

# One Chatbot Per Person: Creating Personalized Chatbots based on Implicit User Profiles

Zhengyi Ma<sup>2,3</sup>, Zhicheng Dou<sup>1,3</sup>, Yutao Zhu<sup>5</sup>, Hanxun Zhong<sup>2,3</sup>, and Ji-Rong Wen<sup>3,4</sup>

<sup>1</sup> Gaoling School of Artificial Intelligence, Renmin University of China

<sup>2</sup> School of Information, Renmin University of China

<sup>3</sup> Beijing Key Laboratory of Big Data Management and Analysis Methods

<sup>4</sup> Key Laboratory of Data Engineering and Knowledge Engineering, MOE

<sup>5</sup> Université de Montréal, Montréal, Québec, Canada

{zymaa,dou,hanxun\_zhong}@ruc.edu.cn,yutao.zhu@umontreal.ca,jirong.wen@gmail.com

## ABSTRACT

Personalized chatbots focus on endowing chatbots with a consistent personality to behave like real users, give more informative responses, and further act as personal assistants. Existing personalized approaches tried to incorporate several text descriptions as explicit user profiles. However, the acquisition of such explicit profiles is expensive and time-consuming, thus being impractical for large-scale real-world applications. Moreover, the restricted predefined profile neglects the language behavior of a real user and cannot be automatically updated together with the change of user interests. In this paper, we propose to learn implicit user profiles automatically from large-scale user dialogue history for building personalized chatbots. Specifically, leveraging the benefits of Transformer on language understanding, we train a personalized language model to construct a general user profile from the user's historical responses. To highlight the relevant historical responses to the input post, we further establish a key-value memory network of historical post-response pairs, and build a dynamic post-aware user profile. The dynamic profile mainly describes what and how the user has responded to similar posts in history. To explicitly utilize users' frequently used words, we design a personalized decoder to fuse two decoding strategies, including generating a word from the generic vocabulary and copying one word from the user's personalized vocabulary. Experiments on two real-world datasets show the significant improvement of our model compared with existing methods.

## CCS CONCEPTS

• **Computing methodologies** → **Discourse, dialogue and pragmatics; Natural language generation.**

## KEYWORDS

Personalized Chatbots; Response Generation; Implicit User Profile

## ACM Reference Format:

Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. 2021. One Chatbot Per Person: Creating Personalized Chatbots based on Implicit User Profiles. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, July 11–15, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3404835.3462828>

## 1 INTRODUCTION

Faced with extensive information available on the Internet, it is very appealing to have an intelligent assistant that can provide the most relevant information [16, 46, 47], collaborate with us on day-to-day problems [14, 45], or even act as our agent for some specific tasks [37]. Towards this ultimate goal, in the dialogue system area, building digital agents has attracted more and more attention and had some preliminary applications in our daily life [14, 24, 45]. In the future, some chit-chat between humans will inevitably be completed by digital agents. In this paper, we carry out a preliminary study toward this goal and focus on the problem of developing personalized chatbots. A personalized chatbot aims at leveraging personalized information (e.g., a predefined persona [13, 41]) to provide personalized responses when communicating with others. Such personalized information can help chatbots generate more consistent and informative replies. More importantly, if the personalized information can be well captured, the chatbots can perform similar behaviors like real users (e.g., serving as an agent of the user and give similar responses to others when the user is busy, as shown in Figure 1), thus having the potential to be an intelligent agent with that specific personality [37].

Many models have been proposed for improving the chatbot's capability to generate personalized responses. Early studies tried to integrate the user ID embeddings to a sequence-to-sequence (Seq2Seq) model for identifying the user and generating user-related responses [3, 4, 9]. Recently, some studies proposed to assign predefined personas to chatbots so as to generate more personalized responses [23, 31, 41]. They assumed that personality can be described in several sentences or attributes and help chatbots generate more persona-related responses. Different from existing studies, in this work, we propose letting the chatbot **learn the implicit user profile automatically from the user dialogue history and generate personalized responses based on the learned user profile**. In this way, the chatbot can be personalized by user's (e.g., user A) historical data, behave like this user, act as this user's agent, and chat with any other users (e.g., user B, C, etc.).

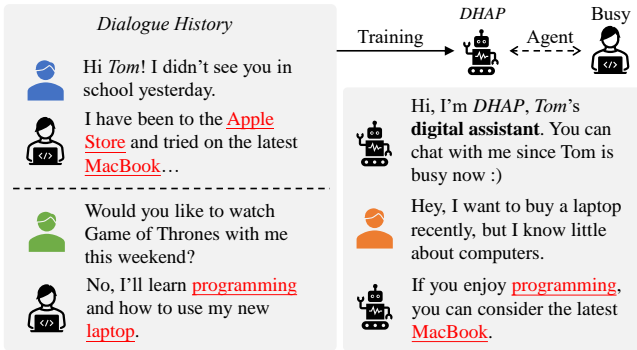
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8037-9/21/07...\$15.00

<https://doi.org/10.1145/3404835.3462828>



**Figure 1: An example of the personalized chatbot serving as an intelligent agent for user Tom when Tom is busy.**

Our idea is motivated by: (1) Contrary to explicit persona descriptions, user dialogue history is easier to be gathered on the user’s client devices. It is evident that obtaining explicit descriptions for massive users is impractical in real applications [4]: First, users may be lazy to set their profile before using the chatbot [6]. Second, manually collecting user profiles is costly and time-consuming. Furthermore, even if the user profile is collected, it cannot be updated with the change of user interests, thus may be ineffective over time. Finally, a fixed set of properties is not suitable for describing all users. (2) The user dialogue history contains massive personalized information, which is suitable for learning user profiles automatically. As shown in Figure 1, the dialogue history of a user includes their historical responses, and the corresponding posts issued by other users. Intuitively, users’ historical responses can often reflect their language style, background knowledge, frequently used words, and even their interests. For example, there may be many electronic device names that appeared in the historical responses of an electronic hobbyist (e.g., *MacBook* in Figure 1). Besides, the interaction content and style between a specific user and others can be captured from the historical post-response pairs. When faced with a new input post, the chatbot can look for the historical data, check how the user has responded to a similar post before, and apply similar interactions to generate a suitable response. In addition, user profiles learned from historical data can be gradually updated with more data being collected. In summary, the user dialogue history is easy to obtain and appropriate for building user profile.

To achieve our idea, we propose a model **DHAP** for personalized chatbots, which focuses on learning implicit user profile from user **D**ialogue **H**istory **A**utomatically and generating **P**ersonalized responses. In our model, a general user profile representation is **firstly** constructed from the user’s historical responses to capture the general information, including user interest, background knowledge, and speaking style. This is implemented by a personalized language model based on Transformer. **Then**, we design a personalized post encoder to construct the personalized post representation. The general user profile is utilized in the post encoder to better capture the semantic information of the input post. **Next**, we build a key-value memory network to store the user’s historical post-response pairs. Based on this history memory, the dynamic post-aware user profile is built by highlighting the historical responses relevant to the

current input post. **Finally**, we design a personalized decoder to fuse the learned user profile into the response generation process. The personalized decoder can switch between generating a word from a generic vocabulary and copying a word from the user’s personalized vocabulary. Experimental results on two large-scale datasets show that our proposed DHAP significantly outperforms existing response generation models in various evaluation metrics.

Our contributions are three-fold: (1) We learn the implicit user profile from user’s dialogue history automatically for generating personalized responses. By this means, our method can be applied without additional annotations on user profiles, and create personalized chatbots as user’s digital agents. (2) We build two kinds of user profiles from the dialogue history, including the general user profile reflecting the user’s general information and the dynamic post-aware user profile to apply similar interactions in the historical data for the current input. (3) We design a personalized decoder to coordinate two personalized decoding strategies, including generating a word from the generic vocabulary and copying a word from the personalized vocabulary to leverage the user’s word preference.

