Learning Implicit User Profiles for Personalized Retrieval-Based Chatbot

Hongjin Qian[1], Zhicheng Dou[1*], Yutao Zhu[3], Yueyuan Ma[1], Ji-Rong Wen[1,2]
[1] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2] Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China
[3] Université de Montréal, Québec, Canada
{ian,*dou}@ruc.edu.cn

ABSTRACT

In this paper, we explore the problem of developing personalized chatbots. A personalized chatbot is designed as a digital chatting assistant for a user. The key characteristic of a personalized chatbot is that it should have a consistent personality with the corresponding user. It can talk the same way as the user when it is delegated to respond to others’ messages. Many methods have been proposed to assign a personality to dialogue chatbots, but most of them utilize explicit user profiles, including several persona descriptions or key-value-based personal information. In a practical scenario, however, users might be reluctant to write detailed persona descriptions, and obtaining a large number of explicit user profiles requires tremendous manual labour. To tackle the problem, we present a retrieval-based personalized chatbot model, namely IMPChat, to learn an implicit user profile from the user’s dialogue history. We argue that the implicit user profile is superior to the explicit user profile regarding accessibility and flexibility. IMPChat aims to learn an implicit user profile through modeling user’s **personalized language style** and **personalized preferences** separately. To learn a user’s personalized language style, we elaborately build language models from shallow to deep using the user’s historical responses; To model a user’s personalized preferences, we explore the conditional relations underneath each post-response pair of the user. The personalized preferences are dynamic and context-aware: we assign higher weights to those historical pairs that are topically related to the current query when aggregating the personalized preferences. We match each response candidate with the personalized language style and personalized preference, respectively, and fuse the two matching signals to determine the final ranking score. We conduct comprehensive experiments on two large datasets, and the results show that our method outperforms all baseline models. The codes and datasets will be released at Github[1].

CCS CONCEPTS

• **Computing methodologies → Discourse, dialogue and pragmatics**.

KEYWORDS

Personalization, Retrieval-based Chatbots, Implicit User Profile

ACM Reference Format:
Hongjin Qian, Zhicheng Dou, Yutao Zhu, Yueyuan Ma, and Ji-Rong Wen. 2021. Learning Implicit User Profiles for Personalized Retrieval-Based Chatbot. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM ’21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482269

1 INTRODUCTION

Building an open-domain dialogue system is an intriguing but challenging task that draws more and more attention in both academic and industrial communities. In general, there are two pathways to construct an open-domain chatbot: retrieval-based and generation-based. The former usually retrieves relevant response candidates via a retrieval engine and then selects a proper response from these candidates [19, 38, 41, 43, 56]. The latter directly leverages models, such as encoder-decoder, to generate response texts [32, 40, 41]. In this paper, we concentrate on the retrieval-based chatbots which are wildly applied in many industrial products such as Microsoft XiaoIce [54] and E-commerce assistant AliMe [14].

Although plentiful chatbot applications are available, there still exist some challenges that cannot be overlooked [5, 8]. Inconsistent personality is one of the most mentioned challenges of current open-domain chatbots [7, 35]. In the dialogue chatbot domain, personality refers to a set of traits that define a chatbot’s interaction style, character, and behaviors [7, 18]. Inconsistent personality brings more unpredictability and untrustworthiness to the chatbot, which can disorient end users and lead to a strong sense of discomfort [7, 29]. Therefore, endowing chatbots with consistent personality becomes a crucial research task that guarantees a chatbot’s performance in agreement with user’s expectations. To this end, many methods have been proposed to assign a personality to dialogue chatbots [1, 3, 16, 31, 47]. In this paper, we address the problem of developing “personalized chatbots”. A personalized chatbot is created for a specific user and it functions like the digital chatting assistant of the user. Ideally, it can talk exactly the same way as the user, and can respond to others’ messages when it is delegated by the user. We argue that for such a personalized chatbot,

[1] https://github.com/qhjqhj00/CIKM2021-IMPChat

CIKM ’21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482269

Table 1: An example of a user's dialogue history.

Post	Response
(1) What's your favorite programming language?	(1) Mine is Java.
(2) Rafa wins his 1000 match!	(2) Bravo Rafa!
(3) PC or MAC for college students?	(3) If ur in steam, go windows.
(4) I failed an exam again and feel like a loser.	(4) Good for you.
(5) Australian Open Final Nadal vs Federer!	(5) Vamos, Nadal.

it is extremely important to have a consistent personality with the corresponding user.

There have been a few attempts for developing personalized chatbots. Early studies encoded user information by using user ID embeddings which are learned during training [1, 4, 16], and generate personalized responses with the help of the ID embeddings. Recently, some works focused on building personalized chatbots using manually created user profiles to maintain the consistency of personality for chatbots [11, 13, 17, 23, 36, 37, 48]. Such predefined user profiles are *explicit* user profiles which usually contain several persona descriptions or key-value-based personal information.

In this paper, we propose using the user's dialogue history to automatically build **implicit user profiles** to endow consistent personality to the chatbot. The reasons mainly lie in two aspects. **First**, regarding accessibility and flexibility, the implicit user profile is superior to the explicit user profile. Obtaining a large number of explicit user profiles requires tremendous manual labour [4]. In a practical scenario, users also might be reluctant to write such detailed persona descriptions [9]. Besides, compared with explicit user profiles, implicit user profiles are easier to update with the accumulation of dialogue history as they are automatically learned. **Second**, tremendous personalized style information is usually hidden underneath the user's dialogue history. Here, the personalized style refers to the unique personal characteristics (*e.g.*, speaking style, background knowledge, and preferences) that help distinguish a user from others. Such personalized style information is beneficial to enhance the chatbot's ability to output coherent responses. Table 1 shows an example of a user's dialogue history from which we can find the user's personalized speaking style and preferences. Taking the fifth post "Australian Open Final Nadal vs Federer!" as an example, the user replies "Vamos[2], Nadal". Thus, we know that the user likes the tennis player Nadal instead of Federer. And the user likes using "Vamos" to cheer the sports player on.

Inspired by the beneficial properties of implicit user profiles, we design a model IMPChat to learn the **IM**plicit user profile from a user's dialogue history for **P**ersonalized retrieval-based **Chat**bots. Through the model, we propose modeling the implicit user profile from two aspects: personalized language style and personalized preference. The former aims to summarize personalized language characteristics (*e.g.*, using "Vamos" to cheer sports player on) from the dialogue history without considering the current message. The latter considers the conditional relation between the current message and the user's personal preference (*e.g.*, like the tennis player Nadal instead of Federer) by building a post-aware user profile. Then, we perform matching between the response candidates and the learned implicit user profile to select a personalized response.

