses, respectively. Figure 3 shows the results of the response similarity between volunteers and the response generation models.

As can be seen from Figure 3, in unigram similarity, the $\mathbb{PCM}$, $\mathbb{STM}$, $\mathbb{NPM}$ and $\mathbb{OURS}$ have close performance and they all outperform the $\mathbb{LMP}$. While, in bigram similarity, $\mathbb{OURS}$ outperforms the four baselines. It indicates that the proposed models can better capture the lexical characteristics of the volunteers than the baselines so that to generate more volunteer-like responses in conversations.

[10] https://en.wikipedia.org/wiki/Cosine_similarity

Table 2 The experimental results of the baseline models (LMP, PCM, STM, NPM) and the proposed personalized responding models (OURS) by human judgment

	LMP1	PCM1	STM1	NPM1	OURS1
n_{imi}	3	6	6	8	11
n_{gr}	50	50	50	50	50
r_{imi}	6%†	12%†	12%‡	16%†	**22%**
avg_q	0.53	**0.58**	0.54	**0.58**	**0.58**
	LMP2	PCM2	STM2	NPM2	OURS2
n_{imi}	5	8	10	8	10
n_{gr}	50	50	50	50	50
r_{imi}	10%†	16%†	**20%**	16%†	**20%**
avg_q	0.56	**0.57**	**0.57**	0.54	0.56
	LMP3	PCM3	STM3	NPM3	OURS3
n_{imi}	1	8	8	9	12
n_{gr}	50	50	50	50	50
r_{imi}	2%†	16%†	16%‡	18%†	**24%**
avg_q	0.54	0.57	0.55	0.57	**0.60**
	LMP4	PCM4	STM4	NPM4	OURS4
n_{imi}	4	13	15	13	16
n_{gr}	50	50	50	50	50
r_{imi}	8%†	26%†	30%‡	26%†	**32%**
avg_q	0.55	0.56	**0.58**	**0.58**	0.57
	LMP5	PCM5	STM5	NPM5	OURS5
n_{imi}	4	10	16	18	18
n_{gr}	50	50	50	50	50
r_{imi}	8%†	20%†	32%‡	**36%**	**36%**
avg_q	0.57	0.54	0.53	0.57	**0.59**

avg_q denotes the average quality score, which is judged by 3 annotators, of each model. † and ‡ denote that the results of our proposed models significantly outperform the results of the baselines in statistics with $p < 0.01$ and $p < 0.05$, respectively

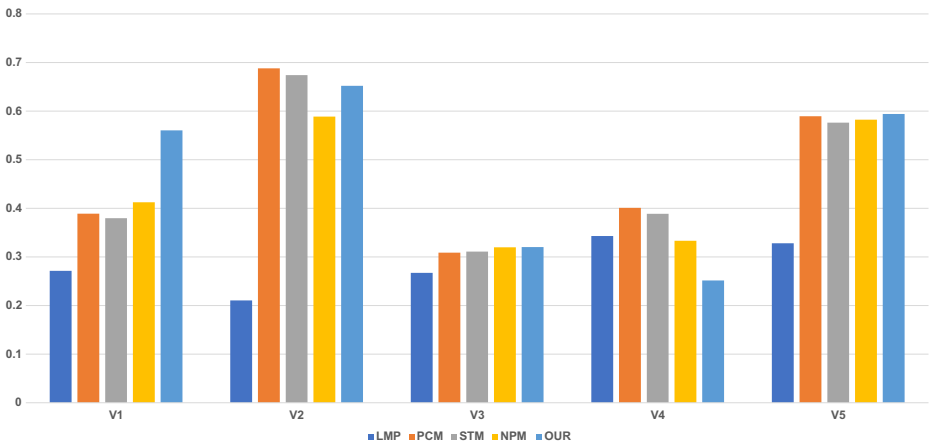
Values in bold denote the best performance on each corresponding evaluation metrics

Figure 4 shows the impact of number of samples used for adaptation on the performance of OURS1 ∼5.