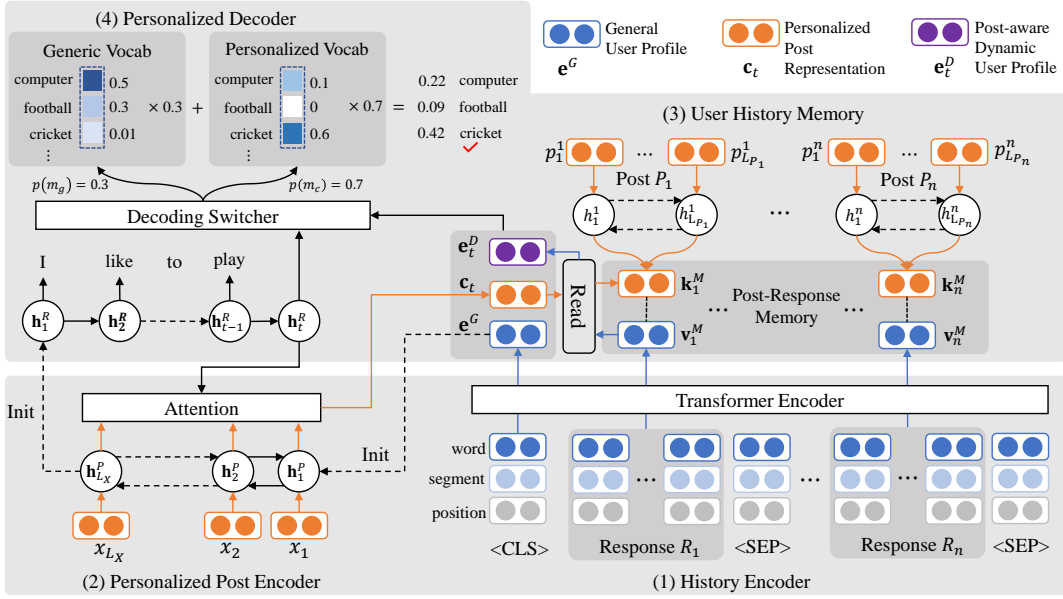
## 2 RELATED WORK

*Open-domain Chatbots and Response Generation.* Open-domain chatbots have attracted more and more attention, due to their broad application in real applications, such as Microsoft XiaoIce [45]. Typical methods can be categorized into two groups: retrieval-based and generation-based. Retrieval-based methods aim to select a suitable response from a large repository [38, 49, 50], while generation-based methods aim at generating a response from scratch [26, 28, 32, 48]. In this study, we focus on the response generation problem.

Some early studies treat the response generation task as a statistical machine translation problem because of its end-to-end and data-driven features [25, 34]. More recently, with the progress of deep learning, Seq2Seq methods have been applied to response generation and achieve great performance [8, 26, 27, 34]. Many Seq2Seq-based extensions have been applied to tackle the “safe response” problem [8]; to incorporate external knowledge [44]; to generate responses with emotions or personas [9, 23, 43]; and to model the hierarchical structure of the dialogue context [26, 27].

*Personalized Chatbots.* Endowing chatbots with a coherent personality is a challenging but necessary step to achieve the ultimate goal of building an intelligent assistant. With personality, a chatbot can generate more informative and user-specific responses [9, 41], and has the potential to perform similar behaviors as real humans.

Traditional personalized chatbots focus on modeling the user’s psychological behavior such as the “Big Five” of speakers [17]. In recent years, deep-learning based methods were proposed to learn the persona information directly from large-scale dialogue datasets via end-to-end neural networks [4, 9, 23, 41]. Some researchers first tried to input the user ID embeddings into the decoder of a Seq2Seq model to generate more personalized responses [1, 3, 4, 9]. Despite users can be identified by their IDs, the personalized performance is limited because no user-related information is used in the model. Therefore, another group of researchers proposed assigning explicit profiles (personas) for chatbots to generate personalized responses. For example, Zhang et al. [41] published the PERSONA-CHAT dataset, in which each user is assigned with several persona



**Figure 2: The overall structure of the proposed model DHAP, which consists of (1) a history encoder, (2) a personalized post encoder, (3) a user history memory, and (4) a personalized decoder.**

description sentences and the conversations between users are collected. With explicit profile descriptions, chatbots can learn to generate more consistent and informative dialogues. On this dataset, many methods have achieved encouraging performance, such as variational autoencoders [30], pre-trained language models [29, 36], and multi-task modeling [29, 31, 35]. These explicit persona-based methods enjoy the high quality of predefined personas, but the acquisition of these persona data is expensive and even impossible when applied to real-world systems [4].

In this work, we propose using the *implicit user profile* to drive the response generation. Such user profile is implicitly learned with neural models from user dialogue history rather than predefined explicitly. Due to the fact that the user dialogue history contains abundant user-specific information and is easily accessible in real-world applications, our method is more applicable in practice.

### 3 METHODOLOGY

In this section, we first provide an overview of our proposed model DHAP. The details of each component are provided later, and the model training is introduced finally.

#### 3.1 The Overview of DHAP

Suppose that for a user  $u$ , we have their dialogue history  $H$  including a series of responses issued by  $u$  and the corresponding posts:  $H = ((P_1, R_1), \dots, (P_n, R_n))$ , where  $n$  is the number of historical post-response pairs. Note that the posts  $(P_1, \dots, P_n)$  here can be issued by different users, but the responses  $(R_1, \dots, R_n)$  are all issued by the same user  $u$ . We call them *historical posts* and *historical responses* respectively in the following sections. Under the single-turn setting, given an input post  $X = (x_1, \dots, x_{L_X})$  and the user dialogue history  $H$ , with sequence-to-sequence modeling, our task

is to generate a personalized response  $Y = (y_1, \dots, y_{L_Y})$  as:

$$p(Y|X, u) = p(Y|X, H) = \prod_{t=1}^{L_Y} p(y_t | y_{<t}, X, H), \quad (1)$$

where  $y_t$  denotes the word generated at the  $t$ -th step, and  $y_{<t}$  denotes the previous generated words  $(y_1, \dots, y_{t-1})$ . It is worth noting that the dialogue history  $H$  here includes dialogues between the user  $u$  and several other users, thus the history is not a multi-turn dialogue between two fixed interlocutors.

To compute the probability  $p(Y|X, H)$ , we design a model called DHAP, which stands for learning user profiles from Dialogue History Automatically for Personalized chatbots. The structure of DHAP is shown in Figure 2. We briefly introduce the key components of DHAP as follows. In general, DHAP considers personalized information in both the encoder and the decoder side.

**3.1.1 Encoder.** DHAP has two different encoders, which encode the input post and user’s historical responses, respectively:

**Part (1): History encoder and general user profile.** Since abundant personalized information (e.g., background knowledge and speaking style) is often hidden in user’s historical responses, DHAP firstly establishes a Transformer-based personalized language model to encode historical responses  $(R_1, \dots, R_n)$ , then summarizes the general user profile  $e^G$  based on the historical responses. As  $e^G$  does not depend on the input  $X$ , we call it *general user profile*. “General” does not mean it is general among all users or cannot be updated together with time. In contrast, every user has their own profile, which can be updated once they issue a new response.

**Part (2): Personalized post encoder.** DHAP also has an encoder for the input post, which is implemented by a bidirectional

GRU (BiGRU). To make the post encoder aware of the user’s personalized information, we use the general user profile  $\mathbf{e}^G$  for initialization. Consequently, the post  $X = (x_1, \dots, x_{L_X})$  is represented by hidden state sequence  $\mathbf{h}^P = (\mathbf{h}_1^P, \dots, \mathbf{h}_{L_X}^P)$ . These representations will be dynamically aggregated as a personalized post representation  $\mathbf{c}_t$  by an attention mechanism at the decoding step  $t$ .

