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

Qiao Qian[1], Minlie Huang[1], Haizhou Zhao[2], Jingfang Xu[2], Xiaoyan Zhu[1]

[1]State Key Laboratory of Intelligent Technology and Systems

Tsinghua National Laboratory for Information Science and Technology

Dept. of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China

[2]Sogou, Inc. , Beijing, 100084, PR China

`qianqiaodecember29@126.com, aihuang@tsinghua.edu.cn,`
`zhaohaizhou@sogou-inc.com, xujingfang@sogou-inc.com,`
`zxy-dcs@tsinghua.edu.cn`

Abstract

Endowing a chatbot with personality or an identity is quite challenging but critical to deliver more realistic and natural conversations. In this paper, we address the issue of generating responses that are coherent to a pre-specified agent profile. We design a model consisting of three modules: a profile detector to decide whether a post should be responded using the profile and which key should be addressed, a bidirectional decoder to generate responses forward and backward starting from a selected profile value, and a position detector that predicts a word position from which decoding should start given a selected profile value. We show that general conversation data from social media can be used to generate profile-coherent responses. 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 conversation generation models 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 can be viewed as a composite of the identity (the background and profile) that

a chatbot is endowed with, linguistic style that an agent exhibits during interactions (Walker et al., 1997), and many more explicit and implicit cues that may portray character. Though personality is a more abstract and broad concept, we use *personality, profile, and identity* interchangeably in this paper.

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.

Profile key	Profile value
Name	汪仔(Wang Zai)
Age	三岁(3)
Gender	男孩(Boy)
Hobbies	动漫(Cartoon)
Speciality	钢琴(Piano)

Table 2: An exemplar agent profile.

Endowing a chatbot with personality is well motivated by the simple example as shown in Table 1. We can clearly see that a general sequence-to-sequence (Seq2Seq) model cannot exhibit coherent personality/identity while our model is more **coherent to a given identity**. The motivation is also verified by (Yu et al., 2016) which reports that users ask for much personal information of a chatbot in human-machine interaction. It is

evident that personal information of a chatbot is much attended by users during conversation, particularly at the early stage of interaction.

The recent work dealing with personality in large-scale conversation generation can be seen in (Li et al., 2016) where speaker-specific conversation style is learned by user embedding. Another work which models user personalization can be seen in (Al-Rfou et al., 2016), with a similar technique of user embedding. Both studies require dialogue data from each user to model her/his personality.

The major departure to the previous works lies in: **First**, we address the problem of endowing a chatbot with a given identity. Such a task requires chatbots to *generate not only consistent responses, but also responses that are coherent to its prespecified identity/personality*. **Second**, the previous works on personality modeling require conversation data from each user, however, it's impractical here since dialogue data from the chatbot are unavailable before the release of the chatbot. Instead of just learning personality from dialogue data, our work can assign a desired identity to a chatbot by making use of general conversation data from social media. Our contributions lie in two folds:

- We investigate the problem of endowing a chatbot with a given identity and enabling a chatbot to generate responses that are coherent to its given identity. Instead of learning personality from dialogue data, our work can assign a desired identity to a chatbot.

- To address the problem, we propose a model consisting of a profile detector, a position detector, and a bidirectional decoder. Post-level and session-level evaluation shows that when giving an agent profile, our model can generate more coherent responses with more language variety.

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; Wen et al., 2016; Bordes and Weston, 2016) 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; Vinyals and Le, 2015; Sordoni et al., 2015; Shang et al., 2015; Serban et al., 2015, 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 (Xing et al., 2017; 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 formally define in statistical 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 first attempt to model persona can be seen in (Li et al., 2016) where the authors proposed to learn speaker-specific conversation style by user embedding. Our work differs from this work significantly: our task is to endow the chatbot with a fixed personality while (Li et al., 2016) learns personalized conversational styles. In other words, our task requires to generate not only consistent responses, but also responses that are coherent to the chatbot's prespecified identity. Further, (Li et al., 2016) requires many dialogue data from each user while our work has no such requirement.