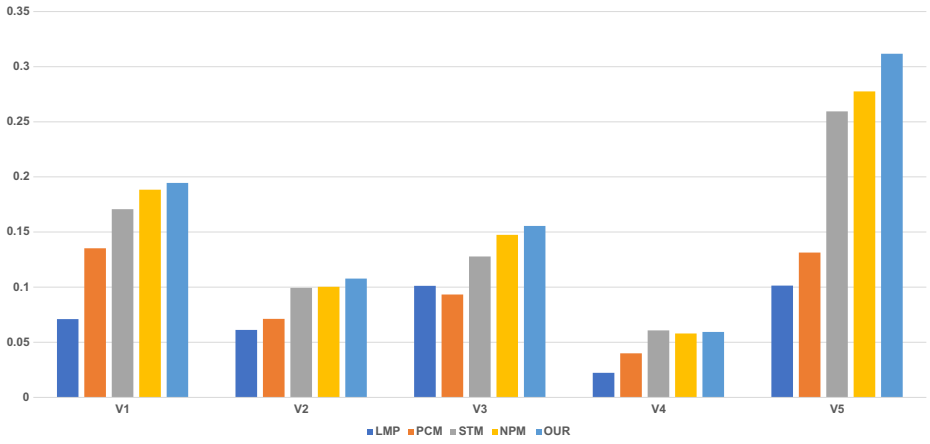
We can see from Figure 4 that OURS2 and OURS5 need less adaptation samples than other 3 models. The reason may be that the personalized data of V2 and V5 that are used for adaptation is more major-specialized than other 3 volunteers. Therefore, the lexical features of V2 and V5 are more distinguishable than other volunteers.

4.5.3 Quasi-turing test

We again test the imitating ability of the personalized response generation models through an online real-time conversation. We ask the tester to use the 50 input messages to chitchat