**3.1.2 Decoder.** In the decoder side, DHAP incorporates the personalized information in two perspectives:

**Part (3): User history memory and dynamic post-aware user profile.** In the general user profile, all historical responses are considered in a general view. However, for a specific input post  $X$ , the historical responses may play different roles. Intuitively, the historical responses relevant to  $X$  are more valuable than the irrelevant ones. To highlight the historical responses relevant to the input post, DHAP builds a key-value memory of historical post-response pairs  $((P_1, R_1), \dots, (P_n, R_n))$ . Given  $\mathbf{c}_t$  as a personalized representation of  $X$ , DHAP builds a dynamic post-aware user profile  $\mathbf{e}_t^D$  by selecting and aggregating the historical responses from the user history memory. As  $\mathbf{e}_t^D$  is dynamic with different input posts  $X$  and summarizes related user history information, we call it *dynamic post-aware user profile*.

**Part (4): Personalized decoder.** Finally, the personalized post representation  $\mathbf{c}_t$ , general user profile  $\mathbf{e}^G$ , and dynamic post-aware user profile  $\mathbf{e}_t^D$  are fused together to decode the response sequentially. Inspired by CopyNet [7], DHAP can switch between generating a word from a generic vocabulary  $(p(y_t|m_g))$  and copying a word from a user’s personalized vocabulary  $(p(y_t|m_c))$  as:

$$p(y_t|y_{<t}, X, H) = p(m_g)p(y_t|m_g) + p(m_c)p(y_t|m_c), \quad (2)$$

where  $p(m_g)$  and  $p(m_c)$  are computed by our designed decoding switcher.  $p(y_t|m_g)$  and  $p(y_t|m_c)$  are calculated based on same inputs (e.g.,  $\mathbf{c}_t$ ,  $\mathbf{e}^G$  and  $\mathbf{e}_t^D$ ) with different functions.

In the remaining part of this section, we will introduce these four components of DHAP in detail.

## 3.2 History Encoder and General User Profile

Based on our observation, there is a large amount of personalized information in the user’s historical responses. For example, a fan of cricket may talk a lot about cricket topics with others. Furthermore, different users can hold different speaking styles, such as enjoying speaking slang. Therefore, our first idea is to devise a model for learning such personalized language information in a general view.

Inspired by the strong ability of Transformer [33] to aggregate context and model sequences, we use a Transformer encoder to learn contextual representations of historical responses. In particular, we first add special tokens and concatenates all  $n$  historical responses as  $R^w = [\langle \text{CLS} \rangle; R_1; \langle \text{SEP} \rangle; \dots; R_n; \langle \text{SEP} \rangle]$ , where each response  $R_i = (r_1^i, \dots, r_{L_{R_i}}^i)$  contains  $L_{R_i}$  words, and  $[\cdot]$  is the concatenation operation. A “ $\langle \text{SEP} \rangle$ ” token is added at the tail of each response for segmentation, while a “ $\langle \text{CLS} \rangle$ ” token is added at the sequence head for summary. Then we map all words and special tokens into embeddings, and learn their contextual representations:

$$[\mathbf{e}^G; \mathbf{E}^R] = \text{Transformer}_N([\mathbf{e}_{\text{CLS}}; \mathbf{R}_1; \mathbf{e}_{\text{SEP}}; \dots; \mathbf{R}_n; \mathbf{e}_{\text{SEP}}]), \quad (3)$$

$$\mathbf{R}_i = [\mathbf{e}_1^{r_i, \text{fuse}}; \dots; \mathbf{e}_{L_{R_i}}^{r_i, \text{fuse}}], \quad (4)$$

where  $\mathbf{e}_j^{r_i, \text{fuse}}$  is the sum of the word embedding, segment embedding, and position embedding of the  $j$ -th word. These three embeddings are used in a similar way like BERT [5]. The representation of “ $\langle \text{CLS} \rangle$ ” token ( $\mathbf{e}^G$ ) summarizes the information in the entire historical responses, thus we call it the general user profile.  $\mathbf{E}^R$  contains contextual representations of words in historical responses. For  $R_i$ , we denote its contextual representations as  $(\mathbf{e}_1^{r_i}, \dots, \mathbf{e}_{L_{R_i}}^{r_i})$ .  $\text{Transformer}_N(\cdot)$  is a  $N$ -layer bidirectional Transformer encoder identical to the original implementation described in [33].

With the history encoder, we obtain: (1) a general user profile  $\mathbf{e}^G$ , which summarizes all historical responses and contains personalized information of the user; and (2) the contextual representations of words in the historical responses  $\mathbf{E}^R$ . These representations will be used to build the dynamic post-aware user profile in Section 3.4.

## 3.3 Personalized Post Encoder

In real-world applications, the posts are often very short, even ambiguous. Thus, building accurate encoding of the input post is difficult, which further leads to poor quality of the generated responses [9, 28]. Fortunately, with personalized background knowledge, the chatbot can get more input information and is promising to better capture the semantic information of the post. Let us use an example to explain this: given a post “The new MAC is so beautiful”, different users may have different understandings. For a programmer, “MAC” may refer to the Apple’s laptop; but for a fashion girl, she may associate “MAC” with the lipstick. Therefore, the user’s history can help distinguish the word “MAC” and provide more background knowledge for the input post, so that the post representation can be significantly enhanced. We call this encoder “personalized post encoder” - the encoded representations of the same post can be different for various users, because these users may have different profiles and different understandings of the same post.

Specifically, DHAP employs a BiGRU to encode the input post. Here we choose RNN-based architectures because they are better at capturing local correlations and encoding positional information than Transformers for short texts [19, 39]. To facilitate the encoder with personalized information, DHAP uses the general user profile  $\mathbf{e}^G$  to initialize hidden states of BiGRU. Given the post  $X = (x_1, \dots, x_{L_X})$ , its representations  $(\mathbf{h}_1^P, \dots, \mathbf{h}_{L_X}^P)$  are built as:

$$\mathbf{h}_i^P = \text{BiGRU}(\mathbf{h}_{i-1}^P, \mathbf{x}_i), \quad \mathbf{h}_0^P = \text{ReLU}(\text{MLP}(\mathbf{e}^G)), \quad (5)$$

where  $\mathbf{x}_i$  is the embedding obtained by the embedding table.

These hidden states draw personalized information from the general user profile, and in the decoding phase, they are aggregated by the decoding hidden state  $\mathbf{h}_t^R$  through an attention mechanism:

$$\mathbf{c}_t = \sum_{i=1}^{L_X} \alpha_{t,i} \mathbf{h}_i^P, \quad (6)$$

$$\alpha_{t,i} = \frac{\exp(s_{t,i}^{rp})}{\sum_{j=1}^{L_X} \exp(s_{t,j}^{rp})}, \quad s_{t,i}^{rp} = \mathbf{v}_{rp}^\top \tanh(\text{MLP}([\mathbf{h}_t^R; \mathbf{h}_i^P])). \quad (7)$$

The detailed calculation and updating scheme of the decoding state  $\mathbf{h}_t^R$  at time step  $t$  will be described in Section 3.5.4. Based on the attention mechanism, the personalized post representation  $\mathbf{c}_t$  can

dynamically focus on some important words of the post according to the current decoding state. In the next section, DHAP will use  $\mathbf{c}_t$  to build the dynamic post-aware user profile.

### 3.4 User History Memory and Dynamic Post-aware User Profile

With the general user profile, DHAP can capture personalized information of a user in a general view. However, when faced with a new input post, the historical data may play different roles. For example, when a crazy fan of cricket meets a post about cricket, they will be talkative and post a lot of things. But when they face other daily topics, they may behave much more gently. Hence, it is valuable to dynamically select the information that is most relevant to the input, and the chatbot can behave differently as the input varies. Following this idea, we propose to dynamically aggregate historical responses that are highly relevant to the current input post, and leverage them as a reference to drive the response generation.

To achieve this, DHAP uses a key-value memory network [18] to store the user’s historical post-response pairs. Then the personalized post representation is used as the query to select highly related keys (historical posts) from the user history memory, and their corresponding values (historical responses) are aggregated as the dynamic post-aware user profile.