Specifically, we **first** design a **Personalized Style Matching** module that uses multi-layer attentive modules to capture multi-grained personalized language characteristics. At each layer, we simultaneously perform self-attention within historical responses and cross-attention with the corresponding historical posts. Through this way, we not only obtain deep personalized style features but understand the context-aware relations between posts and responses. We perform matching between the response candidates and the personalized style. **Second**, we build a **Post-Aware Personalized Preference Matching** module. Given the current query, this module first selects topically related dialogue history. Then, it models the personalized preferences from the selected dialogue history to enhance the personalized response selection. The dialogue history selection is performed by measuring the topical relatedness between the current query and each historical post. We think that the word ambiguity of the current query might bias the topical relatedness. To deal with the bias, we create a tailored multi-hop method to stretch the current query's semantic richness by fusing context information into the current query. We then perform matching between the response candidates and the post-aware personalized preference. **Finally**, in the fusion module, we combine the two matching features to determine the most proper response. To validate the effectiveness of our model, we conduct comprehensive experiments on two publicly available datasets for personalized response selection. Experimental results show that our model outperforms all baseline models.

Our contributions are threefold: (1) To the best of our knowledge, this is the first study that seeks to build an implicit user profile using the user's dialogue history to enhance personality consistency for the retreival-based chatbot. (2) We propose IMPChat that constructs an implicit user profile considering the user's personalized language style and personalized preference. (3) Extensive experiments show that our model outperforms the state-of-the-art models.

2 RELATED WORK

2.1 Retrieval-Based Chatbots

To build an open-domain dialogue chatbot, there exist two major research directions: generation-based and retrieval-based. The first usually learns to generate responses with an encoder-decoder structure [15, 27, 33, 34, 40, 60]. The second learns a matching model to select proper responses from candidates given input query and context [19, 38, 44, 55, 62, 63]. In this paper, we focus on the latter.

The general objective of retrieval-based methods is to select a proper response from a candidate list based on a given query[3]. The candidates are usually pre-retrieved by an index system. Along this direction, early studies mainly focus on single-turn response selection [6, 12, 41, 42]. Recently, study interests move to multi-turn response selection, which uses dialogue session information as context to enhance the single-turn matching. For example, Dual-LSTM model encodes context and responses into hidden states and the last hidden states are used to compute relevance score [19]. Multi-view model [55] performs multi-grained context-response matching.

[2] "Vamos" is a Spanish word which means "let's go".

[3] Here, "query" has different meanings with that often used in IR.

These methods concatenate the context to a long sequence which may undermine the relationships among utterances. To tackle this problem, the representation-matching-aggregation framework is then applied by many recent works, which conducts interaction between the response and each utterance in the context and then aggregate these matching features using CNN or RNN [44]. Following the framework, deep utterance aggregation network (DUA) [50] designs turns-aware aggregation mechanism that assigns different weights to each dialogue turn. Deep attentive matching network (DAM) jointly uses self-attention and cross-attention to obtain multi-grained representations [56]. Multi-hop selection network (MSN) [46] designs a multi-hop method to make the context selection more robust. Besides, the interaction-over-interaction network (IoI) argues that one time of interaction captures limited matching features, and the IoI network stacks multiple interaction blocks [38].

2.2 Personalized Chatbots

Endowing personality to open-domain chatbots is intriguing but challenging. It is an inevitable barrier to achieve truly applicable intelligent assistant. With a consistent personality, the ability of a chatbot to be precise regarding how to use language would be greatly improved [20, 26]. Early work tries to model personality using heuristic methods such as "Big Five" of speakers [21] . With the rising of deep learning recently, many end-to-end neural models are designed to tackle the challenge. Basically, these methods can be categorized into two groups. The first group learns a user embedding for each user. The user embedding is then used to conduct matching or generation [1, 3, 16]. Albeit the user embeddings are updating through training, they cannot directly interact with dialogue texts which contain abundant personalized information.

The second group learns personality using explicit user profile (persona) which contains either several persona sentences or key-value agent profile [31, 47]. These explicit user profiles are either human-annotated or extracted from the dialogue history using rule-based methods. PERSONA-CHAT is such a dataset that is widely used [47]. The major problem is how to use persona sentences (documents) and the context properly. Along with the dataset, a key-value profile memory network is used, which takes the dialogue history as input and performs attention over persona sentences [25, 47]. DGMN lets the context and document attend to each other to obtain fused representations [52]. CSN performs document selection over the persona sentences [61]. RSM-DCK conducts selection over both dialogue context and user profiles and lets the response candidates interact with the selected results [13].

Constructing a dataset like PERSONA-CHAT is exhausted. Some other works tackle this issue by generating user profiles through rule-based methods [23, 53]. For example, Mazaré et al. [23] construct much larger datasets with explicit user profile using Reddit[4] dataset in a heuristic way, and it empirically proves that models training on such automatically-generated user profile with enough data can achieve similar or even better performance. Besides, the topic of personalization has also been explored in many other scenarios such as personalized search and user modelling [57–59].

Unlike the studies above, in this paper, we seek to build an implicit user profile using the user's dialogue history. Such implicit user profile contains rich personalized information and can be automatically learned from the historical dialogues.

3 METHODOLOGY

Assigning personality to chatbots is crucial to satisfy user's expectations. As introduced in Section 1, most current works seek to model personality with explicit user profiles. In this paper, we instead study the problem of automatically modeling implicit profiles of a user based on the user's dialogue history, and leverage the implicit profile to select the personalized response. The chatbot acts like a virtual agent of the user and is able to reply messages when the user is busy and asks the chatbot to generate responses.

3.1 Preliminary

The general goal of a retrieval-based chatbot is to return the most proper response within a database for an input message. Assume g is a scoring model evaluating the quality of a candidate response r for a given message q under the context C, the chatbot will output the response r^* with the highest value of g from a repository of response $\mathcal{R}$, *i.e.*, we have:

$$r^* = \underset{r \in \mathcal{R}}{\arg\max}\, g(q, r, C).$$

For a single-turn chatbot, $C = \phi$ and the response is selected merely based on the input message q. For a multi-turn chatbot, C is the conversation context comprised of previous turns of dialogues. In this work, we investigate the problem of retrieving a personalized response that matches the speaking style and knowledge background of a user u. Suppose $\mathcal{U}$ is the corresponding profile of u, and we have $C = \mathcal{U}$. $\mathcal{U}$ is usually a compact representation of a user's interests and knowledge. Hence, the personalized chatbot aims to return the response with the highest value of $g(q, r, \mathcal{U})$.