(a) Unigram Similarity

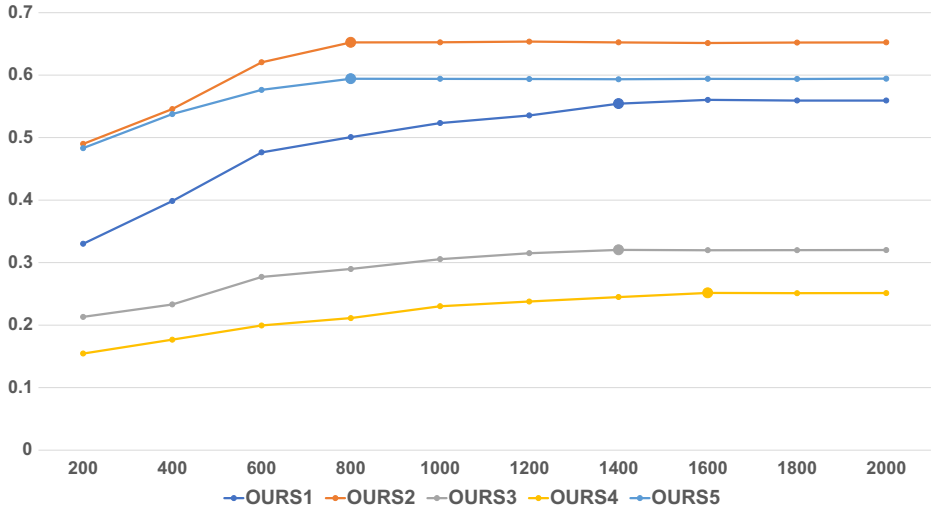


(b) Bigram Similarity

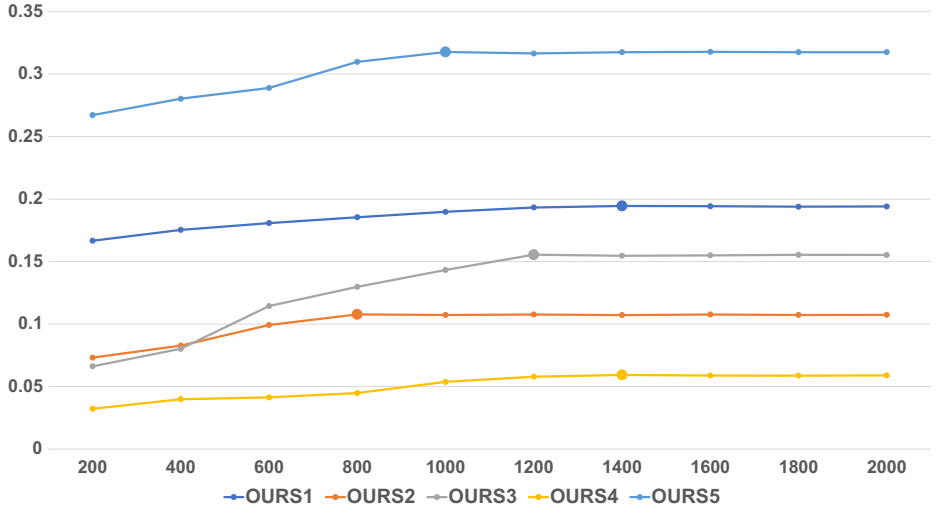
Figure 3 The unigram and bigram cosine similarity of the responses generated by $\mathbb{LMP}1 \sim 5$, $\mathbb{PCM}1 \sim 5$, $\mathbb{STM}1 \sim 5$, $\mathbb{NPM}1 \sim 5$, $\mathbb{OURS}1 \sim 5$ and the volunteers(V1∼5), respectively

with the volunteers online. For each input message, a response is randomly chosen from the 2 responses that are online generated by the chatbot and the volunteer. After the finish of each conversation, the tester is asked to judge whether each response in the conversation is from the volunteer or someone else. We also use *imitation rate* (See (9)) to evaluate the performance of the chatbots on imitating the personalized responding/language style of the volunteers. Table 3 shows the experimental results of personalized response generation by the proposed approach.

We can see from Table 3 that our proposed models outperform the four baselines in average imitation rate ($\text{Avg}_{r_{imi}}$). To compare the results from Tables 2 and 3, we can see that the average r_{imi} scores (26.8%) of the 5 personalized responding models ($\mathbb{OURS}1$-5) are lower than those of the corresponding average imitation rate (35.46%) in the quasi-Turing test as shown in Table 3. The reason is that in the quasi-Turing test, the responses generated by a chatbot are randomly mixed with a volunteer's responses in a conversation. Meanwhile, the process of the quasi-Turing test is context-aware. Therefore, due to the "coherent model"



(a) Unigram Similarity



(b) Bigram Similarity

Figure 4 The varying of unigram (**a**) and bigram (**b**) cosine similarity of the responses generated by $\mathbb{OURS}1$ ∼5 and the volunteers(V1∼5) on the number of samples used for adaptation. x-axis denotes the number of samples for adaptation, y-axis denotes the cosine similarity

in mind, the volunteers may tend to coordinate the chatbots to complete a conversation. That may increase the difficulty of the tester's judgment.

4.5.4 Diversity result of generated response

Besides the above subjective and objective evaluations, we also compare the diversity result of responses generated by these models. We utilized 4 objective evaluation metrics, namely

Table 3 The online real-time conversation results obtained by the judgment of the tester