**3.4.1 User History Memory.** We firstly transform the historical post-response pairs into key-value pairs and build the memory.

As we discussed earlier, the historical posts are usually issued by different users. Thus the language style and topic of them may be various. Under this circumstance, it is more reasonable to treat them independently, so DHAP applies a BiGRU to represent each historical post, respectively. In our implementation, this BiGRU shares parameters with the personalized post encoder (introduced in Section 3.3). Consider the  $i$ -th historical post  $P_i$ , its representation is computed by a summing pooling over the word dimension as  $\mathbf{p}_i^c = \sum_{j=1}^{L_{P_i}} \mathbf{h}_j^i$ , where  $\mathbf{h}_j^i$  is the hidden state of the BiGRU for the  $j$ -th word in  $P_i$ . For all historical posts  $(P_1, \dots, P_n)$ , their representations are denoted as  $(\mathbf{p}_1^c, \dots, \mathbf{p}_n^c)$ . Similarly, the representation of historical responses is also computed by the same pooling strategy as  $\mathbf{r}_i^c = \sum_{j=1}^{L_{R_i}} \mathbf{e}_j^{r_i}$ , where  $\mathbf{e}_j^{r_i}$  is the contextual representation of the  $j$ -th word in  $R_i$ . Different from historical posts, the contextual word representations of historical responses are obtained by the history encoder in Equation (3). Thus, all historical responses are represented as  $(\mathbf{r}_1^c, \dots, \mathbf{r}_n^c)$ .

Finally, we build the user history memory by using the post representations as key and the corresponding response representations as value, *i.e.*,  $\{\mathbf{k}_1^M : \mathbf{v}_1^M, \dots, \mathbf{k}_n^M : \mathbf{v}_n^M\} = \{\mathbf{p}_1^c : \mathbf{r}_1^c, \dots, \mathbf{p}_n^c : \mathbf{r}_n^c\}$ .

**3.4.2 Dynamic Post-aware User Profile.** After building the user history memory, DHAP can select and aggregate the most relevant historical responses and build the dynamic post-aware user profile based on the input post. Specifically, the personalized representation of the input post obtained in Equation (6) is used as the query and attends to the memory keys to find the most relevant historical posts. The relevance is measured by the attention weights. Then the corresponding historical responses are summed up based on

the normalized weights to construct the dynamic profile  $\mathbf{e}_t^D$ :

$$\mathbf{e}_t^D = \sum_{i=1}^n \beta_{t,i} \mathbf{v}_i^M, \quad (8)$$

where  $\beta_{t,i}$  is the attention weight of the  $i$ -th historical response based on the personalized post representation  $\mathbf{c}_t$  in a similar way like Equation (7). Note that the dynamic post-aware user profile is computed at each decoding time step  $t$ . The most relevant information hidden in historical responses can thus be selected to help response generation.

### 3.5 Personalized Decoder

Finally, the generation probability of responses can be calculated by the personalized post representation  $\mathbf{c}_t$ , the general user profile  $\mathbf{e}^G$ , and the dynamic post-aware user profile  $\mathbf{e}_t^D$ . Inspired by CopyNet [7], in addition to leveraging the personalized information captured by the implicit user profile, we construct a personalized vocabulary so that the model is allowed to directly select personalized words that the user frequently used in history.

Specifically, the probability of the word  $y_t$  generated in the personalized decoder is computed as Equation (2), where  $p(m_g)$  is the probability of general decoding mode and  $p(m_c)$  is the probability of copy decoding mode. They are computed by our proposed decoding switcher.  $p(y_t|m_g)$  and  $p(y_t|m_c)$  are the probabilities of generating  $y_t$  under two modes, respectively.

It is worth noting that: (1) The switching probability  $p(m_g)$  and  $p(m_c)$  are both in  $[0, 1]$ . Thus, it is a “soft” decoding switcher. (2) The generic vocabulary also contains words in the personalized vocabulary. The generation probability of  $y_t$  is obtained by the sum of probabilities under two decoding modes. Therefore, DHAP is just biased to the personalized words rather than lost in it.

**3.5.1 Decoding switcher.** The decoding switcher determines the probability of the two decoding modes, *i.e.*, predicting a word from the generic vocabulary to maintain sentence fluency, or copying a word directly from the personalized vocabulary to make the response more informative and personalized. Specifically, DHAP computes the switching probability based on the matching degree between the decoder state and the concatenation of the personalized post representation, general user profile, and dynamic post-aware user profile, and further calculates two decoding mode probabilities.

$$[p(m_g), p(m_c)] = \text{Softmax}(\mathbf{d}_t), \quad (9)$$

$$\mathbf{d}_t = \text{MLP}([\mathbf{h}_t^R; \mathbf{c}_t; \mathbf{e}^G; \mathbf{e}_t^D]), \quad (10)$$

where  $\mathbf{h}_t^R$  is the decoding hidden state at step  $t$ , and  $\mathbf{d}_t \in \mathbb{R}^2$  is the matching degree vector to estimate the two mode probabilities. The softmax function guarantees that  $p(m_g) + p(m_c) = 1$ .

**3.5.2 Personalized general decoding.** While general decoding, the decoder should predict a word  $y_t$  from the generic vocabulary:

$$p(y_t|m_g) = \text{Softmax}(\text{MLP}([\mathbf{h}_t^R; \mathbf{c}_t; \mathbf{e}^G; \mathbf{e}_t^D])). \quad (11)$$

**3.5.3 Personalized copy decoding.** The personalized vocabulary of user  $u$  is composed of the words that appear in their historical responses. DHAP can directly select a word from this vocabulary

to generate a more personalized response. Inspired by copy mechanism [7], the probability of selecting a word  $y_t$  is computed as:

$$p(y_t|m_c) = \sum_{i:r_i=y_t} \gamma_{t,i}, \quad (12)$$

where  $\gamma_{t,i}$  is the attention weight calculated by the personalized post representation  $c_t$  attentively reading the representation of historical responses  $E^R$  with the same attention process in Equation (6).

**3.5.4 Decoder state updating.** DHAP applies a GRU as the decoder. The hidden state at decoding step  $t$  is calculated as:

$$\mathbf{h}_t^R = \text{GRU}(\mathbf{h}_{t-1}^R, [\mathbf{y}_{t-1}; \mathbf{c}_t; \mathbf{e}^G; \mathbf{e}_t^D]), \quad (13)$$

where  $\mathbf{y}_{t-1}$  is the embedding vector of the last generated word. The decoding states are initialized by the last hidden state of the personalized post encoder:

$$\mathbf{h}_0^R = \text{ReLU}(\text{MLP}(\mathbf{h}_{L_X}^P)). \quad (14)$$

### 3.6 Training and Optimization

Our goal is to maximize the generation probability of the target response given the input post and user’s dialogue history. A length penalty is applied as [9] to alleviate the generation of meaningless responses. As a result, the loss function of DHAP is defined as:

$$\mathcal{L} = - \sum_{t=1}^{L_Y} \log [p(y_t|y_{<t}, X, H)] - \eta L_Y, \quad (15)$$

where  $\eta$  is a hyper-parameter to control the associated length penalty weight.  $p(y_t|y_{<t}, X, H)$  is the generation probability of word  $y_t$  based on the given input post and user’s history, which is computed in Equation (2). All parameters are optimized by the loss function and the whole model is trained in an end-to-end manner.

## 4 EXPERIMENTS

### 4.1 Datasets

Although there are many public datasets for response generation, as far as we know, none of them contain user identification. To collect each user’s dialogue history and evaluate the effectiveness of our model, we use two datasets extracted from two open online chatting forums, *i.e.*, Weibo and Reddit. The two datasets contain massive dialogue utterances (*i.e.*, post-response pairs) and user identification information, thus we can sample the data by users. To guarantee enough personalized information, we retrieve users with more than ten utterances to maintain an effective dialogue history. Each utterance is used as the target response for generation, while its former responses and the corresponding posts are treated as the dialogue history. We divide the utterances by users into 8:1:1 as training, validation, and test set respectively in time order. Besides, given a user, we ensure that the time of its records in the validation set and test set are behind the records in the training set.