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 $\boldsymbol{x} = x_1 x_2 \cdots x_n$, and an agent 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 $\boldsymbol{y} = y_1 y_2 \cdots y_m$ that is coherent to the agent profile. The generation process can be briefly stated as below:

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

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

Note that the post/response pairs $< \boldsymbol{x}, \boldsymbol{y} >$ are collected from social media, and the agent profile value may not occur in the response $\boldsymbol{y}$. This leads to the discrepancy between training and test, which will be addressed in the *Position Detector* section.
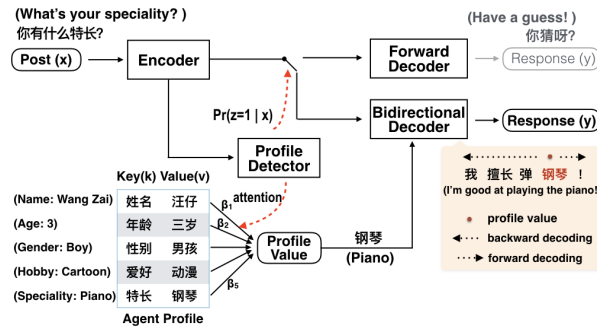
3.2 Overview

Figure 1: The overall process.

Our model works as follows (see Figure 1): given a post, the profile detector will predict whether the agent 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 better train the bidirectional decoder (see Figure 2), the position detector addresses the discrepancy issue between training and test, by predicting a word position from which decoding should start given the selected profile value. Note that the position detector will not participate in generation during test.

3.3 Encoder

The encoder aims to encode a post to a vector representation. Given a post $\boldsymbol{x} = x_1 x_2 \cdots x_n$, the hidden states of the post $\boldsymbol{h} = (h_1, h_2, \cdots, h_n)$ are obtained by a gated recurrent unit (GRU) (Chung et al., 2014), as follows:

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

where x_t is the embedding of the t-th word.

3.4 Profile Detector

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

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

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

where W_p is the parameter of the classifier and $\widetilde{h} = \sum_j h_j$, 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:

$$
\begin{aligned}
\beta_i &= \mathbf{MLP}([\widetilde{h}, k_i, v_i]) \\
&= f(W \cdot [\widetilde{h}; k_i; v_i])
\end{aligned} \quad (4)
$$

where W is the weight and k_i/v_i is the embedding of a profile key/value respectively, they are all parameters of our model. $\widetilde{h} = \sum_j h_j$ is the representation of the post. f is a nonlinear activation

function, in this equation f is a $softmax$ function over all β_i. The above equation can be viewed as a multi-class classifier that produces 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:

$$P^{bi}(\boldsymbol{y}|\boldsymbol{x},\{<k_i,v_i>\}) = P^{bi}(\boldsymbol{y}|\boldsymbol{x},\widetilde{v}) \quad (5)$$

3.5 The 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 forward decoder, but with a key difference that a position detector is employed to predict a start decoding position.

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

$$P^{bi}(\boldsymbol{y}|\boldsymbol{x},\widetilde{v}) = P^b(\boldsymbol{y^b}|\boldsymbol{x},\widetilde{v}) * P^f(\boldsymbol{y^f}|\boldsymbol{y^b},\boldsymbol{x},\widetilde{v}))$$

$$P^b(\boldsymbol{y^b}|\boldsymbol{x},\widetilde{v}) = \prod_{j=t-1}^{1} P^b(y_j^b|y_{>j}^b,\boldsymbol{x},\widetilde{v})$$

$$P^f(\boldsymbol{y^f}|\boldsymbol{y^b},\boldsymbol{x},\widetilde{v}) = \prod_{j=t+1}^{m} P^f(y_j^f|y_{<j}^f,\boldsymbol{y^b},\boldsymbol{x},\widetilde{v})$$

(6)