	LMP1	LMP2	LMP3	LMP4	LMP5	$\text{Avg}_{r_{imi}}$
n_{gr}	27	26	23	28	30	
n_{vr}	23	24	27	22	20	
n_{test}	50	50	50	50	50	
n_{imi}	10	9	9	8	8	
r_{imi}	37.04%	34.62%	39.13%	28.57%	26.67%	33.21%
	PCM1	PCM2	PCM3	PCM4	PCM5	$\text{Avg}_{r_{imi}}$
n_{gr}	26	31	24	28	28	
n_{vr}	24	19	26	22	22	
n_{test}	50	50	50	50	50	
n_{imi}	9	10	7	10	9	
r_{imi}	34.62%	32.26%	29.17%	35.71%	32.14%	32.78%
	STM1	STM2	STM3	STM4	STM5	$\text{Avg}_{r_{imi}}$
n_{gr}	25	27	23	31	29	
n_{vr}	25	23	27	19	21	
n_{test}	50	50	50	50	50	
n_{imi}	8	9	8	10	10	
r_{imi}	32.00%	33.33%	34.78%	32.26%	34.48%	33.37%
	NPM1	NPM2	NPM3	NPM4	NPM5	$\text{Avg}_{r_{imi}}$
n_{gr}	32	27	28	24	28	
n_{vr}	18	23	22	26	22	
n_{test}	50	50	50	50	50	
n_{imi}	12	10	9	7	10	
r_{imi}	37.50%	37.04%	32.14%	29.17%	35.71%	34.31%
	OURS1	OURS2	OURS3	OURS4	OURS5	$\text{Avg}_{r_{imi}}$
n_{gr}	29	26	21	33	33	
n_{vr}	21	24	29	17	17	
n_{test}	50	50	50	50	50	
n_{imi}	11	9	8	13	9	
r_{imi}	37.93%	34.62%	38.10%	39.40%	27.27%	35.46%

n_{gr} and n_{vr} represents the number of responses that are generated by the chatbot and the volunteer respectively. n_{test} is the total number of input messages for testing. n_{imi} denotes the number of responses that are generated by the chatbot but are judged as the responses of the volunteer by the tester. r_{imi} denotes the imitation rate, which is defined in (9)

distinct-1 4, which are calculated by the ratios of unique unigram, bigram, trigram and four-gram. Taking the distinct-1 as an example, it equals to the number of distinct unigrams generated by a specific model divided by the total number of distinct unigrams generated by all the compared models. The experiment results are shown in Table 4.

Table 4 The diversity result of the generated responses of the baseline models (LMP, PCM, STM, NPM) and the proposed personalized responding models (OURS)

	LMP1†	PCM1†	STM1†	NPM1†	OURS1
distinct-1	0.32	0.36	0.31	0.34	0.38
distinct-2	0.75	0.78	0.75	0.78	0.83
distinct-3	0.83	0.88	0.82	0.87	0.91
distinct-4	0.88	0.92	0.88	0.90	0.97
	LMP2†	PCM2‡	STM2†	NPM2†	OURS2
distinct-1	0.30	0.33	0.29	0.31	0.35
distinct-2	0.68	0.70	0.67	0.69	0.78
distinct-3	0.73	0.77	0.71	0.75	0.87
distinct-4	0.78	0.81	0.77	0.79	0.88
	LMP3‡	PCM3‡	STM3†	NPM3†	OURS3
distinct-1	0.17	0.20	0.16	0.16	0.20
distinct-2	0.53	0.59	0.50	0.52	0.61
distinct-3	0.65	0.73	0.62	0.65	0.75
distinct-4	0.77	0.81	0.73	0.75	0.83
	LMP4†	PCM4†	STM4†	NPM4†	OURS4
distinct-1	0.17	0.19	0.17	0.19	0.21
distinct-2	0.48	0.52	0.50	0.52	0.56
distinct-3	0.57	0.61	0.55	0.60	0.65
distinct-4	0.65	0.68	0.62	0.67	0.71
	LMP5†	PCM5†	STM5†	NPM5†	OURS5
distinct-1	0.31	0.35	0.31	0.34	0.38
distinct-2	0.63	0.70	0.63	0.68	0.75
distinct-3	0.78	0.82	0.79	0.82	0.86
distinct-4	0.81	0.86	0.82	0.84	0.88

† and ‡ denote that the results of our proposed models significantly outperform the results of the baselines in statistics with $p < 0.01$ and $p < 0.05$, respectively