There are multiple ways to model $\mathcal{U}$. Previous works either learn user ID embeddings [1, 3, 16] or build explicit user profiles[13, 47, 52, 61]. For example, Zhang et al. [47] used explicit persona descriptions and Qian et al. [31] used key-value profile to model $\mathcal{U}$. In this paper, we instead model $\mathcal{U}$ using the user's dialogue history. Suppose $H = \left\{(p_j, r_j)\right\}, j \in [1, t]$ is the user's dialogue history where p_j represents posts issued by others and r_j means responses made by the current user, we propose selecting responses by:

$$g(q, r, \mathcal{U}) = g(q, r, \mathcal{F}(H)),$$

where $\mathcal{F}(\cdot)$ defines a model that learns implicit user profile from dialogue history. Note that we further denote the user's historical posts and responses in H as $P = \{p_j\}$ and $R = \{r_j\}$ respectively.

3.2 IMPChat Overview

Intuitively, there are two major factors that impact the manner of expressing. First, a user has a personalized language style regardless of the conversation context. Such language style is usually determined by the user's knowledge background and preferred expressions. Second, the way a user makes responses is also conditioned on the intrinsic relations between a post and the user's personalized preferences. Given the same dialog context (e.g., a post), different users may give different responses and we try to learn the personalized preferences from history.

[4] https://www.reddit.com/r/datasets/ comments/3bxlg7/

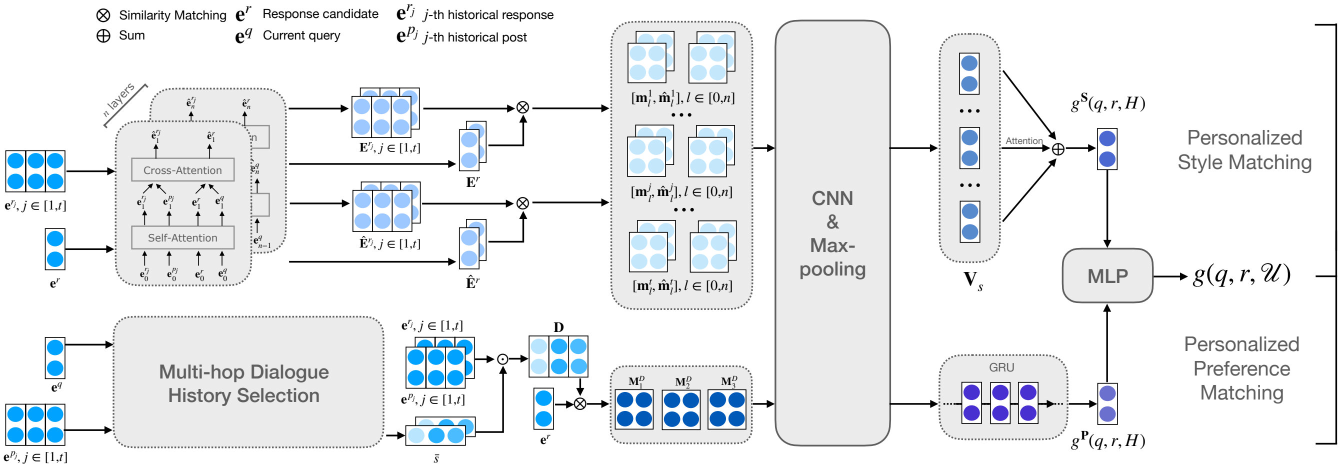

⊗ Similarity Matching
⊕ Sum
$\mathbf{e}^r$ Response candidate
$\mathbf{e}^q$ Current query
$\mathbf{e}^{r_j}$ j-th historical response
$\mathbf{e}^{p_j}$ j-th historical post
n layers
$\mathbf{e}^{r_j}, j \in [1,t]$
$\mathbf{e}^r$
Cross-Attention
Self-Attention
$\hat{\mathbf{e}}_n^{r_j}$
$\hat{\mathbf{e}}_n^r$
$\hat{\mathbf{e}}_1^{r_j}$
$\hat{\mathbf{e}}_1^r$
$\mathbf{e}_1^{r_j}$ $\mathbf{e}_1^{p_j}$ $\mathbf{e}_1^r$ $\mathbf{e}_1^q$
$\mathbf{e}_0^{r_j}$ $\mathbf{e}_0^{p_j}$ $\mathbf{e}_0^r$ $\mathbf{e}_0^q$
$\mathbf{e}_n^q$
$\mathbf{e}_{n-1}^q$
$\mathbf{E}^{r_j}, j \in [1,t]$
$\mathbf{E}^r$
$\hat{\mathbf{E}}^{r_j}, j \in [1,t]$
$\hat{\mathbf{E}}^r$
$[\mathbf{m}_l^1, \hat{\mathbf{m}}_l^1], l \in [0,n]$
$[\mathbf{m}_l^j, \hat{\mathbf{m}}_l^j], l \in [0,n]$
$[\mathbf{m}_l^t, \hat{\mathbf{m}}_l^t], l \in [0,n]$
CNN & Max-pooling
$\mathbf{V}_s$
Attention
$g^{\mathbf{S}}(q,r,H)$
Personalized Style Matching
MLP
$g(q,r,\mathcal{U})$
Personalized Preference Matching
$\mathbf{e}^q$
$\mathbf{e}^{p_j}, j \in [1,t]$
Multi-hop Dialogue History Selection
$\mathbf{e}^{r_j}, j \in [1,t]$
$\mathbf{e}^{p_j}, j \in [1,t]$
$\bar{s}$
$\mathbf{D}$
$\mathbf{e}^r$
$\mathbf{M}_1^D$ $\mathbf{M}_2^D$ $\mathbf{M}_3^D$
GRU
$g^{\mathbf{P}}(q,r,H)$

Figure 1: The overall Architecture of the proposed model IMPChat which is comprised of the Personalized Style Matching module and the Personalized Preference Matching module. The detail of Multi-hop module is at figure 2.

With this in mind, we assume the personalized user profile $\mathcal{U}$ is comprised of two parts: (1) the personalized language style $\mathbf{S}$ and (2) the post-aware personalized preferences $\mathbf{P}$, and we have:

$$g(q,r,\mathcal{U}) = m\left(g^{\mathbf{S}}\left(q,r,H\right), g^{\mathbf{P}}\left(q,r,H\right)\right),$$

where $g^{\mathbf{S}}$ and $g^{\mathbf{P}}$ return the matching features of the candidate response r regarding the two parts of the personalized user profile.