In order to encode more contexts in the forward decoder, the first half of generated response ($\boldsymbol{y^b}$), 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

$$P^b(y_j^b|y_{>j}^b,\boldsymbol{x},\widetilde{v}) \propto \mathbf{MLP}([s_j^b;y_{j+1}^b;c_j^b])$$

$$P^f(y_j^f|y_{<j}^f,\boldsymbol{y^b},\boldsymbol{x},\widetilde{v}) \propto \mathbf{MLP}([s_j^f;y_{j-1}^f;c_j^f])$$

(7)

where $s_j^{(*)}$ is the state of the corresponding decoder, $c_j^{(*)}$ is the context vector, and $* \in \{b,f\}$

---

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

where b indicates the backward decoder and f the forward. The vectors are updated as follows:

$$s_j^{(*)} = \mathbf{GRU}(s_{j+l}^{(*)}, [y_{j+l}^{(*)}; c_j^{(*)}])$$

$$c_j^{(*)} = \sum_{t=1}^{n} \alpha_{j,t}^{(*)} h_t$$

(8)

where $\alpha_{j,t}^{(*)} \propto \mathbf{MLP}([s_{j+l}^{(*)}, h_t])$ can be viewed as the similarity between decoder state $s_{j+l}^{(*)}$ and encoder hidden state 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.
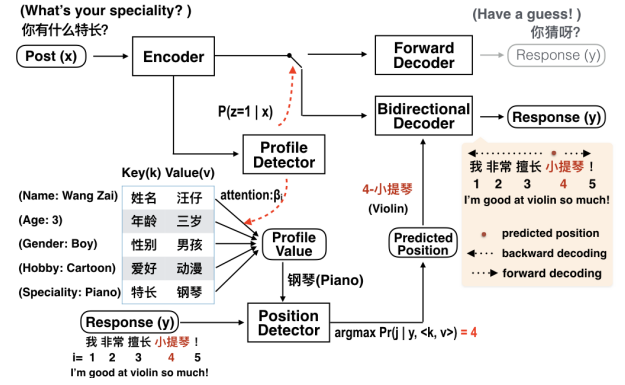
3.6 Position Detector

Figure 2: The training process of the model. Given a pair $< \boldsymbol{x}, \boldsymbol{y} >$, the position detector will predict a position 小提琴-4(violin) 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, which is only used during training. 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 $(\boldsymbol{x}, \boldsymbol{y}, <k,v>)$, the value ($v$) may not occur in $\boldsymbol{y}$ 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 given.

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).

The position detector is designed to provide a start decoding position to the decoder during training. For instance, given a post $\boldsymbol{x}$ ="你-1 有-2 什么-3 特长-4 ? -5 (what's your speciality?)[2]" and a response $\boldsymbol{y}$ ="我-1 非常-2 擅长-3 小提琴-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-4 (violin)" in the response can be replaced by the profile value "钢琴(piano)" to ensure grammaticality. The predicted position "小提琴-4 (violin)" is then passed to the decoder (see Eq. 6) to signal the start decoding position.

In order to find an appropriate position at which the profile value can be replaced, we need to estimate the probability: $\boldsymbol{P}(j|y_1y_2\cdots y_m, < k, v >)), 1 \le j \le m$ 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 maximal similarity.

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

where $cos(y_j, v)$ 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:

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

---

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

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 $\boldsymbol{y}$ 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 $\boldsymbol{P}(z|\boldsymbol{x})$ (see Eq. 3): $\boldsymbol{P}(z = 1|\boldsymbol{x}) = 1$ for profile-related pairs and $\boldsymbol{P}(z = 1|\boldsymbol{x}) = 0$ 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., $\mathcal{L} = \mathcal{L}_1 + \alpha\mathcal{L}_2$):

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

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. $< \boldsymbol{x}, \boldsymbol{y}, z >$ is obtained by manual annotation while $(\boldsymbol{x}, \boldsymbol{y}, \widehat{k})$ 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 prepare several datasets:

Weibo Dataset (WD) - D: We collect $9,697,651$ post-response pairs from Weibo. The dataset is used for training $\boldsymbol{P}^{fr}(\boldsymbol{y}|\boldsymbol{x})$ and $\boldsymbol{P}^{bi}(\boldsymbol{y}|\boldsymbol{x}, \widetilde{v})$ at the early stage and 7,000 pairs are used for validation to make early stop.