Table 5 The BLEU scores of the NRM, DRL and LTS for response generation

	BLEU-1	BLEU-2	BLEU-3
NRM	0.5283	0.0553	0.0013
DRL	0.5195	0.0674	0.0035
LTS	**0.5303**	**0.0816**	**0.0063**

Here, BLEU-1, BLEU-2 and BLEU-3 denote the unigram, bigram and trigram overlaps between the generated response and the reference, respectively

Values in bold denote the best performance on each corresponding evaluation metrics

Table 6 The evaluation results of manually assigning the quality scores of the generated responses by NRM, DRL and LTS

	0	1	2	Mean	Agreement
NRM	66%	17%	17%	0.51	0.230
DRL	67.3%	13.2%	**19.5%**	0.52	0.242
LTS	**59.7%**	21.7%	18.6%	**0.59**	0.206

The agreement is calculated by using the Fleiss' kappa [11]

Values in bold denote the best performance on each corresponding evaluation metrics

From Table 4, we can see that our model significantly outperforms all the baselines in the four evaluation metrics. It illustrates that in the generation of personalized responses, our proposed approach is less likely generating vague or generic responses than baselines.

4.5.5 Responding quality optimization result

To verify the effect of the proposed responding quality optimization scheme, namely Learning to Start (LTS), we empirically compare the performance of LTS and two baselines. They are neural responding machine [34], which we call it NRM for short, and deep reinforcement learning based dialogue generation model [18], which we call it DRL for short. All the

Table 7 The sampled cases of the same inputs with different responses generated by the vanilla RNN based encoder-decoder model for response generation (Vanilla) and 𝕆𝕌ℝ𝕊1-5 models

Input	什么编程语言最好?
	(What is the best programming language?)
Vanilla	我不知道。(I don't know.)
𝕆𝕌ℝ𝕊1	日语。(Japanese.)
𝕆𝕌ℝ𝕊2	通常来说是Java。(Generally speaking, it is Java.)
𝕆𝕌ℝ𝕊3	计算机语言。(The computer language.)
𝕆𝕌ℝ𝕊4	不用。(I don't need.)
𝕆𝕌ℝ𝕊5	PHP。(PHP.)
Input	Java和C哪个好?
	(Which is better, Java or C?)
Vanilla	都挺好的。(Both of them are good.)
𝕆𝕌ℝ𝕊1	那个。(That one.)
𝕆𝕌ℝ𝕊2	Java。(Java.)
𝕆𝕌ℝ𝕊3	好的。(It's okay.)
𝕆𝕌ℝ𝕊4	不知道。(I don't know.)
𝕆𝕌ℝ𝕊5	C。(C.)
Input	哪里可以找到妹子?
	(Where can I find a girl?)
Vanilla	我是。(I am.)
𝕆𝕌ℝ𝕊1	你要干什么? (What are you doing?)
𝕆𝕌ℝ𝕊 2	我找。(I will find.)
𝕆𝕌ℝ𝕊3	不知道。(I don't know.)
𝕆𝕌ℝ𝕊4	你要找? (Are you looking for?)
𝕆𝕌ℝ𝕊5	我的。(Mine.)

Table 8 The sample cases that generated by our proposed approach and baselines on test set

	Model 1	Model 2	Model 3	Model 4	Model 5
Input	昨天晚上下大雨。	这课好难啊！	什么地方可以租房子?	晚上几点睡最好?	北京到上海的火车票。
	It was a heavy rain last night.	The course is so difficult!	Where can I rent a house?	What is the best time to sleep?	Train tickets from Beijing to Shanghai.
LMP	晚上还好吧。	没事的话就好。	老师说这个我知道。	给我买么?	估计是一个人。
	It was okay at night.	It's all right.	Teacher said I knew this.	Buy for me?	It's about one person.
PCM	晚上7点204。	所以千万不要想太多。	如何么?	想实现什么?	估计是直接就第二。
	7 pm, 204.	So, don't think too much.	How is it?	What do you want to achieve?	Maybe, it's just the second.
STM	晚上么?	明天正式开始。	谢谢我。	晚上7点。	这些一共200。
	At night?	It will formally start tomorrow.	Thank me.	7 pm.	These are 200.
NPM	我不知道。	世界真小。	有事?	晚上还不会给。	学习吧。
	I don't know.	It's a small world.	What's the matter?	It won't be given at night.	Go to study.
OURS	学习吧。	这个是太难了。	我的中介给你。	8点。	我只知道携程是快速的。
	It was okay at night.	It's too difficult.	Give you my agency.	8 pm.	I only know Ctrip is fast.