Figure 1 shows the structure of our proposed IMPChat. Specifically, we first build a **Personalized Style Matching** module (which will be introduced in Section 3.4) which aims to capture the personalized language style using the user's historical responses and get $g^{\mathbf{S}}$ which measures the style consistency of the response candidates. Then, we design a **Post-Aware Personalized Preference Matching** module (in Section 3.5) which learns post-aware user preferences and get $g^{\mathbf{P}}$ which measures the matching degree between the post-aware user preferences and the response candidates. The two modules perform matching separately, and in the **fusion module** (in Section 3.6), we combine the two matching features to compute the final matching score $g(q,r,\mathcal{U})$. In the remaining parts of this section, we will introduce the details of each component.

3.3 Foundation: Attentive Module

We first introduce the structure of Attentive Module, which is a basic component in our method. Following the previous work [46, 56], we use the Attentive Module to endow contextual semantics into word embeddings. Attentive Module is proposed in [56]. It is a variant of Transformer model [39]. Instead of the multi-head attention that is used in the Transformer, the Attentive Module uses single-head attention. Specifically, the Attentive Module takes $\mathbf{Q} \in \mathbb{R}^{t\times d}$, $\mathbf{K} \in \mathbb{R}^{t\times d}$, and $\mathbf{V} \in \mathbb{R}^{t\times d}$ as input, where t and d represent sequence length and embedding dimension respectively.

The Attentive Module defines an attention function $\text{Att}(\mathbf{Q},\mathbf{K},\mathbf{V})$ to map the query $\mathbf{Q}$ and key-value pair $(\mathbf{K},\mathbf{V})$ to a weighted output. The weights are computed by letting each word in the query sentence attend to words in the key sentence via scaled dot production [39], which can be formulated as:

$$\text{Att}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\cdot\mathbf{K}^\top}{\sqrt{d}})\mathbf{V}. \tag{1}$$

A residual connection with layer normalization [2] is then applied to get a better fused representation $\mathbf{x}$ and prevent vanishing or exploding of gradients. A feed-forward network (FFN) with ReLU [10] activation is further applied to the normalized results as:

$$\text{FFN}(\mathbf{x}) = \text{ReLU}(\mathbf{x}\cdot\mathbf{W}_1 + \mathbf{b}_1)\cdot\mathbf{W}_2 + \mathbf{b}_2, \tag{2}$$

where $\mathbf{x}$ is a 2D tensor with the same shape of the query $\mathbf{Q}$; and $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are parameters. The final output is obtained via a residual connection with normalization between $\text{FFN}(\mathbf{x})$ and $\mathbf{x}$. We denote the whole Attentive Module as $f_{\text{att}}(\cdot,\cdot,\cdot)$.

3.4 Personalized Style Matching

A personalized chatbot should coherently output responses that portray consistent personalized styles (*e.g.*, speaking style and vocabulary). We think that these personalized styles are very helpful when we expect to retrieve personalized responses. Taking the pet phrase (a type of speaking style) as an example, regarding how to express congratulation, some users might use "neat" or "congrats", while the user in Table 1 would like to use "bravo". Users tend to use their preferred expressions more frequently, and these personalized preferred expressions are underneath the user's historical responses. Under such an observation, we design a *Personalized Style Matching* module that aims to model a user's preferred speaking style from the user's historical responses.

Formally, given a user u with dialogue history $H = \left\{\left(p_j, r_j\right)\right\}$ and the response candidate r, the Personalized Style Matching module aims to get a style matching feature vector $g^{\mathbf{S}}(q,r,H)$ which measures the style consistency between the response candidate r and the historical responses $R = \{r_j\}, j \in [1,t]$. The Personalized Style Matching module achieves the goal via three layers: (1) **Representation**: it extracts multi-grained semantic representations for historical responses and the response candidate; (2) **Matching**: it performs matching at each semantic level, and (3) **Aggregation**: it dynamically fuses matching signals between the response candidate and all historical responses to obtain $g^{\mathbf{S}}(q, r, H)$. We will introduce each layer in detail as follows.

The **representation layer** aims to obtain multi-grained contextual representations $\mathbf{E} = \{\mathbf{e}_0, \cdots, \mathbf{e}_n\}$ and cross-attention representations $\hat{\mathbf{E}} = \{\hat{\mathbf{e}}_0, \cdots, \hat{\mathbf{e}}_n\}$ for the response candidate r and each historical response $r_j, j \in [1, t]$ using n attentive modules.

Taking the j-th response r_j as an example, the contextual representations $\mathbf{E}^{r_j} = \{\mathbf{e}_0^{r_j}, \cdots, \mathbf{e}_n^{r_j}\}$ model the contextual semantic pattern of r_j from shallow to deep. Specifically, we first initialize word representations $\mathbf{e}_0^{r_j}$ by looking up a word embedding table (*e.g.*, Word2Vec). Then, we obtain deep contextual response representations by feeding the word embeddings into n attentive modules:

$$\mathbf{e}_l^{r_j} = f_{\text{att}}(\mathbf{e}_{l-1}^{r_j}, \mathbf{e}_{l-1}^{r_j}, \mathbf{e}_{l-1}^{r_j}), \quad 1 \le l \le n, \tag{3}$$

where $\mathbf{e}_l^{r_j}$ is the contextual representation output by the l-th attentive module. Through n attentive modules, we obtain the $n+1$ representations which depict the co-occurrence pattern of words at different granularities.

Furthermore, we think that the personalized style of a response is also conditioned on the post. For example, "good for you" is semantically similar to "bravo". However, in Table 1, "good for you" is a response to the post "I failed an exam again and feel like a loser". The user instead uses the phrase in an ironical way. In view of this, we let the response r_j's contextual representation $\mathbf{e}_l^{r_j} \in \mathbf{E}^{r_j}$ attend to the corresponding post p_j's contextual representation $\mathbf{e}_l^{p_j} \in \mathbf{E}^{p_j}$ to obtain the cross-attention representations $\hat{\mathbf{E}}^{r_j} = \{\hat{\mathbf{e}}_0^{r_j}, \cdots, \hat{\mathbf{e}}_n^{r_j}\}$:

$$\hat{\mathbf{e}}_l^{r_j} = f_{\text{att}}(\mathbf{e}_l^{r_j}, \mathbf{e}_l^{p_j}, \mathbf{e}_l^{p_j}), \quad l \in [0, n], \tag{4}$$

where $\mathbf{E}^{p_j}$ is obtained in the same way as $\mathbf{E}^{r_j}$. For the response candidate r, we obtain $\mathbf{E}^r = \{\mathbf{e}_0^r, \cdots, \mathbf{e}_n^r\}$ and $\hat{\mathbf{E}}^r = \{\hat{\mathbf{e}}_0^r, \cdots, \hat{\mathbf{e}}_n^r\}$ in the same way.