Profile Binary Subset (PB - $D^{(pb)} \in D$): We

extract $76,930$ pairs from WD for 6 profile keys ({*name, gender, age, city, 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 ($\boldsymbol{P}(z|\boldsymbol{x})$) (see $D^{(pb)}$ in Eq. 11). 3,000 pairs are used for test and the remainder for training. The statistics of the dataset is shown in the supplementary file.

Profile Related Subset (PR - $D^{(pr)} \in D^{(pb)}$**):** This dataset 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 old is your sister?*) should not. This dataset is used to test the performance on real conversation data rather than social media data.

All datasets are available upon request. **Implementation details of the model are shown in the supplementary.**

4.2 Human Evaluation

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

We name our model *Identity-Coherent Conversation Machine (ICCM)* and compare 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 ($\boldsymbol{P}(z|\boldsymbol{x}) > 0.5$), 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 de-

coder which starts decoding forwardly from the value of the selected key.
ICCM-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).

4.2.1 Post-level Evaluation

To conduct post-level evaluation, we use 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.

Results in Table 4 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 (ICCM vs. ICCM-Pos). Exemplar responses generated by these models are shown in Table 3 which also demonstrate the effectiveness of our model.

4.2.2 Session-level Evaluation

In order to compare these models in real conversation sessions, we randomly generate sessions

	Chinese	English (Translated)
	Post:你是不是美少女呀?	Post: Are you a beautiful girl ?
Seq2Seq	是呀。	Yes, I am.
Seq2Seq +PV	<u>男生</u>	<u>Boy</u>
Seq2Seq +PVD	<u>男生</u>。	<u>Boy</u>.
ICCM-Pos (ours)	你是<u>男生</u>?	Are you a <u>Boy</u>?
ICCM (ours)	我是<u>男生</u>!	I am a <u>Boy</u>!

Table 3: Sample responses generated by our model and baselines. The profile value in the agent profile is marked in <u>underline</u>.

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%
ICCM-Pos (ours)	87.4%	50.0%	41.8%
ICCM (ours)	**88.9%**	**55.9%**	**44.2%**

Table 4: Evaluation of responses to the 600 posts from MD.

based on MD. For each profile key, we randomly choose 3 *positive* posts[3] from MD, generate responses to the 3 posts for each model, and obtain 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 are contradictory responses with respect to the given profile. Score 1 indicates that all the three responses are consistent to the profile, and score 0 otherwise.

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 6 and we show some session examples in Table 5.

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 ICCM-Pos show that the position detector improves consistency and variety remarkably.

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

---

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

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 5: Samples of consistent conversations generated by our model. U/S indicates User/System.

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%
ICCM-Pos (ours)	46.7%	21.2%
ICCM (ours)	**60.8%**	**33.3%**

Table 6: Consistency and variety on the 240 sessions generated from MD.

4.3 Automatic Evaluation

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

4.3.1 Profile Detection

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

The classifiers are trained on Weibo social data. Results in Table 7 show that the profile detector

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

Table 7: Classification accuracy of the profile detector.

obtains fairly good accuracy. But the classifiers have a remarkable drop when test on the manual dataset (comparing two rows: MD(600) vs. PB(3000)). This indicates the difference between Weibo social data and real human conversations.

4.3.2 Position Detection

As mentioned previously, the position detector plays a key role in improving the naturalness, logic, and correctness of responses (see ICCM vs. ICCM-pos in Table 4), and the consistency and variety of conversational sessions (see Table 6). Thus, it is necessary to evaluate the performance of this module separately.

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

Table 8: Accuracy for predicting the start decoding position.

We randomly sample 200 post-response pairs from PR for each key (1200 pairs in total), and then manually annotate the optimal position from which decoding should start. The results are shown in Table 8. 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 (ICCM vs. ICCM-Pos).