three models are trained using the 1.15 million post and response pairs. 1,000 posts (with 43,008 responses) and 1,000 posts (with 42,422 responses) are randomly sampled from the dataset released by [34] as development set and test set, respectively. Both the development and test set have no overlap to the training data. It means that for 1 post, there are about 42 responses in average as the reference. We use the BLEU score [27] and manual labeling for evaluation. Tables 5 and 6 show the BLEU score and manual labeling results of response generation, respectively.

As can be seen, the proposed LTS model outperforms the NRM and DRL model in all the 3 BLEU scores. For the evaluation of manual annotation, we recruit 3 annotator to assign a score (0,1 or 2) for each generated response. Here, **0** means bad, which indicates that the generated responses have some errors in grammar or fluency, or they are not relevant to the post. **1** means neutral, which represents that the generated responses are fluent and have no grammar errors. Meanwhile, they are suitable responses in some particular scenario. **2** means good, which denotes that the generated responses are quite appropriate to the post. They are also fluent and have no grammar errors. Moreover, the generated responses are independent to scenario.

We can see from Table 6 that the proposed LTS model outperforms the NRM and DRL model in the human evaluation. It illustrates that the LTS model can generate more fluent and relevant responses than the baselines. Meanwhile, we also find that DRL model trained on the experimental data generates more good and bad responses than LTS model.

4.5.6 Qualitative analysis and discussion

For qualitative analysis, Table 7 shows the sampled cases of the same inputs with different responses generated by the vanilla RNN based encoder-decoder model for response generation (Vanilla) [1] and $\mathbb{OURS}$1-5 models. The Vanilla is also trained in the 1.15 million one-to-one post and message pairs.

As we can see from Table 7, $\mathbb{OURS}$2 and $\mathbb{OURS}$5 are good at responding the messages in the programming topic. It is because the background of volunteer 2 and 5 is computer science. There are a lot of content about programming, algorithm, database, etc, in their personal conversation data. For the third sampled conversation, $\mathbb{OURS}$2 and $\mathbb{OURS}$5 generate a generic response, which is quite close to the response generated by Vanilla. It reveals that the proposed personalized responding models can effectively capture the personality of responding/language style and generate personalized responses. However, when an input message is out of domain (a special language style), the personalized responding model tend to respond as a general neural response generation model (the Vanilla). It also illustrates that the proposed model can adopt the advantages of the general neural response generation models in personalized response generation.

Furthermore, we also randomly sample responses generated by baselines and our proposed model for qualitative analysis as shown in Table 8.

Here, Model 1 to 5 denote the models that trained on the personalized data of the corresponding volunteers. From Table 8, we can see that the responses generated by our proposed model are more fluent and readable than those generated by the baselines.

5 Conclusion and future work

In this paper, we proposed a two-phase approach, namely *initialization then adaptation*, to generate personalized responses for conversational robots. The proposed approach is

first pre-trained on a large scale general single-turn conversation data and then fine-tuned on a small scale personal conversation data. Taking the general conversation data as the source domain and the personal data as the target domain, the proposed approach thus can be seen as a domain adaptation process. The proposed personalized response generation framework can partially overcome the shortage of the lack of personal conversation data for training and fully adopt the advantages of general neural response generation models. Meanwhile, we also proposed a novel human aided method to evaluate the ability of the personalized responding model for imitating the responding/language styles of the volunteers. Experimental results show that the proposed personalized responding model outperforms the state-of-the-art language model personalization and persona-based neural conversation model on the automatic evaluation, offline human judgment and quasi-Turing test.