The **matching layer** aims to obtain a personalized style matching matrix $\mathbf{M}_s$ which measures the style matching degree between the response candidate and each historical response at multiple granularities. Specifically, given a candidate response r and the j-th historical response r_j, we compute $\mathbf{m}_l^j$ and $\hat{\mathbf{m}}_l^j$ which are the matching matrices for the contextual representations $\{\mathbf{e}_l^{r_j}, \mathbf{e}_l^r\}$ and the cross-attention representations $\{\hat{\mathbf{e}}_l^{r_j}, \hat{\mathbf{e}}_l^r\}$:

$$\mathbf{m}_l^j = \frac{\mathbf{e}_l^{r_j} \cdot \mathbf{e}_l^{r\top}}{\sqrt{d}}, \quad \hat{\mathbf{m}}_l^j = \frac{\hat{\mathbf{e}}_l^{r_j} \cdot \hat{\mathbf{e}}_l^{r\top}}{\sqrt{d}}, \quad l \in [0, n], \tag{5}$$

where d is the dimension of the embeddings and l refers the representation output by the l-th attentive module. Hence, for t historical responses $R = \{r_1, \cdots r_t\}$, we have two groups of multi-grained matching matrices $\mathbf{M}_l = \{\mathbf{m}_l^1, \cdots, \mathbf{m}_l^t\}$ and $\hat{\mathbf{M}}_l = \{\hat{\mathbf{m}}_l^1, \cdots, \hat{\mathbf{m}}_l^t\}$.

To transform these matching matrices into a shared feature space, we first concatenate them into a stacked matching matrix:

$$\mathbf{M}_s = f_{\text{stack}}\left(\{\mathbf{M}_0, \cdots, \mathbf{M}_n, \hat{\mathbf{M}}_0, \cdots, \hat{\mathbf{M}}_n\}\right), \tag{6}$$

where $f_{\text{stack}}(\cdot)$ refers to concatenation along a new dimension. $\mathbf{M}_s \in \mathbb{R}^{2(n+1) \times t \times L \times L}$ and L is the maximum sequence length.

Next, in the **aggregation layer**, following [46, 56, 61], we extract matching features from the matching matrix $\mathbf{M}_s$ via CNN. The extracted feature is then linearly mapped into a lower dimension:

$$\mathbf{V}_s = \text{MLP}\left(f_{\text{CNN}}\left(\mathbf{M}_s\right)\right), \tag{7}$$

where $\mathbf{V}_s \in \mathbb{R}^{t \times d}$, and MLP$(\cdot)$ represents a multi-layer perceptron.

Note that the personalized style matching feature $\mathbf{V}_s$ contains the matching signals between the response candidate r and each historical response r_j. Although post-response pairs are sorted by time in the dialogue history, temporal patterns vanish for historical responses solely. Besides, each historical response may impact differently on personalized style. Therefore, we apply self-attention to dynamically sum up the personalized style matching feature $\mathbf{V}_s$. Finally, we obtain the style matching features $g^{\mathbf{S}}(q, r, H)$:

$$\mathbf{s}_{\text{att}} = \text{softmax}(\text{MLP}(\tanh(\text{MLP}(\mathbf{V}_s)))), \tag{8}$$

$$g^{\mathbf{S}}(q, r, H) = \sum_{dim=0} \mathbf{s}_{\text{att}} \odot \mathbf{V}_s, \tag{9}$$

where $\mathbf{s}_{\text{att}} \in \mathbb{R}^{t \times d}$ represents the attention weights, and $\odot$ is the element-wise multiplication.

3.5 Post-Aware Personalized Preference Matching

In addition to personalized language styles, a user's personalized preferences also have a great impact on the personalized response selection. For example, people who prefer MAC to PC would give a positive comment about a Mac-related post. We think that such personalized preferences tend to be captured from the user's dialogue history as the user's preferences are relatively consistent. Hence, how to properly utilize the dialogue history to enhance the personalized preference consistency is important.

The *Post-Aware Personalized Preference Matching* module aims to obtain a matching vector $g^{\mathbf{P}}(q, r, H)$ which measures whether a response candidate r can consistently reflect the user's personalized preferences given the current query q and a user u with dialogue history H. To properly utilize the dialogue history H, we first transform the dialogue history H into a user profile $\mathbf{D}$ from which we can effectively model the personalized preferences.

Ideally, the user profile $\mathbf{D}$ only contains post-response pairs that are topically related to current post. However, dialogue history reflects multifaceted user interests, some of which might be unrelated to current post. Unrelated context might bring negative impact on response selection [13, 46]. Therefore, we need to filter out the unrelated dialogue history. Intuitively, we can compute a relevance vector $\mathbf{s} \in \mathbb{R}^t$ that measures the topical relatedness between the current post q and historical posts $P = \{p_1, \cdots, p_t\}$. Then,we can obtain the user profile $\mathbf{D}$ by reweighting the dialogue history $\mathbf{H} = \{(\mathbf{e}^{p_j}, \mathbf{e}^{r_j})\}, j \in [1, t]$:

$$\mathbf{D} = \mathbf{s} \cdot \mathbf{H}, \tag{10}$$

where $(\mathbf{e}^{p_j}, \mathbf{e}^{r_j})$ is the representation of the j-th post-response pair.

We compute the relevance vector $\mathbf{s}$ considering two assumptions: (1) topic relatedness can be context-level and word-level; (2) word usage in a post is ambiguous by nature, which biases the topic relatedness. Take the post "Do you like MAC?" as an example. At the context level, it relates to the topic "personal preference".

