Assigning Personality/Profile to a Chatting Machine for Coherent Conversation Generation

Qiao Qian[1], Minlie Huang[1*], Haizhou Zhao[2], Jingfang Xu[2], Xiaoyan Zhu[1]
[1] Conversational AI Group, AI Lab., Dept. of Computer Science, Tsinghua University
[1]Beijing National Research Center for Information Science and Technology, China
[2]Sogou, Inc. , Beijing 100084, China
qianqiaodecember29@126.com, aihuang@tsinghua.edu.cn

Abstract

Endowing a chatbot with personality is challenging but significant to deliver more realistic and natural conversations. In this paper, we address the issue of generating responses that are coherent to a pre-specified personality or profile. We present a method that uses generic conversation data from social media (without speaker identities) to generate profile-coherent responses. The central idea is to detect whether a profile should be used when responding to a user post (by *a profile detector*), and if necessary, select a key-value pair from the profile to generate a response forward and backward (by *a bidirectional decoder*) so that a personality-coherent response can be generated. Furthermore, in order to train the bidirectional decoder with generic dialogue data, *a position detector* is designed to predict a word position from which decoding should start given a profile value. Manual and automatic evaluation shows that our model can deliver more coherent, natural, and diversified responses.

1 Introduction

Generating human-level conversations by machine has been a long-term goal of AI since the Turing Test [Turing, 1950]. However, as argued by [Vinyals and Le, 2015], the current conversational systems are still unable to deliver realistic conversations to pass the Test. Amongst the many limitations, the lack of a coherent personality is one of the most challenging difficulties. Though personality is a well-defined concept in psychology [Norman, 1963; Gosling *et al.*, 2003], while in this paper, the personality of a chatbot refers to the character that the bot plays or performs during conversational interactions. In this scenario, personality settings include age, gender, language, speaking style [Walker *et al.*, 1997], level of knowledge, areas of expertise, and other explicit and implicit cues that may portray character [Shum *et al.*, 2018][1] .

A chatbot needs to present a coherent personality to gain confidence and trust from the user [Yu *et al.*, 2016]. Personality can make the chatbot easier to communicate with, more predictable and trustable, and therefore helps to establish an emotional connection with the user [Shum *et al.*, 2018]. Thus, generating responses that reflect a coherent personality is important for the chatbot [Güzeldere and Franchi, 1995].

General seq2seq model
User: Are you a boy or a girl? Chatbot: I am a boy. User: Are you a girl? Chatbot: Yes, I am a girl.
Our model with personality
User: Are you a boy or a girl? Chatbot: I am a handsome boy. User: Are you a girl? Chatbot: No, I am a boy.

Table 1: Exemplar conversations with/without coherent personality.

However, existing generation models are unable to demonstrate coherent personality, as exemplified in Figure 1. Recent works proposed in [Li *et al.*, 2016] and [Al-Rfou *et al.*, 2016] can handle ***implicit*** personality using user embeddings in response generation (projecting each user into a vector). However, such models can not assign an ***explicit*** profile to generate coherent responses. Moreover, these models need to be trained on dialogue data from many different users, which is expensive and has a sparsity issue: some users have very few dialogue data.

In this paper we define personality as a set of profile keys and values [2] and propose a model consisting of three key modules: a *profile detector* which detects whether a profile key and which key should be addressed, a *bidirectional decoder* that generates a response backward and forward from a selected profile value, and a *position detector* which predicts a proper word position at which a profile value can be replaced during the training of the decoder. The contributions of this paper are two-fold:

First, we address a novel problem of endowing a chatbot with an explicit personality or profile, which allows system developers to control the profile of a chatting machine specifically.
Second, instead of just learning subtle, implicit personality

* Corresponding Author: Minlie Huang.

[1]Though personality is a more abstract and broader concept, we use *personality/profile/identity* interchangeably in this paper.

[2]The profile keys include name, gender, age, location, constellation, etc.

from speaker-tagged dialogue data, our model deals with specific, explicit personality from generic dialogue data. Our model does not depend on speaker-tagged dialogue data which are more expensive and sparse.

2 Related Work

There has been a large amount of work for dialogue/conversation generation. These works can be categorized into task-oriented [Young *et al.*, 2013] or chat-based. Recently, researchers found that social data such as Twitter/Weibo posts and replies [Ritter *et al.*, 2011; Shang *et al.*, 2015], and movie dialogues can be used to learn and generate spoken language.