In future, we first plan to explore the user profiling information for the personalized response generation. Second, we plan to design an evaluation method to directly compare the performance of different models in the online real-time conversation.

Acknowledgements This paper is supported by NSFC (No. 61502120, 61472105, 61772153) and Heilongjiang philosophy and social science research project (No. 16TQD03).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. Computer Science (2014)
2. Banchs, R.E.: Movie-dic: A movie dialogue corpus for research and development. In: ACL (2012)
3. Banchs, R.E., Li, H.: Iris: A chat-oriented dialogue system based on the vector space model. In: ACL, pp. 37–42 (2012)
4. Bang, J., Noh, H., Kim, Y., Lee, G.G.: Example-based chat-oriented dialogue system with personalized long-term memory. In: ICBDSC, pp. 238–243 (2015)
5. Bergstra, J., Bastien, F., Breuleux, O., Lamblin, P., Pascanu, R., Delalleau, O., Desjardins, G., Warde-Farley, D., Goodfellow, I.J., Bergeron, A., Bengio, Y.: Theano: Deep learning on gpus with python. In: NIPS (2011)
6. Berry, P.M., Gervasio, M., Peintner, B., Yorke-Smith, N.: Ptime: Personalized assistance for calendaring. ACM Trans. Intell. Syst. Technol. (TIST) **2**(4), 40 (2011)
7. Bin, Y., Yang, Y., Shen, F., Xie, N., Shen, H.T., Li, X.: Describing video with attention-based bidirectional LSTM. IEEE Transactions on Cybernetics (2018)
8. Cao, J., Wu, Z., Mao, B., Zhang, Y.: Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. World Wide Web-internet Web Inf. Syst. **16**(5–6), 729–748 (2013)
9. Casanueva, I., Hain, T., Christensen, H., Marxer, R., Green, P.: Knowledge transfer between speakers for personalised dialogue management. In: SIGDD, pp. 12–21 (2015)
10. Cho, K., Merrienboer, B.V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation (2014)
11. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychol. Bull. **76**(5), 378 (1971)
12. Genevay, A., Laroche, R.: Transfer learning for user adaptation in spoken dialogue systems. In: ICAAMS, pp. 975–983 (2016)
13. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: ICML, pp. 1319–1327 (2013)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
15. Kim, Y., Bang, J., Choi, J., Ryu, S., Koo, S., Lee, G.G.: Acquisition and use of long-term memory for personalized dialog systems. In: MAAAHMI, pp. 78–87 (2014)
16. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. NAACL (2015)
17. Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., Dolan, B.: A persona-based neural conversation model. In: ACL, pp. 994–1003 (2016)