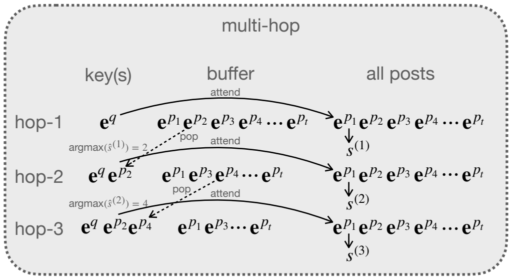

multi-hop
key(s)
buffer
all posts
attend
hop-1
$\mathbf{e}^q$
$\mathbf{e}^{p_1}\,\mathbf{e}^{p_2}\,\mathbf{e}^{p_3}\,\mathbf{e}^{p_4}\cdots\mathbf{e}^{p_t}$
$\mathbf{e}^{p_1}\,\mathbf{e}^{p_2}\,\mathbf{e}^{p_3}\,\mathbf{e}^{p_4}\cdots\mathbf{e}^{p_t}$
$s^{(1)}$
pop
argmax$(\hat{s}^{(1)}) = 2$
attend
hop-2
$\mathbf{e}^q\,\mathbf{e}^{p_2}$
$\mathbf{e}^{p_1}\,\mathbf{e}^{p_3}\,\mathbf{e}^{p_4}\cdots\mathbf{e}^{p_t}$
$\mathbf{e}^{p_1}\,\mathbf{e}^{p_2}\,\mathbf{e}^{p_3}\,\mathbf{e}^{p_4}\cdots\mathbf{e}^{p_t}$
$s^{(2)}$
pop
argmax$(\hat{s}^{(2)}) = 4$
attend
hop-3
$\mathbf{e}^q\,\mathbf{e}^{p_2}\mathbf{e}^{p_4}$
$\mathbf{e}^{p_1}\,\mathbf{e}^{p_3}\cdots\mathbf{e}^{p_t}$
$\mathbf{e}^{p_1}\,\mathbf{e}^{p_2}\,\mathbf{e}^{p_3}\,\mathbf{e}^{p_4}\cdots\mathbf{e}^{p_t}$
$s^{(3)}$

Figure 2: Multi-hop method, we illustrate hop-3.

At the word level, it relates to the topic "MAC". Meanwhile, the word "MAC" is ambiguous. For the user in Table 1, it has a great chance to be referred to as Apple's "Macintosh" as we can find a "MAC"-related post "PC or MAC for college students" in the dialogue history. But for a beauty blogger, it might refer to cosmetics "MAC".

In view of the first assumption, we decompose the relevance vector $\mathbf{s}$ into the word-level relevance vector $\mathbf{s}_1$ and the context-level relevance vector $\mathbf{s}_2$.

At word level, for t historical posts with a maximum length of L, we obtain the word-level matching matrix $\mathbf{M}_w \in \mathbb{R}^{(t+1)\times L\times L}$ by:

$$\mathbf{M}_w = \mathbf{W}_3^\top \tanh\left(\tilde{\mathbf{E}}^P \cdot \mathbf{W}_2 \cdot \tilde{\mathbf{e}}^{q\top}\right), \tag{11}$$

where $\mathbf{W}_2 \in \mathbb{R}^{d\times d\times(t+1)}$ and $\mathbf{W}_3 \in \mathbb{R}^{(t+1)\times 1}$ are parameters. $\tilde{\mathbf{E}}^P = \{\tilde{\mathbf{e}}^{p_1}, \cdots, \tilde{\mathbf{e}}^{p_t}, \tilde{\mathbf{e}}^q\}$ is the contextual representations for the current post and historical posts, which is obtained via a single attentive module: $\tilde{\mathbf{e}}^{(\cdot)} = f_{\text{att}}(\mathbf{e}^{(\cdot)}, \mathbf{e}^{(\cdot)}, \mathbf{e}^{(\cdot)})$ and $\mathbf{e}^{(\cdot)}$ is initialized by looking up a word embedding table. We then conduct max-pooling over word-level matching matrix to obtain the most important matching features, which are then linearly mapped into the word-level relevance vector $\mathbf{s}_1$ using the softmax function:

$$\mathbf{s}_1 = \text{softmax}\left(\text{MLP}\left(\left[\max_{dim=2} \mathbf{M}_w; \max_{dim=3} \mathbf{M}_w\right]\right)\right), \tag{12}$$

where $[;]$ is the concatenation operation.

At context level, we obtain context-level relevance vector $\mathbf{s}_2$ by:

$$\mathbf{s}_2 = \frac{\mathbf{U}^P \cdot \mathbf{u}^q}{\left\|\mathbf{U}^P\right\|_2 \left\|\mathbf{u}^q\right\|_2}, \tag{13}$$

where $\mathbf{u}^q = \underset{dim=1}{\text{mean}}\, \tilde{\mathbf{e}}^q$ and $\mathbf{U}^P = \underset{dim=2}{\text{mean}}\, \tilde{\mathbf{E}}^P$ are sentence representation obtained by mean pooling over the word dimension. We combine word-level and context-level relevance vector by:

$$\mathbf{s} = \alpha \cdot \mathbf{s}_1 + (1-\alpha)\cdot \mathbf{s}_2, \tag{14}$$

where α is a trainable parameter and is initialized by 0.5.

For now, the relevance vector $\mathbf{s}$ is obtained by using the current query q as a key to attend to historical posts. To tackle the bias discussed in the second assumption, we then design a multi-hop method that alleviates the word ambiguity by stretching the semantic richness of the current query. Figure 2 demonstrates the multi-hop method. Specifically, we store the historical posts in a buffer. At each hop, the most related historical post p_j will be popped out as a new key, where $j = \arg\max(\hat{s})$. $\hat{s}$ is the relevance scores of posts in the buffer. For the post "Do you like MAC?" in Table 1, at hop-1, the key is itself. After attending to historical posts, we expect the historical post "PC or MAC for college students" can be popped out as a new key. In this way, the word "MAC" is linked to "PC" and the ambiguity of the word "MAC" can be alleviated.

We denote all popped keys as $\mathbf{E}_s^P$. It is a subset of all historical posts $\mathbf{E}^P = \{\mathbf{e}^{p_1}, \cdots, \mathbf{e}^{p_t}\}$. At hop-1, $\mathbf{E}_s^P = \phi$ and the relevance score is computed by Eq. (14) with the current query $\mathbf{e}^q$. Afterwards, we update the current query $\mathbf{e}^q$ by:

$$\mathbf{e}^q = \underset{dim=2}{\text{mean}}\, f_{\text{stack}}(\mathbf{e}^q \cup \mathbf{E}_s^P). \tag{15}$$