Large-scale conversation generation with social media data was firstly proposed in [Ritter *et al.*, 2011] and has been greatly advanced by applying sequence-to-sequence models [Sutskever *et al.*, 2014; Shang *et al.*, 2015; Serban *et al.*, 2016]. Many studies are focusing on improving the generation quality. These works include: dealing with unknown words [Gu *et al.*, 2016; Gulcehre *et al.*, 2016], avoiding universal responses [Jiwei Li, 2016], generating more diverse and meaningful responses [Mou *et al.*, 2016], and many more.

As argued by [Vinyals and Le, 2015], it's still quite impossible for current chatbots to pass the Turing Test, while one of the reasons is the lack of a coherent personality. Though personality has been well defined in psychology [Norman, 1963], it is implicit, subtle, and challenging to be revealed in language generation. Linguistic style can be an indicator of personality [Mairesse and Walker, 2006; Mairesse *et al.*, 2007], and conversation can be clues for personality recognition [Walker *et al.*, 1997; 2012]. In reverse, spoken language can be generated in accordance to particular personality [Mairesse and Walker, 2007].

A persona generation model can be seen in [Li *et al.*, 2016] where implicit speaker-specific conversation styles are represented by user embeddings. And a neural generative dialog model was proposed in [Kottur *et al.*, 2017] which conditioned on speakers as well as context history. Our work differs from this work significantly: our task is to endow the chatbot with an explicit personality while the previous works learn implicit persona. In other words, our task requires to generate responses that are coherent to the chatbot's pre-specified personality. Further, [Li *et al.*, 2016] and [Kottur *et al.*, 2017] requires many dialogue data from different users while our model is trained on generic dialogue data.

Another related work is generative question answering (GenQA) [Yin *et al.*, 2015] which generates a response containing an answer extracted from a knowledge base (KB). However, endowing a chatbot with personality is more than just question answering over KB, where there arise challenging problems such as semantic reasoning and conversation style modeling. Further, GenQA requires that the answer from KB must appear in the response to provide sufficient supervision while our work avoids the limitation by applying a position detector during training.

3 Model

3.1 Task Definition

The task can be formally defined as follows: given a post $X = x_1 x_2 \cdots x_n$, and an explicit profile defined as a set of key-value pairs $\{< k_i, v_i > | i = 1, 2, \cdots, K\}$, the task aims to generate a response $Y = y_1 y_2 \cdots y_m$ that is coherent to the profile. We consider several common attributes in personality including *name, gender, age, weight, location, and constellation*. The generation process can be briefly stated as below:

$$\begin{aligned} &\boldsymbol{P}(Y|X, \{< k_i, v_i >\}) \\ =&\boldsymbol{P}(z = 0|X) \cdot \boldsymbol{P}^{fr}(Y|X) \\ +&\boldsymbol{P}(z = 1|X) \cdot \boldsymbol{P}^{bi}(Y|X, \{< k_i, v_i >\}) \end{aligned} \tag{1}$$

where $\boldsymbol{P}(z|X)$ is the probability of using the profile given post X, which is computed by the *Profile Detector*; $\boldsymbol{P}^{fr}(Y|X) = \prod_{t=1}^{m} \boldsymbol{P}^{fr}(y_t|Y_{<t}, X)$ is given by a general forward decoder, the same as [Sutskever *et al.*, 2014], and $\boldsymbol{P}^{bi}(Y|X, \{< k_i, v_i >\})$ is given by a *Bidirectional Decoder*.

Note that post/response pair $< X, Y >$ is collected from social media, and the profile value may not occur in the response Y at all. This leads to the discrepancy between training and test, as described in the *Position Detector* section.

3.2 Overview

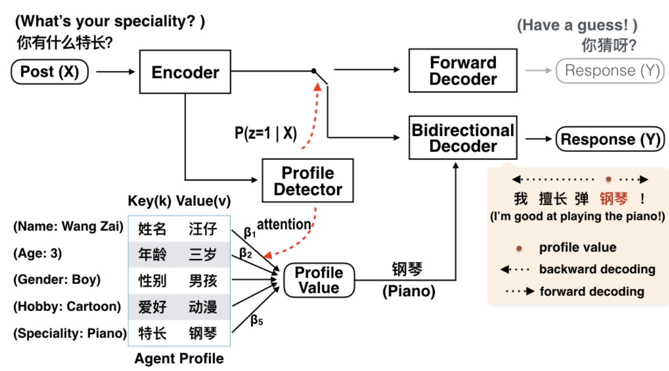

Figure 1: The overall process.

Our model works as follows (see Figure 1): given a post, the profile detector will predict whether the profile should be used. If not, a general seq2seq decoder will be used to generate the response; otherwise, the profile detector will further select an appropriate profile key and its value. Starting from the selected profile value, a response will be generated forward and backward by the bidirectional decoder. To train the bidirectional decoder on generic dialogue data (see Figure 2), the position detector predicts a word position from which decoding should start given the selected profile value. Note that the position detector will not be used during test.