18. Li, J., Monroe, W., Ritter, A., Dan, J.: Deep reinforcement learning for dialogue generation, pp. 1192–1202 (2016)
19. Luan, Y., Ji, Y., Ostendorf, M.: Lstm based conversation models (2016)
20. Luan, Y., Brockett, C., Dolan, B., Gao, J., Galley, M.: Multi-task learning for speaker-role adaptation in neural conversation models. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1, Long Papers). Asian Federation of Natural Language Processing, pp. 605–614. Taipei (2017)
21. Mairesse, F., Young, S.: Stochastic language generation in dialogue using factored language models. Comput Linguis. **40**(4), 763–799 (2014)
22. Mairesse, F., Jurcicek, M., Ek, F., Keizer, S., Thomson, B., Yu, K., Young, S.: Phrase-based statistical language generation using graphical models and active learning. In: ACL, pp. 1552–1561 (2010)
23. Marjan, G., Chris, B., Ming-Wei, C., Bill, D., Jianfeng, G., Wen-tau, Y., Michel, G.: A knowledge-grounded neural conversation model (2018)
24. Mei, H., Bansal, M., Walter, M.: Coherent dialogue with attention-based language models (2017)
25. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. NIPS **26**, 3111–3119 (2013)
26. Mo, K., Li, S., Zhang, Y., Li, J., Yang, Q.: Personalizing a dialogue system with transfer learning (2016)
27. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: A method for automatic evaluation of machine translation. In: ACL, pp. 311–318 (2002)
28. Peng, M., Zeng, G., Sun, Z., Huang, J., Wang, H., Tian, G.: Personalized app recommendation based on app permissions. World Wide Web-internet Web Inf. Syst., 1–16 (2017)
29. Rasch, K., Li, F., Sehic, S., Ayani, R., Dustdar, S.: Context-driven personalized service discovery in pervasive environments. World Wide Web-internet Web Inf. Syst. **14**(4), 295–319 (2011)
30. Ritter, A., Cherry, C., Dolan, B.: Unsupervised modeling of twitter conversations. In: NAACL, pp. 172–180 (2010)
31. Ritter, A., Cherry, C., Dolan, W.B.: Data-driven response generation in social media. In: EMNLP, pp. 583–593 (2011)
32. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A., Pineau, J.: Hierarchical neural network generative models for movie dialogues (2015)
33. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. Computer Science (2016)
34. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. In: ACL, pp. 1577–1586 (2015)
35. Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., Nie, J.Y.: A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In: CIKM (2015)
36. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversational responses. In: NAACL, pp. 196–205 (2015)
37. Sutskever, I., Vinyals, O., Le, Q.V., Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. NIPS **4**, 3104–3112 (2014)
38. Vinyals, O., Le, Q.: A neural conversational model. Computer Science (2015)
39. Wang, C., Zhang, M., Ma, S., Ru, L.: Automatic online news issue construction in web environment. In: WWW, pp. 457–466 (2008)
40. Wang, D., Jojic, N., Brockett, C., Nyberg, E.: Steering output style and topic in neural response generation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 2140-2150. Copenhagen (2017)
41. Wen, T.H., Heidel, A., Lee, H.Y., Tsao, Y., Lee, L.S.: Recurrent neural network based language model personalization by social network crowdsourcing. In: INTERSPEECH, pp. 2703–2707 (2013)
42. Wen, T.H., Gasic, M., Kim, D., Mrksic, N., Su, P.H., Vandyke, D., Young, S.: Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. SIGDial (2015)
43. Wen, T.H., Gasic, M., Mrksic, N., Su, P.H., Vandyke, D., Young, S.: Semantically conditioned lstm-based natural language generation for spoken dialogue systems. EMNLP (2015)
44. Wen, T.H., Gašić, M., Mrkšić, N., Rojas-Barahona, L.M., Su, P.H., Vandyke, D., Young, S.: Multi-domain neural network language generation for spoken dialogue systems. In: NAACL, pp. 120–129 (2016)
45. Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W.Y.: Topic augmented neural response generation with a joint attention mechanism. arXiv:1606.08340 (2016)

46. Yang, M., Zhao, Z., Zhao, W., Chen, X., Zhu, J., Zhou, L., Cao, Z.: Personalized response generation via domain adaptation. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1021–1024. ACM (2017)
47. Yao, W., He, J., Huang, G., Cao, J., Zhang, Y.: Personalized recommendation on multi-layer context graph. Lect. Notes Comput. Sci. **8180**, 135–148 (2013)
48. Young, S., Gasic, M., Thomson, B., Williams, J.D.: Pomdp-based statistical spoken dialog systems: A review. Proc. IEEE, 1160–1179 (2013)
49. Zhang, X., LeCun, Y.: Text Understanding from Scratch. Computer Science (2015)
50. Zhou, X., He, J., Huang, G., Zhang, Y.: A personalized recommendation algorithm based on approximating the singular value decomposition (approsvd). In: Ieee/wic/acm International Conferences on Web Intelligence and Intelligent Agent Technology, pp. 458–464 (2012)