4.4 Extensibility

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

Firstly, for the 4 keys, we extract $16,332$ 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 construct 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 rely on profile.

Dataset	Method	Nat.	Logic	Cor.
6 keys	Seq2Seq	71.4%	38.7%	22.3%
	ICCM	88.9%	55.9%	44.2%
4 keys	Seq2Seq	75.6%	39.9%	17.9%
	ICCM	88.8%	50.9%	39.6%

Table 9: Extensibility evaluation on 4 new keys.

5 Conclusion and Future Work

We present a model that can generate responses that are coherent to a pre-specified agent profile. Instead of learning personality from dialogue data, our work can assign a desired identity to a chatbot. Experiments show that our model is effective to generate more coherent and various conversations.

Our work is a very small step to endow a chatbot with its own personality, which is an important issue for a chatbot to pass the Turing Test. There are many future directions:

Conversation style: we have demonstrated that general conversation data can be used to generate profile-coherent responses. Can conversation style that is coherent to a chatbot's personality be modeled without stylistic dialogue data? Learning conversation styles such as talking like young girls or old adults, and introverts or extroverts, will be an interesting direction.

Semantic reasoning: Endowing a chatbot with personality/identity arises the issue of semantic reasoning. When the user asks "are you married?" or "do you play women basketball?" to a five-year-old boy agent, a coherent response requires common sense knowledge and reasoning, however, this is extremely challenging.

References

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

Antoine Bordes and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683* .

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

Samuel D Gosling, Peter J Rentfrow, and William B Swann. 2003. A very brief measure of the big-five personality domains. *Journal of Research in personality* 37(6):504–528.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational*.

Michel Galley Chris Brockett Jianfeng Gao Bill Dolan Jiwei Li. 2016. A diversity-promoting objective function for neural conversation models. In *The 2016 Conference of the North American Chapter of the Association*. pages 110–119.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

François Mairesse and Marilyn Walker. 2006. Automatic recognition of personality in conversation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, pages 85–88.

François Mairesse and Marilyn Walker. 2007. Personage: Personality generation for dialogue. In *Annual Meeting-Association For Computational Linguistics*. page 496.

François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. 2007. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research* pages 457–500.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *26th International Conference on Computational Linguistics, Proceedings of the Conference*. pages 3349–3358.

Warren T Norman. 1963. Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. *The Journal of Abnormal and Social Psychology* 66(6):574.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 583–593.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Hierarchical neural network generative models for movie dialogues. *CoRR* abs/1507.04808.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the Association for Computational Linguistics*. pages 1577–1586.

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 *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics*. 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.

Alan M Turing. 1950. Computing machinery and intelligence. *Mind* 59(236):433–460.

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

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

Marilyn A Walker, Grace I Lin, and Jennifer Sawyer. 2012. An annotated corpus of film dialogue for learning and characterizing character style. In *LREC*. pages 1373–1378.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562* .

Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2015. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. pages 2972–2978.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE* 101(5):1160–1179.

Zhou Yu, Ziyu Xu, Alan W Black, and Alex I Rudnicky. 2016. Strategy and policy learning for non-task-oriented conversational systems. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. page 404.

A   Statistics for Profile Binary Dataset

We extract $76,930$ pairs from WD for 6 profile keys ({*name, gender, age, city, weight, constellation*}) with about 200 hand-crafted patterns. And then they are annotated to *positive* or *negative*. A *positive* post asks for a profile key of chatbot and its response gives a meaningful reaction, such as *"Could you tell me your age"*, while a *negative* post is irrelevant to any profile key, such as *"Guess how old I am"*. The statistics of the dataset is shown in Table 10.

Profile Key	Positive	Negative
Name	6,966	3,442
Gender	7,665	8,259
Age	6,038	3,309
City	6,264	8,350
Weight	6,856	3,800
Constellation	8,404	7,577

Table 10: Statistics of the profile binary dataset. The data is mined from Weibo Dataset by hand-crafted patterns and manually labeled.

B   Implementation Details

In our experiments, the encoder and attentive 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 pairs) by 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.