3.3 Encoder

The encoder aims to encode a post to a vector representation. Given a post $X = x_1 x_2 \cdots x_n$, the hidden states of the post

$\boldsymbol{h_1}, \boldsymbol{h_2}, \cdots, \boldsymbol{h_n}$ are obtained by a gated recurrent unit (GRU) [Chung *et al.*, 2014], as follows:

$$\boldsymbol{h_t} = \mathbf{GRU}(\boldsymbol{h_{t-1}}, \boldsymbol{x_t}) \tag{2}$$

where $\boldsymbol{x_t}$ is the embedding of the t-th word x_t.

3.4 Profile Detector

The profile detector has two roles: first to detect whether the post should be responded with the profile, and second to select a specific $< key, value >$ to be addressed in the response. The **first role** of the profile detector is defined by the probability $\boldsymbol{P}(z|X)$ $(z \in \{0, 1\})$ where $z = 1$ means the profile should be used. For instance, if the post is *"how are you today"*, $\boldsymbol{P}(z = 1|X) \approx 0$, while if the post is *"how old are you"*, $\boldsymbol{P}(z = 1|X) \approx 1$.

$\boldsymbol{P}(z|X)$ is a binary classifier trained on labeled data. More formally, the probability is computed as follows:

$$\boldsymbol{P}(z|X) = \boldsymbol{P}(z|\widetilde{\boldsymbol{h}}) = \sigma(\boldsymbol{W_p}\widetilde{\boldsymbol{h}}) \tag{3}$$

where $\boldsymbol{W_p}$ is the parameter of the classifier and $\widetilde{\boldsymbol{h}} = \sum_j \boldsymbol{h_j}$, simply the sum of all hidden states, but other elaborated methods such as attention-based models are also applicable.

The **second role** of the profile detector is to decide which profile value should be addressed in a generated response. This is implemented as follows:

$$\beta_i = \mathbf{MLP}([\widetilde{\boldsymbol{h}}, \boldsymbol{k_i}, \boldsymbol{v_i}]) = softmax(\boldsymbol{W} \cdot [\widetilde{\boldsymbol{h}}; \boldsymbol{k_i}; \boldsymbol{v_i}]) \tag{4}$$

where $\boldsymbol{W}$ is the weight and $\boldsymbol{k_i}/\boldsymbol{v_i}$ is the embedding of a profile key/value respectively. $\widetilde{\boldsymbol{h}} = \sum_j \boldsymbol{h_j}$ is the representation of the post. β is a probability distribution over profile keys.

The optimal profile value is selected with the maximal probability: $\widetilde{v} = v_j \quad where\ j = argmax_i(\beta_i)$. As long as a profile value $\widetilde{v}$ is obtained, the decoding process will be determined by the bidirectional decoder, as follows:

$$\boldsymbol{P}^{bi}(Y|X, \{< k_i, v_i >\}) = \boldsymbol{P}^{bi}(Y|X, \widetilde{v}) \tag{5}$$

3.5 Bidirectional Decoder

This decoder aims to generate a response in which a profile value will be mentioned. Inspired by [Mou *et al.*, 2016], we design a bidirectional decoder which consists of a backward decoder and a forward decoder, but with a key difference that a position detector is employed to predict a start decoding position.

Suppose a generated response is $Y = (Y^b, \widetilde{v}, Y^f) = (y_1^b, \cdots, y_{t-1}^b, \widetilde{v}, y_{t+1}^f, \cdots, y_m^f)$ where $\widetilde{v}$ is a selected profile value. The bidirectional decoder will generate Y^b in a backward direction and Y^f forward. The backward decoder ($\boldsymbol{P^b}$) generates Y^b from the given profile value $\widetilde{v}$ to the start of the response. The forward decoder ($\boldsymbol{P^f}$)[3] generates Y^f from $\widetilde{v}$ to the end of the response, but takes as input the already generated first half, Y^b. The process is defined formally as follows:

$$\begin{aligned}
&\boldsymbol{P}^{bi}(Y|X, \widetilde{v}) = \boldsymbol{P}^b(Y^b|X, \widetilde{v}) * \boldsymbol{P}^f(Y^f|Y^b, X, \widetilde{v})) \\
&\boldsymbol{P}^b(Y^b|X, \widetilde{v}) = \prod_{j=t-1}^{1} \boldsymbol{P}^b(y_j^b|Y_{>j}^b, X, \widetilde{v}) \\
&\boldsymbol{P}^f(Y^f|Y^b, X, \widetilde{v}) = \prod_{j=t+1}^{m} \boldsymbol{P}^f(y_j^f|Y_{<j}^f, Y^b, X, \widetilde{v})
\end{aligned} \tag{6}$$