We then obtain a new relevance score $\mathbf{s}$ via Eq. 14 using the updated representation $\mathbf{e}^q$. We denote the relevance score of hop-i as $\mathbf{s}^i$. After k hops, we have $\mathrm{S} = \{\mathbf{s}^1, \cdots, \mathbf{s}^k\}$. We then linearly map these scores into the final reweighting scores: $\bar{\mathbf{s}} = \mathrm{S}\cdot\beta$ where $\beta \in \mathbb{R}^{k\times 1}$ and $\bar{\mathbf{s}} \in \mathbb{R}^{t+1}$. Thus, we can rewrite the Eq. (10) to:

$$\mathbf{D} = \bar{\mathbf{s}} \cdot \mathbf{H}. \tag{16}$$

To thoroughly measure the relevance between the response candidate r and the post-aware user profile $\mathbf{D}$, we construct three matching matrices:

$$\mathbf{M}_1^D = \left[\frac{\mathbf{e}^r \mathbf{A}_1 \mathbf{D}^\top}{\sqrt{d}}; \frac{\mathbf{e}^r \cdot \mathbf{D}^\top}{\|\mathbf{e}^r\|_2 \|\mathbf{D}\|_2}\right], \tag{17}$$

$$\mathbf{M}_2^D = \left[\frac{\tilde{\mathbf{e}}^r \mathbf{A}_2 \tilde{\mathbf{D}}^\top}{\sqrt{d}}; \frac{\tilde{\mathbf{e}}^r \cdot \tilde{\mathbf{D}}^\top}{\|\tilde{\mathbf{e}}^r\|_2 \|\tilde{\mathbf{D}}\|_2}\right], \tag{18}$$

$$\mathbf{M}_3^D = \left[\frac{\hat{\mathbf{e}}^r \mathbf{A}_3 \hat{\mathbf{D}}^\top}{\sqrt{d}}; \frac{\hat{\mathbf{e}}^r \cdot \hat{\mathbf{D}}^\top}{\|\hat{\mathbf{e}}^r\|_2 \|\hat{\mathbf{D}}\|_2}\right], \tag{19}$$

where $\mathbf{A}_* \in \mathbb{R}^{d\times d}$ and d is the embedding size. $\tilde{\mathbf{e}}^r$ and $\tilde{\mathbf{D}}$ are the contextual embeddings obtained via a single-layer attentive module:

$$\tilde{\mathbf{e}}^r = f_{\text{att}}(\mathbf{e}^r, \mathbf{e}^r, \mathbf{e}^r), \quad \tilde{\mathbf{D}} = f_{\text{att}}(\mathbf{D}, \mathbf{D}, \mathbf{D}). \tag{20}$$

Meanwhile, $\hat{\mathbf{e}}^r$ and $\hat{\mathbf{D}}$ are obtained via cross-attention:

$$\hat{\mathbf{e}}^r = f_{\text{att}}(\tilde{\mathbf{e}}^r, \tilde{\mathbf{D}}, \tilde{\mathbf{D}}), \quad \hat{\mathbf{D}} = f_{\text{att}}(\tilde{\mathbf{D}}, \tilde{\mathbf{e}}^r, \tilde{\mathbf{e}}^r). \tag{21}$$

Thereafter, the three matching matrices are concatenated together:

$$\mathbf{M}_p = [\mathbf{M}_1^D; \mathbf{M}_2^D; \mathbf{M}_3^D]. \tag{22}$$

Same to the Personalized Style Matching, we use 2D CNN with max-pooling to extract high-level matching features. As the dialogue history is sorted by time, we utilize a single-layer GRU to capture the temporal signal of post-response pairs in the dialogue history.We use the GRU's final state as the Post-Aware Personalized Preference Matching feature $g^{\mathbf{P}}(q, r, H)$.

3.6 Module Fusion

In Section 3.4 and Section 3.5, we obtain two matching features: (1) personalized style matching feature $g^{\mathbf{S}}(q, r, H)$, which measures the personalized style consistency of the response candidates; and (2) post-aware personalized preference matching feature $g^{\mathbf{P}}(q, r, H)$, which measures the relevance of a response candidate and the user's personalized preferences. We concatenate the two matching features to get the final matching vector. We then use an MLP with a sigmoid activation function to compute the final matching score:

$$\mathcal{G} = g\left(q, r, \mathcal{U}\right) = \sigma\left(\text{MLP}\left(\left[g^{\mathbf{S}}\left(q, r, H\right); g^{\mathbf{P}}\left(q, r, H\right)\right]\right)\right). \tag{23}$$

We use cross-entropy loss to train the model:

$$\mathcal{L}(\theta) = -\frac{1}{|D|}\sum_{D}[y\log(\mathcal{G}) + (1-y)\log(1-\mathcal{G})]. \tag{24}$$

4 EXPERIMENTS

4.1 Dataset and Evaluation

4.1.1 Datasets. There are many datasets to evaluate retrieval-based dialogue models [19, 43, 48, 51]. However, none of them contains user identifications. In this paper, we expect to learn implicit user profiles from the user's dialogue history. Thus, we need datasets that contain users' identifications. We use two public datasets crawled from two social networking sites: Weibo and Reddit. The Weibo dataset is a subset of the PChatbotW dataset released by [30]. The PChatbotW dataset contains one-year Weibo logs from Sept. 10, 2018. Weibo is a popular social network in China. A Weibo user can post short messages publicly visible, and other users can make responses to the post. In the dataset, all posts and responses have user IDs. The Reddit dataset is extracted from the online forum Reddit from Dec. 1, 2015 to Oct. 30, 2018 [49]. In the dataset, the discussions can be expanded as tree-structured reply chains where each parent node can be considered as a post to its child nodes. Thus, we generate post-responses pairs by traversing the tree structure.

4.1.2 Dataset Construction. After aggregating users' dialogue history, we filter users with less than fifteen history dialogues to guarantee enough personalized information. Besides, we limit the number of words to 50 for each utterance in the user's dialogue history. For each user, we sort the dialogue history by time and use the latest post as the current query. Following previous works on constructing retrieval-based dialogue datasets [19, 43, 48, 51], we create a list of ten response candidates for the current query.

The response candidates can be divided into three groups: (1) we use the user's response under the current query as the ground-truth (personalized response); (2) we select other users' responses under the current query (non-personalized response) as part of the response candidates; (3) following [43], we retrieve response candidates via a retrieval engine (relevant response) and filter out responses issued by the current user as part of the response candidates. Notably, both personalized responses and non-personalized responses can be considered as *proper responses* to the current query.

Compared to previous candidates sampling strategies, such as random sampling [19, 48, 51] or only retrieval [43], our response sampling method is more advantageous, especially regarding personalized chatbot, because: First, in a practical scenario, the retrieval-based chatbots usually retrieve response candidates relevant to the query via a retrieval engine and then select a proper response from the candidates. Thus, instead of justifying the differences between irrelevant candidates and proper responses, learning to select proper responses from a list of relevant responses is more useful and challenging. Furthermore, we expect that the dialogue chatbot has a consistent personality. Thus, it should also be able to recognize the personalized response from a list of proper responses, including personalized responses and non-personalized ones. The statistical information of the two datasets is shown in Table 2.