**Weibo dataset** is a subset of PChatbotW [22], which is collected from Weibo for the one-year period beginning from Sept. 10, 2018. On Weibo, a user can post short messages visible to the public, which will be referred to as posts. Other users can make comments on a published post, which will be referred to as response. For data cleaning, we remove hashtags, URLs, emoticons, and duplicate text as [22]. We also remove the utterances whose length is less than

five words or more than 100 words. We use comparable scales of samples with [4] to conduct our experiments. It comprises 300,000 users and 8,618,374 utterances, in total 31M words.

**Reddit dataset** is extracted from comment chains scraped from Reddit from Dec. 1, 2015 to Oct. 30, 2018 [42]. Since the Reddit discussions can be naturally expanded as tree-structured reply chains, we pair the parent node with all its child nodes respectively, and construct multiple post-response pairs. We treat the parent node and the child node as the post and response, respectively. As a result, a parent node can be a submission or a comment, while a child node only refers to a comment. For each submission, we use its title as the post text. We clean the raw data by removing instances containing word repetitions, offensive words, or multi-language sentences. It contains 315,340 users and 24,162,464 utterances, in total 55M words.

### 4.2 Baselines

We evaluate the performance of our approach by comparing it with four groups of highly related and strong baseline methods:

(1) *Non-personalized response generation models.* **Seq2SeqWA** [2] is a standard GRU-based Seq2Seq model with attention mechanism. **MMI** [8] is a Seq2SeqWA using Maximum Mutual Information as loss function to improve diversity.

(2) *Personalized models using user ID embeddings.* **Speaker** [9] is also based on Seq2SeqWA but using user ID embeddings as additional input to the decoder. **PersonaWAE** [4] is built on an augmented Wasserstein autoencoder. It utilizes user ID embeddings for building a personalization Gaussian mixture distribution, and fuses personalization in the decoder.

(3) *Personalized models using explicit user profiles.* Since no explicit user profiles are given in our datasets, we use the historical responses of users as their persona texts. **GPMN** [41] enhances the Seq2SeqWA with a memory module, which encodes each piece of persona description as an individual memory representation. It uses the input message as the query to aggregate and incorporate the memory representations for response generation. **PerCVAE** [30] uses the user profile descriptions as conditions and applies a conditional variational autoencoder to generate diverse responses.

(4) *Personalized models using implicit user profiles.* Since no existing methods consider mining user profiles from dialogue history implicitly, we adapt several state-of-the-art multi-turn response generation models to personalized response generation. We replace the dialogue context in the original models by the user’s historical post-response pairs. **VHRED-P** [27] extends the hierarchical recurrent encoder-decoder with a latent variable to model the complex dependencies among multiple utterances in the context. **ReCoSa-P** [40] applies a self-attention mechanism to measure the relevance between the response and each utterance in the context.

### 4.3 Evaluation Metrics

**Automatic Evaluation:** We consider several automatic metrics in different perspectives to jointly evaluate the generated responses. (1) We use BLEU-1, BLEU-2 [20], and ROUGE-L [12] to measure word overlaps between the generated response and ground truth. A higher value of these metrics indicates a higher word-level similarity between the generated response and the golden response.

**Table 1: Automatic evaluation results of all models. All models are categorized into four groups: (1) non-personalized; (2) using user ID; (3) using explicit user profile; and (4) using dialogue history. “†” denotes the result is significantly worse than our method DHAP in t-test with  $p < 0.05$  level. The best results are in bold and the second best results are underlined.**

Dataset	Model	Word Overlap			Diversity		Embedding Similarity			Personalization	
		BLEU-1	BLEU-2	ROUGE-L	Dist-1	Dist-2	Average	Extrema	Greedy	P-F1(%)	P-Cover
Weibo	(1) Seq2SeqWA	3.335 <sup>†</sup>	0.294 <sup>†</sup>	8.740 <sup>†</sup>	0.935 <sup>†</sup>	2.184 <sup>†</sup>	0.321 <sup>†</sup>	0.266 <sup>†</sup>	0.254 <sup>†</sup>	1.698 <sup>†</sup>	0.041 <sup>†</sup>
	(1) MMI	3.632 <sup>†</sup>	0.095 <sup>†</sup>	5.317 <sup>†</sup>	10.714 <sup>†</sup>	43.479 <sup>†</sup>	0.477 <sup>†</sup>	0.695 <sup>†</sup>	0.305	1.874 <sup>†</sup>	0.054 <sup>†</sup>
	(2) Speaker	4.997 <sup>†</sup>	0.113 <sup>†</sup>	7.993 <sup>†</sup>	6.035 <sup>†</sup>	19.017 <sup>†</sup>	0.492 <sup>†</sup>	0.712 <sup>†</sup>	0.311	2.119 <sup>†</sup>	0.082 <sup>†</sup>
	(2) PersonaWAE	3.503 <sup>†</sup>	0.155 <sup>†</sup>	11.305 <sup>†</sup>	2.493 <sup>†</sup>	19.716 <sup>†</sup>	<u>0.513</u>	<u>0.724</u>	0.307	5.108 <sup>†</sup>	<u>0.093</u>
	(3) GPMN	4.901 <sup>†</sup>	0.696 <sup>†</sup>	8.090 <sup>†</sup>	11.726 <sup>†</sup>	32.734 <sup>†</sup>	0.353 <sup>†</sup>	0.391 <sup>†</sup>	0.301 <sup>†</sup>	4.512 <sup>†</sup>	0.084 <sup>†</sup>
	(3) PerCVAE	5.115 <sup>†</sup>	0.299 <sup>†</sup>	7.952 <sup>†</sup>	<u>14.095</u>	<u>49.739</u>	0.469 <sup>†</sup>	0.659 <sup>†</sup>	0.299 <sup>†</sup>	3.817 <sup>†</sup>	0.086 <sup>†</sup>
	(4) VHRED-P	6.992 <sup>†</sup>	0.709 <sup>†</sup>	10.695 <sup>†</sup>	2.122 <sup>†</sup>	7.874 <sup>†</sup>	0.437 <sup>†</sup>	0.560 <sup>†</sup>	0.307	5.459 <sup>†</sup>	0.065 <sup>†</sup>
	(4) ReCoSa-P	<u>7.266</u>	<u>0.844</u>	<u>11.469</u>	1.271 <sup>†</sup>	4.442 <sup>†</sup>	0.419 <sup>†</sup>	0.510 <sup>†</sup>	<u>0.312</u>	<u>5.717</u>	0.061 <sup>†</sup>
	(4) DHAP (ours)	<b>9.324</b>	<b>0.894</b>	<b>14.122</b>	<b>15.175</b>	<b>58.806</b>	<b>0.523</b>	<b>0.747</b>	<b>0.313</b>	<b>7.013</b>	<b>0.144</b>
	Reddit	(1) Seq2SeqWA	1.819 <sup>†</sup>	0.023 <sup>†</sup>	4.069 <sup>†</sup>	5.203 <sup>†</sup>	19.485 <sup>†</sup>	0.545 <sup>†</sup>	0.554 <sup>†</sup>	0.472 <sup>†</sup>	0.516 <sup>†</sup>
(1) MMI		2.065 <sup>†</sup>	0.011 <sup>†</sup>	3.784 <sup>†</sup>	5.914 <sup>†</sup>	31.093 <sup>†</sup>	0.543 <sup>†</sup>	0.607 <sup>†</sup>	0.454 <sup>†</sup>	0.828 <sup>†</sup>	0.038 <sup>†</sup>
(2) Speaker		2.642 <sup>†</sup>	0.054 <sup>†</sup>	4.469 <sup>†</sup>	8.951 <sup>†</sup>	34.187 <sup>†</sup>	0.538 <sup>†</sup>	0.606 <sup>†</sup>	0.457 <sup>†</sup>	1.455 <sup>†</sup>	0.031 <sup>†</sup>
(2) PersonaWAE		2.637 <sup>†</sup>	0.113 <sup>†</sup>	8.199 <sup>†</sup>	1.758 <sup>†</sup>	25.915 <sup>†</sup>	0.629 <sup>†</sup>	<u>0.685</u>	0.442 <sup>†</sup>	3.392 <sup>†</sup>	0.032 <sup>†</sup>
(3) GPMN		2.686 <sup>†</sup>	0.376 <sup>†</sup>	4.776 <sup>†</sup>	<u>12.325</u>	35.762 <sup>†</sup>	0.406 <sup>†</sup>	0.331 <sup>†</sup>	0.358 <sup>†</sup>	3.026 <sup>†</sup>	0.037 <sup>†</sup>
(3) PerCVAE		5.933 <sup>†</sup>	0.576 <sup>†</sup>	8.112 <sup>†</sup>	9.631 <sup>†</sup>	<u>40.213</u>	0.637 <sup>†</sup>	0.649 <sup>†</sup>	0.499 <sup>†</sup>	3.456 <sup>†</sup>	0.040 <sup>†</sup>
(4) VHRED-P		5.802 <sup>†</sup>	0.648 <sup>†</sup>	8.345 <sup>†</sup>	2.750 <sup>†</sup>	30.756 <sup>†</sup>	0.558 <sup>†</sup>	0.635 <sup>†</sup>	0.472 <sup>†</sup>	3.772 <sup>†</sup>	<u>0.047</u>
(4) ReCoSa-P		<u>6.113</u>	<u>0.686</u>	<u>8.899</u>	2.593 <sup>†</sup>	25.767 <sup>†</sup>	0.574 <sup>†</sup>	0.632 <sup>†</sup>	<u>0.510</u>	<u>3.998</u>	0.044 <sup>†</sup>
(4) DHAP (ours)		<b>6.858</b>	<b>0.737</b>	<b>11.720</b>	<b>18.707</b>	<b>66.932</b>	<b>0.709</b>	<b>0.721</b>	<b>0.539</b>	<b>4.639</b>	<b>0.111</b>