In order to encode more contexts in the forward decoder, the first half of a generated response (Y^b), along with the profile value ($\widetilde{v}$), serves as initial input to the forward decoder. The probability $\boldsymbol{P}^b$ and $\boldsymbol{P}^f$ is calculated via

$$\begin{aligned}
&\boldsymbol{P}^b(y_j^b|Y_{>j}^b, X, \widetilde{v}) \propto \mathbf{MLP}([\boldsymbol{s_j^b}; \boldsymbol{y_{j+1}^b}; \boldsymbol{c_j^b}]) \\
&\boldsymbol{P}^f(y_j^f|Y_{<j}^f, Y^b, X, \widetilde{v}) \propto \mathbf{MLP}([\boldsymbol{s_j^f}; \boldsymbol{y_{j-1}^f}; \boldsymbol{c_j^f}])
\end{aligned} \tag{7}$$

where $\boldsymbol{s_j^{(*)}}$ is the state of the corresponding decoder, $\boldsymbol{c_j^{(*)}}$ is the context vector, and $* \in \{b, f\}$ where b indicates the backward decoder and f the forward decoder. The vectors are updated as follows:

$$\begin{aligned}
\boldsymbol{s_j^{(*)}} &= \mathbf{GRU}(\boldsymbol{s_{j+l}^{(*)}}, [\boldsymbol{y_{j+l}^{(*)}}; \boldsymbol{c_j^{(*)}}]) \\
\boldsymbol{c_j^{(*)}} &= \sum_{t=1}^{n} \alpha_{j,t}^{(*)} \boldsymbol{h_t}
\end{aligned} \tag{8}$$

where $\alpha_{j,t}^{(*)} \propto \mathbf{MLP}([\boldsymbol{s_{j+l}^{(*)}}, \boldsymbol{h_t}])$ can be viewed as the similarity between decoder state $\boldsymbol{s_{j+l}^{(*)}}$ and encoder hidden state $\boldsymbol{h_t}$, $l = 1$ when $* = b$ (backward), and $l = -1$ when $* = f$ (forward). And these $\mathbf{MLP}$s have the same form as Eq. 4, but with different parameters.

3.6 Position Detector

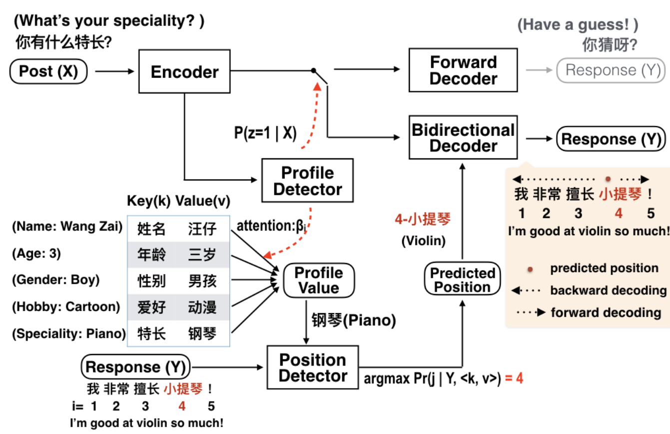

Figure 2: The training process of the model. Given a pair $< X, Y >$, the position detector will predict a position 小提琴(violin)-4 at which the profile value 钢琴(piano) can be replaced, and the position will be used to train the bidirectional decoder.

The position detector is designed to provide more supervision to the bidirectional decoder. It is designed to provide a

[3]Note that this decoder is different from $\boldsymbol{P}^{fr}(y_t|Y_{<t}, X)$.

start decoding position to the decoder during training. For instance, given a post $X =$"你(you)-1 有(have)-2 什么(what)-3 特长(speciality)-4 ? -5 (what's your speciality?)[4]" and a response $Y =$"我(I)-1 非常(very)-2 擅长(good_at)-3 小提琴(violin)-4(I am good at playing violin)", and a profile key value pair "<特长, 钢琴> ($< hobby, piano >$)", the position detector will predict that "小提琴(violin)-4" in the response can be replaced by the profile value "钢琴(piano)" to ensure grammaticality. The predicted position "小提琴(violin)-4" is then passed to the decoder (see Eq. 6) to signal the start decoding position.

As mentioned, the bidirectional decoder starts from a profile value to generate the entire sequence at the test stage. However, in our training dataset, the profile values may be rarely mentioned in the responses. For instance, given the profile key value pair <爱好, 冰球> ($< hobby, hockey >$), the value 冰球*(hockey)* rarely occurs in the training corpus. In other words, even though we have a training instance $(X, Y, < k, v >)$, the value (v) may not occur in Y at all. Hence, the bidirectional decoder is not aware from which word decoding should start. This leads to the discrepancy between training and test: during training, the decoder is unaware of the start decoding position but during test, the start decoding word is specified.

This issue makes our work differ substantially from previous approaches where supervision is directly observable either between post and response [Gu *et al.*, 2016] or between response and knowledge base [Yin *et al.*, 2015]. Experiments also show that the position detector contributes much to the performance improvement than a random position picking strategy [Mou *et al.*, 2016].

In order to find an appropriate position at which the profile value can be replaced, we need to estimate the probability: $\boldsymbol{P}(j|y_1 y_2 \cdots y_m, < k, v >)), 1 \leq j \leq m$ which indicates how likely the word y_j can be replaced by the profile value v. We apply a simple technique to approximate the probability: a word can be replaced by a given profile value if the word has the maximal similarity.

$$\boldsymbol{P}(j|Y, < k, v >)) \propto cos(\boldsymbol{y_j}, \boldsymbol{v}) \tag{9}$$

where $cos(\boldsymbol{y_j}, \boldsymbol{v})$ denotes the cosine similarity between a word in a response and a profile value. More elaborated techniques, for instance, language models, will be studied as future work.

3.7 Loss Function and Training

Two loss functions are defined: one on the generation probability and the other on the profile detector. The first loss is defined as below:

$$\begin{aligned}
&\mathcal{L}_1(\boldsymbol{\theta}, D^{(c)}, D^{(x,y)}) \\
&= - \sum_{(X,Y) \in D^{(c)} \cup D^{(pr)}} log\boldsymbol{P}(Y|X, \{< k_i, v_i >\}) \\
&= - \sum_{(X,Y) \in D^{(c)}} log\boldsymbol{P}^{fr}(Y|X) - \sum_{(X,Y) \in D^{(pr)}} log\boldsymbol{P}^{bi}(Y|X, \widetilde{v})
\end{aligned} \tag{10}$$

The first term is the negative log likelihood of observing $D^{(c)}$ and the second term for $D^{(pr)}$. $\widetilde{v}$ is a word in Y whose position is predicted by the position detector during training. $D^{(pr)}$ consists of pairs where a post is related to a profile key and its response gives a meaningful reaction to the post, and $D^{(c)}$ has only general post-response pairs.

The two decoders ($\boldsymbol{P}^{fr}$ and $\boldsymbol{P}^{bi}$) have no shared parameters. Since the number of instances in $D^{(c)}$ is much larger than that of $D^{(pr)}$, we apply a two-stage training strategy: $D^{(c)}$ will be used to train $\boldsymbol{P}^{bi}$ at the early stage for several epoches, where $\widetilde{v}$ is a randomly chosen word in a response, and then $D^{(pr)}$ for further training at the later stage.

The above formulation generally adopts the hard form of $\boldsymbol{P}(z|X)$ (see Eq. 3): $\boldsymbol{P}(z = 1|X) = 1$ for profile-related pairs and $\boldsymbol{P}(z = 1|X) = 0$ for others. In order to better supervise the learning of the profile detector, we define the second loss and add it to the first one with a weight α as the overall loss (i.e., $\mathcal{L} = \mathcal{L}_1 + \alpha\mathcal{L}_2$):

$$\begin{aligned}
&\mathcal{L}_2(\theta, D^{(pb)}, D^{(pr)}) \\
&= - \sum_{(X,Y,z) \in D^{(pb)}} log\boldsymbol{P}(z|X) - \sum_{(X,Y,\widehat{k}) \in D^{(pr)}} \sum_{j=1}^{K} \widehat{\beta}_j log\beta_j
\end{aligned} \tag{11}$$

where the first term is for binary prediction of using profile or not, and the second for profile key selection. $\widehat{k}$ is the profile key whose value should be addressed, K is the total number of keys, $\boldsymbol{\beta}$ is the predicted distribution over profile keys as defined by Eq. 4, and $\widehat{\boldsymbol{\beta}}$ is one-hot representation of the gold distribution over keys. $< X, Y, z >$ is obtained by manual annotation while $(X, Y, \widehat{k})$ is obtained by matching the keywords and synonyms in the profile with the post, which is noisy. This works well in practice and reduces manual labors largely.

4 Experiment

4.1 Data Preparation

We prepared several datasets[5]:

Weibo Dataset (WD) - D: We collected 9,697,651 post-response pairs from Weibo. The dataset is used for training $\boldsymbol{P}^{fr}(Y|X)$ and $\boldsymbol{P}^{bi}(Y|X, \widetilde{v})$ at the early stage and 10,000 pairs are used for validation to make early stop.