**Table 2: Human evaluation results on Weibo dataset. “†” denotes the result is significantly worse than our method in t-test with  $p < 0.05$  level. The best results are in bold and the second best results are underlined. The Fleiss Kappa is 0.42.**

Model	Readability	Informativeness	Personalization
(1) Seq2SeqWA	2.10 <sup>†</sup>	1.85 <sup>†</sup>	0.19 <sup>†</sup>
(1) MMI	2.06 <sup>†</sup>	1.88 <sup>†</sup>	0.23 <sup>†</sup>
(2) Speaker	<u>2.14</u>	1.93 <sup>†</sup>	0.25 <sup>†</sup>
(2) PersonaWAE	2.07 <sup>†</sup>	1.99 <sup>†</sup>	0.36 <sup>†</sup>
(3) GPMN	2.12 <sup>†</sup>	1.92 <sup>†</sup>	0.35 <sup>†</sup>
(3) PerCVAE	2.04 <sup>†</sup>	<u>2.01</u>	0.39 <sup>†</sup>
(4) VHRED-P	2.09 <sup>†</sup>	1.96 <sup>†</sup>	<u>0.47</u>
(4) ReCoSa-P	2.12 <sup>†</sup>	1.93 <sup>†</sup>	<u>0.44</u>
(4) DHAP (ours)	<b>2.26</b>	<b>2.09</b>	<b>0.56</b>
Ground-truth	2.69	2.35	0.84

(2) Following [8], we employ Dist-1 and Dist-2 to evaluate the diversity of the generated response. Responses with more distinct unigrams/bigrams will have higher Dist-1/Dist-2. (3) As suggested by [4], we use three embedding-based metrics to measure the semantic relevance between the generated response and the ground-truth response. Concretely, we use the bag-of-words embeddings to represent both the generated and ground-truth response, and calculate their average similarity (Ave.), greedy similarity (Gre.), and extrema similarity (Ext.). The pre-trained word embeddings for Weibo and Reddit corpus are offered by Li et al. [10] and Pennington et al. [21], respectively. (4) Furthermore, since the goal of our model is to leverage user history for personalization, we evaluate the personalized performance by measuring how much information in the dialogue

history is reflected in the generated response. Following [11, 15], we use Persona F1 (P-F1) to measure the unigram F1 between the generated response and user’s historical responses. Thus, the more historical words the generated response contains, the higher P-F1 we will get. Since the importance of the shared words can be different, following [30], we further use Persona Coverage (P-Cover) to measure the IDF-weighted word overlap between generated response and dialogue history. Specifically, for  $n$  historical responses  $\{R_1, \dots, R_n\}$  and the generated response  $Y$ , P-Cover is defined as:

$$\text{P-Cover} = \max_{j \in [1, n]} \frac{\sum_{w_k \in W_j} \text{IDF}(w_k)}{|Y|}, \quad (16)$$

where  $W_j$  is the set of shared words between  $R_j$  and  $Y$ .

**Human Evaluation:** The automatic evaluation metrics can measure the quality of the generated response with respect to the ground-truth. However, due to the diversity of human dialogues, a response different from the ground-truth may also be acceptable. Thus, we randomly sample 100 test samples to conduct human evaluations. We present the generated responses, the corresponding post, and the user’s historical post-response pairs to three well-educated annotators. The annotators will evaluate the quality of the generated responses in a double-blind fashion. Following [4], the evaluation criterion includes: (1) Readability, which measures the grammatical correctness and smoothness of generated responses; (2) Informativeness, which measures whether the responses are informative or trivial; and (3) Personalization, which measures if the response can reflect personalized information (sharing some information with the history of the user). For the former two perspectives, we use a score 1/2/3 for bad/normal/good quality. For personalization, we use the score 0/1 to judge whether a response

**Table 3: Performance of ablation models on Weibo dataset. “†” denotes the result is significantly worse than our method in t-test with  $p < 0.05$  level. The best results are denoted in bold font.**

Model	Word Overlap			Diversity		Embedding Similarity			Personalization	
	BLEU-1	BLEU-2	ROUGE-L	Dist-1	Dist-2	Average	Extrema	Greedy	P-F1(%)	P-Cover
DHAP	<b>9.324</b>	<b>0.894</b>	<b>14.122</b>	<b>15.175</b>	<b>58.806</b>	<b>0.523</b>	<b>0.747</b>	<b>0.313</b>	7.013	0.144
w/o G	7.726†	0.801†	11.815†	12.176†	49.808†	0.495†	0.707†	0.294†	6.179†	0.107†
w/o D	8.503†	0.855†	12.610†	13.699†	54.623†	0.499†	0.713†	0.303†	6.286†	0.109†
w/o PC	8.830	0.868†	13.981	14.457	56.263†	0.503†	0.728†	0.301†	6.884	0.120†
w/o GEN	4.982†	0.328†	9.571†	9.051†	32.566†	0.478†	0.571†	0.276†	<b>9.331</b>	<b>0.165</b>
w/o COP	8.347†	0.837†	12.585†	13.487†	52.087†	0.499†	0.717†	0.298†	6.234†	0.110†
w FIX	8.549†	0.855†	12.871†	13.904†	54.539†	0.496†	0.716†	0.301†	6.326†	0.113†

reflects personalized information or not. The Fleiss Kappa is 0.42 that indicates the annotators achieve a substantial agreement.

#### 4.4 Implement Details

To determine the parameters of the model, we conducted multiple sets of experiments. The final parameters are selected as follows. For all datasets, we use 512 as the hidden size of GRU, 0.001 as the learning rate. The hidden size and number of heads of Transformer are 256 and 8. The number of Transformer layers  $N = 6$ . The history length is set to 25. The vocabulary size is limited to 40,000. The word embedding dimension is 300/100 for Weibo/Reddit datasets, respectively. We use the Adam optimizer with a batch size of 256. We train all models for 10 epochs and select the best model based on the validation results on BLEU-1.