Profile Binary Subset (PB - $D^{(pb)} \in D$**)**: We extracted 76,930 pairs from WD for 6 profile keys ({*name, gender, age, location, weight, constellation*}) with about 200 regular expression patterns. The dataset is annotated by 13 annotators. Each pair is manually labeled to *positive* if a post is asking for a profile value and the response is a logic reaction to the post, or *negative* otherwise.

This dataset is used to train the binary classifier ($\boldsymbol{P}(z|X)$) (see $D^{(pb)}$ in Eq. 11). 3,000 pairs are used for test and the remainder for training.

Profile Related Subset (PR - $D^{(pr)} \in D^{(pb)}$**)**: This dataset

[4]The number indicates the position of each word.

[5]The data are available at: `http://coai.cs.tsinghua.edu.cn/hml/dataset/#AssignPersonality`

only contains pairs whose posts are positive in PB. In total, we have 42,193 such pairs. This dataset is used to train the bidirectional decoder.

Manual Dataset (MD): This dataset has 600 posts written by 4 human curators, including 50 negative and 50 positive posts for each key. A positive post for a profile key (e.g., *how old are you?*) means that it should be responded by a profile value, while a negative post (e.g., *how are you today?*) should not. This dataset is used to test the performance on real conversation data rather than social media data.

4.2 Experimental Settings

In our experiments[6], the encoder and decoders are all have 4 layers of GRUs with a 512-dimensional hidden state. The dimension of word embedding is set to 100. The vocabulary size is limited to $40,000$. The word embeddings are pre-trained on an unlabeled corpus (about $60,000,000$ Weibo posts) with word2vec. And the other parameters are initialized by sampling from a uniform distribution $U(-sqrt(3/n), sqrt(3/n))$, where n is the dimension of parameters. Training is conducted by stochastic gradient descent (SGD) with a mini-batch of 128 pairs. The learning rate is initialized with 0.5 and the decay factor is 0.99.

4.3 Human Evaluation

We evaluated our model at both post and session level. At the post level, three metrics (*naturalness, logic, and correctness*) are defined to evaluate the response generated by each model. At the session level, the models are evaluated from the aspects of *consistency and variety* to justify the performance in the real conversational setting.

We named our model *Personality-Coherent Conversation Machine (PCCM)* and compared it with several baselines:

Seq2Seq: a general sequence to sequence model [Sutskever *et al.*, 2014].

Seq2Seq + Profile Value (+PV): if the profile detector decides that a profile value should be used ($\boldsymbol{P}(z|X) > 0.5$), the response is simply the value of the key decided by the profile detector (see Eq. 4); otherwise, a general seq2seq decoder will be used.

Seq2Seq + Profile Value Decoding (+PVD): the response is generated by a general seq2seq decoder which starts decoding forwardly from the value of the selected key.

PCCM-Pos: Instead of using a predicted position obtained by the position detector to start the decoding process, the bidirectional decoder in this setting randomly picks a word in a response during training, the same as [Mou *et al.*, 2016].

Post-level Evaluation

To conduct post-level evaluation, we used 600 posts from MD, 50 positive/negative posts respectively for each key. Each post is input to all the models to get the corresponding responses. Thus, each post has 5 responses and these responses are randomly shuffled and then presented to two curators. Post-response pairs are annotated according to the following metrics, based on a 1/0 scoring schema:

Naturalness (Nat.) measures the fluency and grammaticality of a response. Too short responses will be judged as lack of naturalness.

Logic measures whether the response is a logical reaction to a post. For instance, for post "how old are you", a logical response could be "I am 3 years old" or "I do not know".

Correctness (Cor.) measures whether the response provides a correct answer to a post given the profile. For instance, for post "how old are you", if the profile has a key value pair like $< age, 3 >$, responses like "I am 18" will be judged as wrong.

Each response is judged by two curators. The Cohen's Kappa statistics are 0.46, 0.75 and 0.82 for naturalness, logic, and correctness respectively. Naturalness has a rather lower Kapp because it is more difficult to judge.

Method	Nat.	Logic	Cor.
Seq2Seq	71.4%	38.7%	22.3%
Seq2Seq +PV	85.4%	51.3%	40.2%
Seq2Seq +PVD	84.7%	51.1%	40.3%
PCCM-Pos (ours)	87.4%	50.0%	41.8%
PCCM (ours)	**88.9%**	**55.9%**	**44.2%**

Table 2: Evaluation of responses to the 600 posts from MD.

Chinese	English(Translated)
U:你还没说你几岁呢	U:You haven't told me your age.
S:我三岁了	S:I'm three years old.
U:你今年有15了不	U:Are you 15 years old or not?
S:我还没到呢	S:I'm not yet.
U:你多大啦	U:How old are you?
S:3岁了	S:Three years old.