#### 4.5 Experimental Results

*4.5.1 Automatic Evaluation.* All evaluation results under automatic metrics are reported in Table 1. We can observe that:

(1) Among all models, **DHAP achieves the best results in terms of all evaluation metrics.** DHAP improves performance with a large margin over two strongest baselines VHRED-P and ReCoSa-P, which can also learn implicit user profile. Concretely, DHAP significantly outperforms ReCoSa-P by 28.4%/12.2% improvements in BLEU-1 on Weibo/Reddit dataset. The reason for the improvement reduction on Reddit set is that it has a larger scale and more varied conversations, which leads to more noise. Besides, for the embedding similarity metrics, DHAP also outperforms the best baselines. These results demonstrate that DHAP can generate more semantically relevant responses to the ground-truth by leveraging user’s history. Furthermore, DHAP has dramatic improvements of Dist-1/2, indicating DHAP can generate more informative and diverse responses based on the personalized information. All these results prove that learning implicit user profiles from user’s dialogue history can improve the quality of generated responses.

(2) All personalized methods outperform non-personalized methods, indicating that **personalization is helpful for generating more informative and relevant responses.** Seq2SeqWA generally has the lowest performance, reflecting that the semantic information in the post is insufficient for generating an informative response. MMI improves the diversity performance significantly, but loses some ability on modeling semantic relevance, as it changes

the training objective. Speakers and PersonaWAE use user ID embeddings to identify different users for personalization, and outperform the non-personalized methods. The explicit persona-based model GPMN and PerCVAE show comparable performance to the user ID embedding based baselines. A potential reason is that they are designed for leveraging explicit user profile, which is usually of high quality. In our case, they are only provided with the user’s historical responses, which are much noisy. Therefore, existing personalized methods for explicit user profile is not appropriate for dealing with the implicit user profile contained in the user history.

(3) Among all personalized methods, **the ones using implicit user profile perform better.** VHRED-P and ReCoSa-P show better performance on most metrics, confirming that the dialogue history can be used to mine implicit user profile for a specific user. However, these two methods are originally proposed for multi-turn dialogue generation. Their performance in personalized tasks is limited because the dialogue history covers far more aspects than the context in multi-turn dialogue. On the contrary, our DHAP models the personalized information in both encoder and decoder side, and consider the implicit user profile in both general and dynamic style. Hence, DHAP can achieve significant improvements compared with existing personalized baselines.

*4.5.2 Human Evaluation.* We also conduct a human evaluation for all models on Weibo dataset. The results are shown in Table 2. Generally, DHAP achieves significant improvements in terms of all perspectives, which is consistent with the results on automatic metrics. In particular, we find that DHAP is much better than ReCoSa-P in terms of personalization. This is because DHAP learns the implicit user profile more comprehensively and enhances the influence of personalized words directly in the decoder. DHAP also performs better than other baselines in terms of readability, which shows that DHAP is better at language understanding with the help of user history. Besides, the informativeness of responses generated by DHAP is also improved. This demonstrates that leveraging personalized information is effective to generate more meaningful responses.

In summary, the automatic and human evaluation results strongly verify that **dialogue history is suitable to build the user profile implicitly, and leveraging implicit user profiles is effective to generate meaningful and personalized responses, further achieving a personalized chatbot.**



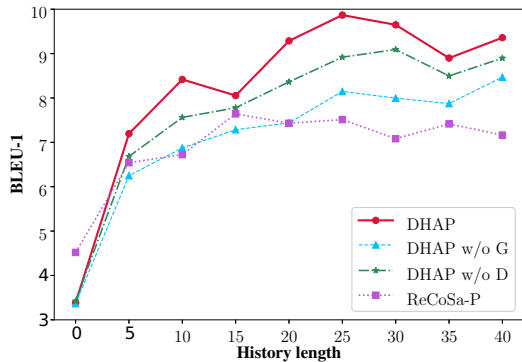


Figure 3: Effectiveness of DHAP on users with different lengths of history for Weibo dataset.

## 4.6 Further Analysis

We further analyze the influence of different modules (Section 4.6.1) and the performance over different history lengths (Section 4.6.2). Both of these experiments are performed on Weibo dataset.

**4.6.1 Ablation Study.** DHAP learns several personalized user profiles based on the dialogue history and designs a decoding switcher and two decoding strategies in the personalized decoder. We remove one of them once a time to analyze its contribution. The experiment results on Weibo dataset are shown in Table 3.

**The ablation on personalized representations.** Three settings are considered: (1) **without G**: the general user profile  $e^G$  is not used; (2) **without D**: the dynamic post-aware user profile  $e_t^D$  is not used; and (3) **without PC**: the post encoder is non-personalized, namely initializing the post encoder with random states rather than the general user profile.

The results show that all of the personalized representations are useful. Specifically, removing the general user profile causes the most decline in all metrics, which confirms the necessity and contribution of it on summarizing personalized information in a general view. The performance degradation caused by removing the dynamic post-aware user profile shows that selecting historical responses relevant to the input post contributes to the further enhancement of user modeling. The influence of removing personalization in post encoder is relatively smaller. It proves that using the user profile to enhance the understanding of the current post is effective but limited, since such information is only provided at the beginning step and decreases with the hidden state update.

**The ablation of components in the personalized decoder.** We test the following variants of our model: (1) **without GEN**: the general decoding is banned; (2) **without COP**: the copying mode is banned; (3) **with FIX**: the probability of two modes are fixed. Specifically, the general decoding probability is set as 0.8 and the copy decoding probability is 0.2, these probabilities are set according to the best results of DHAP in our preliminary experiments.

It can be seen that the results of three variants all underperform the whole framework. Without general decoding, the performance of DHAP drops sharply in terms of all metrics except personalization metrics. Specifically, it drops 46.55% in terms of BLEU-1. This

indicates that only copying words from personalized vocabulary is unable to generate a suitable response, because there are lots of noises irrelevant to the current post and some general words may not be contained in the vocabulary. The reason for its improvement on personalized metrics is that all of the generated words are copied from the history, regardless of the significant hurt on relevance and diversity. Thus, general decoding considering both the post, decoding states, and personalized information is necessary. However, only using general decoding also hurts the performance, which indicates the words reflecting user personalized information is also very valuable. To combine the two decoding strategies, DHAP calculates the possibilities of two decoding strategies dynamically. It works well in DHAP yet using fixed probabilities has lower performance.

**4.6.2 Performance across Various Lengths of Dialogue History.** As we leverage user’s dialogue history for personalization, the length of history may affect the model’s performance. To investigate the influence of history length, we test the performance of DHAP by using different numbers of historical post-response pairs. The results of BLEU-1 on Weibo set are illustrated in Figure 3. We find:

(1) In general, DHAP performs better when a user has a longer dialogue history. This is consistent with our speculation as a longer dialogue history can provide more personalized information. DHAP achieves the best performance with the history length around 25. Unfortunately, when more than 30 historical pairs are used, the performance of DHAP becomes unstable. The potential reason is that more historical data may bring more noise and increase the difficulty of building an effective user profile. (2) When the history is less than 5, ReCoSa-P performs better than DHAP without general user profile. This is because the persona information is extremely limited. Under this circumstance, the more complex architecture of ReCoSa shows its superiority. Nevertheless, our DHAP still performs best, showing its scalability for various history lengths.

## 5 CONCLUSION

In this work, we implemented response generation of personalized chatbots in an alternative way. Different from existing personalized methods, we propose the personalized model DHAP, which learns the implicit user profile automatically from large-scale user dialogue history. We design a personalized language model to capture the user’s general interest from their historical responses and summarize the general user profile. To further highlight the historical responses which are relevant and valuable to the current input post, we build a history memory and construct the dynamic post-aware user profile. We build a personalized decoder to coordinate two personalized decoding strategies. Experimental results confirm the effectiveness of our model on generating informativeness and personalized responses.

## ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 61872370 and No. 61832017, and Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Shandong Provincial Natural Science Foundation under Grant ZR2019ZD06.

## REFERENCES

- [1] 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. *CoRR* abs/1606.00372 (2016). arXiv:1606.00372
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR 2015*.
- [3] JinYeong Bak and Alice Oh. 2019. Variational Hierarchical User-based Conversation Model. In *EMNLP-IJCNLP*. 1941–1950.
- [4] Zhangming Chan, Juntao Li, Xiaopeng Yang, et al. 2019. Modeling Personalization in Continuous Space for Response Generation via Augmented Wasserstein Autoencoders. In *EMNLP-IJCNLP*. 1931–1940.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the NAACL-HLT 2019*. 4171–4186.
- [6] James H. Gerlach and Feng-Yang Kuo. 1991. Understanding Human-Computer Interaction for Information Systems Design. *MIS Q* 15, 4 (1991), 527–549.
- [7] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *ACL 2016*. ACL.
- [8] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the NAACL HLT 2016*. 110–119.
- [9] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A Persona-Based Neural Conversation Model. In *Proceedings of the ACL 2016*. ACL.
- [10] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, et al. 2018. Analogical Reasoning on Chinese Morphological and Semantic Relations. In *ACL*. 138–143.
- [11] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. In *IJCAI 2019*.
- [12] Chin-Yew Lin and Franz Josef Och. 2004. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In *Proceedings of the ACL 2004*. ACL, 605–612.
- [13] Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. 2020. You Impress Me: Dialogue Generation via Mutual Persona Perception. In *Proceedings of the ACL 2020*. ACL, 1417–1427.
- [14] Liangchen Luo, Wenhao Huang, Qi Zeng, Zaiqing Nie, and Xu Sun. 2019. Learning Personalized End-to-End Goal-Oriented Dialog. In *AAAI 2019*. 6794–6801.
- [15] Pengcheng Lv, Shi Feng, et al. 2020. PersonaGAN: Personalized Response Generation via Generative Adversarial Networks. In *DASFAA 2020*. Springer, 570–586.
- [16] Zhengyi Ma, Zhicheng Dou, Guanyue Bian, and Ji-Rong Wen. 2020. PSTIE: Time Information Enhanced Personalized Search. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1075–1084. <https://doi.org/10.1145/3340531.3411877>
- [17] François Mairesse and Marilyn A. Walker. 2007. PERSONAGE: Personality Generation for Dialogue. In *Proceedings of the ACL 2007*. ACL.
- [18] Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the EMNLP 2016*. ACL, 1400–1409.
- [19] Masato Neishi and Naoki Yoshinaga. 2019. On the Relation between Position Information and Sentence Length in Neural Machine Translation. In *CoNLL*.
- [20] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*. 311–318.
- [21] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP 2014*. ACL, 1532–1543.
- [22] Hongjin Qian, Xiaohe Li, Hanxun Zhong, Yu Guo, Yueyuan Ma, Yutao Zhu, Zhanliang Liu, Zhicheng Dou, and Ji-Rong Wen. 2021. Pchatbot: A Large-Scale Dataset for Personalized Chatbot. In *Proceedings of the SIGIR 2021*. ACM. <https://doi.org/10.1145/3404835.3463239>
- [23] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning Personality/Profile to a Chatting Machine for Coherent Conversation Generation. In *Proceedings of the IJCAI 2018*. 4279–4285.
- [24] Minghui Qiu, Feng-Lin Li, Siyu Wang, et al. 2017. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In *ACL 2017*. ACL, 498–503.
- [25] Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-Driven Response Generation in Social Media. In *Proceedings of the EMNLP 2011*. ACL, 583–593.
- [26] Iulian Vlad Serban, Alessandro Sordani, et al. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *AAAI 2016*.
- [27] Iulian Vlad Serban, Alessandro Sordani, Ryan Lowe, et al. 2017. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. In *AAAI 2017*.
- [28] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the ACL 2015*. ACL, 1577–1586.
- [29] Haoyu Song, Yan Wang, Weinan Zhang, Xiaojiang Liu, and Ting Liu. 2020. Generate, Delete and Rewrite: A Three-Stage Framework for Improving Persona Consistency of Dialogue Generation. In *Proceedings of the ACL 2020*. 5821–5831.
- [30] Haoyu Song, Weinan Zhang, et al. 2019. Exploiting Persona Information for Diverse Generation of Conversational Responses. In *IJCAI 2019*. 5190–5196.
- [31] Haoyu Song, Wei-Nan Zhang, et al. 2020. Generating Persona Consistent Dialogues by Exploiting Natural Language Inference. In *AAAI 2020*. 8878–8885.
- [32] Alessandro Sordani, Michel Galley, Michael Auli, Chris Brockett, et al. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *Proceedings of the NAACL HLT 2015*. ACL, 196–205.
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, et al. 2017. Attention is All you Need. In *NIPS*. 5998–6008.
- [34] Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *CoRR* abs/1506.05869 (2015). arXiv:1506.05869
- [35] Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue Natural Language Inference. In *Proceedings of the ACL 2019*. ACL, 3731–3741.
- [36] Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. *CoRR* abs/1901.08149 (2019).
- [37] Michael J. Wooldridge and Nicholas R. Jennings. 1995. Intelligent agents: theory and practice. *Knowl. Eng. Rev.* 10, 2 (1995), 115–152.
- [38] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots. In *Proceedings of the ACL 2017*. ACL, 496–505.
- [39] Rui Xia, Mengran Zhang, and Zixiang Ding. 2019. RTHN: A RNN-Transformer Hierarchical Network for Emotion Cause Extraction. In *IJCAI 2019*. 5285–5291.
- [40] Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019. ReCoSa: Detecting the Relevant Contexts with Self-Attention for Multi-turn Dialogue Generation. In *Proceedings of the ACL 2019*. ACL, 3721–3730.
- [41] Saizhong Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too? In *Proceedings of the ACL 2018*. ACL, 2204–2213.
- [42] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, et al. 2020. DIALOGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the ACL 2020*. ACL, 270–278.
- [43] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory. In *Proceedings of the AAAI 2018*. AAAI Press, 730–739.
- [44] Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense Knowledge Aware Conversation Generation with Graph Attention. In *Proceedings of the IJCAI 2018*. ijcai.org.
- [45] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2018. The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. *CoRR* (2018).
- [46] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Encoding History with Context-aware Representation Learning for Personalized Search. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1111–1120. <https://doi.org/10.1145/3397271.3401175>
- [47] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Enhancing Re-finding Behavior with External Memories for Personalized Search. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 789–797. <https://doi.org/10.1145/3336191.3371794>
- [48] Yutao Zhu, Zhicheng Dou, Jian-Yun Nie, and Ji-Rong Wen. 2020. ReBoost: a retrieval-boosted sequence-to-sequence model for neural response generation. *Inf. Retr. J.* 23, 1 (2020), 27–48. <https://doi.org/10.1007/s10791-019-09364-x>
- [49] Yutao Zhu, Jian-Yun Nie, Kun Zhou, Pan Du, and Zhicheng Dou. 2021. Content Selection Network for Document-Grounded Retrieval-Based Chatbots. In *Proceedings of ECIR 2021 (Lecture Notes in Computer Science)*, Vol. 12656. Springer, 755–769. [https://doi.org/10.1007/978-3-030-72113-8\\_50](https://doi.org/10.1007/978-3-030-72113-8_50)
- [50] Yutao Zhu, Ruihua Song, Zhicheng Dou, Jian-Yun Nie, and Jin Zhou. 2020. ScriptWriter: Narrative-Guided Script Generation. In *Proceedings of ACL 2020*. Association for Computational Linguistics, 8647–8657. <https://doi.org/10.18653/v1/2020.acl-main.765>