Table 3: Samples of consistent conversations generated by our model. U/S indicates User/System, respectively.

Results in Table 2 support the following statements: **First**, our model is better than all other baselines in all metrics, indicating that our model can generate more natural, logical, and correct responses; **Second**, in comparison to simply responding with a profile value (Seq2Seq+PV) where the responses are generally too short, our model can generate more natural responses; **Third**, the position detection contributes to better generation, in comparison to a random position (PCCM vs. PCCM-Pos).

Session-level Evaluation

In order to compare these models in real conversation sessions, we randomly generated sessions based on MD. For each profile key, we randomly chose 3 *positive* posts[7] from MD, generated responses to the 3 posts for each model, and obtained a session of 3 post-response pairs. In this way, 240 sessions are generated, and each key has 40 sessions. The sessions are manually checked with the following metrics:

Consistency measures whether there is a response contradictory to the given profile. Score 1 indicates that all the three responses are consistent to the profile, and score 0 otherwise.

[6]The code is available at: `https://github.com/qianqiao/AssignPersonality.git`.

[7]A *positive* post must be responded with a profile value.

Variety measures the language variety of the three responses in a session. Score 1 indicates that the linguistic patterns and wordings are different between any two of them, and score 0 otherwise.

Results are shown in Table 4 and we presented some session examples in Table 3. We can clearly see the following observations:

1) Our model is remarkably better than all the baselines w.r.t both metrics. Results of our model against Seq2Seq+PVD indicate that the bidirectional decoder can generate responses of much richer language variety. The results of PCCM-Pos show that the position detector improves consistency and variety remarkably.

2) if simply responding with a profile value (Seq2Seq+PV), the model can obtain good consistency but very bad language variety, which is in line with the intuition.

3) The general Seq2Seq model is too weak to generate consistent or linguistically various responses.

Method	Consistency	Variety
Seq2Seq	2.1%	1.6%
Seq2Seq +PV	58.3%	2.1%
Seq2Seq +PVD	47.5%	10.0%
PCCM-Pos (ours)	46.7%	21.2%
PCCM (ours)	**60.8%**	**33.3%**

Table 4: Evaluation on the 240 sessions generated from MD.

4.4 Automatic Evaluation

We also presented results of automatic evaluation for the profile detector and position detector.

Profile Detection

The profile detector is evaluated from two aspects: whether a profile should be used or not ($\boldsymbol{P}(z=1|X)$), and whether a profile key is correctly chosen. Note that the prediction of profile key selection is cascaded on that of $\boldsymbol{P}(z=1|X)$.

Dataset (# samples)	Binary profile	Key selection
PB (3000)	85.1%	74.8%
MD (600)	82.0%	70.5%

Table 5: Classification accuracy of the profile detector.

The classifiers are trained on Weibo social data. Results in Table 5 show that the profile detector obtains fairly good accuracy. But the classifiers have a noticeable drop when tested on the manual dataset. This indicates that real human conversations are different from Weibo social interaction data.

Profile Key	Acc	Profile Key	Acc
Name	35.0%	Gender	96.0%
Age	98.5%	Weight	85.5%
location	99.0%	Constellation	100.0%

Table 6: Accuracy for predicting the start decoding position.

Position Detection

As mentioned previously, the position detector plays a key role in improving the naturalness, logic, and correctness of responses (see PCCM vs. PCCM-pos in Table 2), and the consistency and variety of conversational sessions (see Table 4). Thus, it is necessary to evaluate the performance of this module separately.

We randomly sampled 200 post-response pairs from PR for each key (1200 pairs in total), and then manually annotated the optimal position from which decoding should start. The results are shown in Table 6. The position for most keys can be estimated accurately while for *name* the prediction is bad. This is because the value of the key rarely occurs in our corpus, and the embeddings of such values are not fully trained. Nevertheless, the results are better than a random word picking strategy (see PCCM vs. PCCM-Pos in Table 2).

4.5 Extensibility

The effectiveness of our model is verified on six profile keys, but manual labors are required. We will show the extensibility of the model by evaluating it on four additional keys: *hobby, idol, speciality, and employer*.

Dataset	Method	Nat.	Logic	Cor.
4 keys	Seq2Seq	75.6%	39.9%	17.9%
	Seq2Seq+PV	87.2%	46.3%	35.4%
	Seq2Seq+PVD	87.0%	47.5%	35.3%
	PCCM-Pos	87.8%	47.6%	36.2%
	PCCM	88.8%	50.9%	39.6%

Table 7: Extensibility evaluation on 4 new keys.

Firstly, for the 4 keys, we extracted 16,332 post-response pairs from WD with 79 hand-crafted patterns and each pair is noisily mapped to one of the keys with these patterns. These new pairs, along with the old pairs on the six pairs, are used to retrain the model. Secondly, we constructed a test dataset consisting of 400 posts, 50 positive and 50 negative human-written posts for each key. Responses from our model and Seq2Seq are obtained and then evaluated. The manual labor exists only in hand-crafting the 79 patterns.

Results show that our model has a relative 10% drop on the new keys with respect to *logic and correctness*, and remains unchanged in *naturalness*. Nevertheless, our model is still much better than the Seq2Seq model. The baseline has no drop in *naturalness and logic* because this model does not deal with any pre-specified profile.

5 Conclusion and Future Work

We present a model that can generate responses that are coherent to a pre-specified, explicit personality or profile. Instead of learning implicit personality from speaker-tagged dialogue data, our work allows system developers to control chatbots' profile explicitly using generic dialogue data. Extensive results show that our model is effective to deliver more coherent and diversified conversations.

Our work moves toward endowing a chatbot with controllable personality. As a preliminary attempt, we only experimented with a few attributes of personality. Obviously, there are much more to be explored in this direction such as speaking styles, linguistic cues for extrovert or introvert, and many other subtle traits. More interestingly, this task raises more challenging problems such as semantic reasoning: if you ask a ten-year-old boy chatbot with "are you married?" or "do you play women football?", it requires to make reasoning with commonsense knowledge.

Acknowledgments

This work was partly supported by the National Science Foundation of China under grant No.61272227/61332007 and the National Basic Research Program (973 Program) under grant No. 2013CB329403.

References

[Al-Rfou *et al.*, 2016] Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372*, 2016.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Gosling *et al.*, 2003] Samuel D Gosling, Peter J Rentfrow, and William B Swann. A very brief measure of the big-five personality domains. *Journal of Research in personality*, 2003.

[Gu *et al.*, 2016] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, 2016.

[Gulcehre *et al.*, 2016] Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. In *ACL*, 2016.

[Güzeldere and Franchi, 1995] Güven Güzeldere and Stefano Franchi. Dialogues with colorful "personalities" of early ai. *Stanford Humanities Review*, 4(2):161–169, 1995.

[Jiwei Li, 2016] Chris Brockett Jianfeng Gao Bill Dolan Jiwei Li, Michel Galley. A diversity-promoting objective function for neural conversation models. In *NAACL*, 2016.

[Kottur *et al.*, 2017] Satwik Kottur, Xiaoyu Wang, and Vitor R Carvalho. Exploring personalized neural conversational models. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3728–3734. AAAI Press, 2017.

[Li *et al.*, 2016] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *ACL*, 2016.

[Mairesse and Walker, 2006] François Mairesse and Marilyn Walker. Automatic recognition of personality in conversation. In *NAACL*, 2006.

[Mairesse and Walker, 2007] François Mairesse and Marilyn Walker. Personage: Personality generation for dialogue. In *ACL*, 2007.

[Mairesse *et al.*, 2007] François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *JMLR*, 2007.

[Mou *et al.*, 2016] Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *ACL*, 2016.

[Norman, 1963] Warren T Norman. Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. *The Journal of Abnormal and Social Psychology*, 1963.

[Ritter *et al.*, 2011] Alan Ritter, Colin Cherry, and William B Dolan. Data-driven response generation in social media. In *EMNLP*, 2011.

[Serban *et al.*, 2016] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 2016.

[Shang *et al.*, 2015] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *ACL*, 2015.

[Shum *et al.*, 2018] Heung-Yeung Shum, Xiaodong He, and Di Li. From eliza to xiaoice: Challenges and opportunities with social chatbots. *arXiv preprint arXiv:1801.01957*, 2018.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[Turing, 1950] Alan M Turing. Computing machinery and intelligence. *Mind*, 1950.

[Vinyals and Le, 2015] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[Walker *et al.*, 1997] Marilyn A. Walker, Janet E. Cahn, and Stephen J. Whittaker. Improving linguistic style: social and affective bases for agent personality. In *In: Mller, J. (Ed.), Proceedings of Autonomous Agents*, 1997.

[Walker *et al.*, 2012] Marilyn A Walker, Grace I Lin, and Jennifer Sawyer. An annotated corpus of film dialogue for learning and characterizing character style. In *LREC*, 2012.

[Yin *et al.*, 2015] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. Neural generative question answering. In *IJCAI*, 2015.

[Young *et al.*, 2013] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *IEEE*, 2013.

[Yu *et al.*, 2016] Zhou Yu, Ziyu Xu, Alan W Black, and Alex I Rudnicky. Strategy and policy learning for non-task-oriented conversational systems. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2016.