Table 2: The statistics of the two datasets.

	Weibo	Reddit
Number of users	420,000	280,642
Avg. history length	32.3	85.4
Avg. length of post	24.9	10.5
Avg. length of response	10.1	12.4
Avg. number of non-pers. candidates	3.9	3.2
Avg. number of relevant candidates	5.1	5.8
Number of response candidates	10	10
Number of training samples	3,000,000	2,000,000
Number of validation samples	600,000	403,210
Number of testing samples	600,000	403,210

To train baseline models that require external explicit user profile such as DIM [11] and RSM-DCK [13], following [22, 53], we also collect explicit persona sentences using heuristic methods. Note that these explicit persona sentences are not used in our model.

4.1.3 Evaluation Metrics. As discussed in Section 4.1.2, the negative response candidates are sampled from the retrieval engine and from other users' responses under the same post (non-personalized response). Thus, we consider three types of metrics to evaluate the models' performances. First, we use $\mathbf{R_n@k}$ (recall at position k in n candidates) and **MRR** (Mean Reciprocal Rank) to measure the model's ability to select a personalized response from all candidates. Second, we use **nDCG** (normalized Discounted Cumulative Gain) to measure the model's ability to select a proper response from all candidates. Last, we introduce a new metric $\mathbf{R_p@k}$ to measure the model's ability to select a personalized response from all proper candidates. We introduce the **nDCG** and $\mathbf{R_p@k}$ in the following:

nDCG: the metric assigns scores to both personalized and non-personalized responses. The metric matters in the situation that no personalized response in the list of candidates. In such a scenario, a non-personalized response is better than an irrelevant response. We use **nDCG@5** and set the relevance score as 2 and 1 for the personalized and non-personalized responses, respectively.

$\mathbf{R_p@k}$: for a personalized chatbot, personalized responses are more useful than non-personalized responses. Thus, we introduce recall at position k in the **p**roper candidates to measure the model's ability to distinguish the personalized responses from other proper candidates. Taking $\mathbf{R_p@1}$ as an example, when the personalized response ranks 1st out of all proper responses, we assign 1 score to the sample, otherwise 0.

4.2 Baseline Model

In our task, user's dialogue history contains many single-turn dialogues. Given a query, we can either consider the task as single-turn matching or use the dialogue history as a context. Thus, we consider three types of retrieval-based model as baseline models:

(1) Single-turn matching models:

ARC-I [12]: The model uses a Siamese architecture in which multi-layer 1D CNNs capture multi-grained semantic features of each sentence and then perform matching. **ARC-II** [12]: The model directly builds on the interaction space between two sentences. It uses 1D CNNs to capture low-level features and then uses 2D CNNs to capture deep matching features. **KNRM** [45]: The model uses a kernel-based neural ranking model to model word-level soft

Table 3: Evaluation results of all models on both Weibo and Reddit corpus. "†" and "‡" denote the result is significantly worse than our method in t-test with $p < 0.01$ and $p < 0.05$ level respectively. Models with "◇" are implemented with the provided source code, and models with "♡" are implemented by ourselves. The best results are in bold.

	Weibo Corpus						Reddit Corpus					
	R_{10}@1	**R_{10}@2**	**R_{10}@5**	**MRR**	**nDCG**	**R_p@1**	**R_{10}@1**	**R_{10}@2**	**R_{10}@5**	**MRR**	**nDCG**	**R_p@1**
(1) ARC-I$^\diamond$	$0.173^\dagger$	$0.316^\dagger$	$0.643^\dagger$	$0.378^\dagger$	$0.600^\dagger$	$0.243^\dagger$	$0.459^\dagger$	$0.597^\dagger$	$0.848^\dagger$	$0.617^\dagger$	$0.736^\dagger$	$0.529^\dagger$
(1) ARC-II$^\diamond$	$0.178^\dagger$	$0.318^\dagger$	$0.655^\dagger$	$0.383^\dagger$	$0.622^\dagger$	$0.242^\dagger$	$0.450^\dagger$	$0.594^\dagger$	$0.854^\dagger$	$0.612^\dagger$	$0.731^\dagger$	$0.514^\dagger$
(1) KNRM$^\heartsuit$	$0.271^\dagger$	$0.476^\dagger$	$0.899^\dagger$	$0.502^\dagger$	$0.856^\dagger$	$0.277^\dagger$	$0.558^\dagger$	$0.721^\dagger$	$0.905^\dagger$	$0.698^\dagger$	$0.847^\dagger$	$0.573^\dagger$
(1) Conv-KNRM$^\heartsuit$	$0.323^\dagger$	$0.520^\dagger$	$0.893^\dagger$	$0.538^\dagger$	$0.818^\dagger$	$0.334^\dagger$	$0.576^\dagger$	$0.711^\dagger$	$0.917^\dagger$	$0.712^\dagger$	$0.797^\dagger$	$0.616^\dagger$
(2) SMN$^\diamond$	$0.328^\dagger$	$0.525^\dagger$	$0.894^\dagger$	$0.541^\dagger$	$0.837^\dagger$	$0.342^\dagger$	$0.433^\dagger$	$0.617^\dagger$	$0.909^\dagger$	$0.618^\dagger$	$0.785^\dagger$	$0.475^\dagger$
(2) DAM$^\diamond$	$0.438^\dagger$	$0.644^\dagger$	0.966	$0.635^\dagger$	0.881	$0.442^\dagger$	$0.605^\dagger$	$0.748^\dagger$	$0.965^\dagger$	$0.741^\dagger$	$0.830^\dagger$	$0.626^\dagger$
(2) IOI$^\diamond$	$0.442^\dagger$	$0.651^\dagger$	**0.969**	$0.639^\dagger$	**0.890**	$0.446^\dagger$	$0.620^\dagger$	$0.764^\dagger$	$0.974^\dagger$	$0.753^\dagger$	$0.857^\dagger$	$0.636^\dagger$
(2) MSN$^\diamond$	$0.355^\dagger$	$0.554^\dagger$	$0.931^\dagger$	$0.567^\dagger$	0.878	$0.359^\dagger$	$0.555^\dagger$	$0.733^\dagger$	$0.977^\dagger$	$0.715^\dagger$	$0.875^\dagger$	$0.567^\dagger$
(3) DIM$^\diamond$	$0.388^\dagger$	$0.557^\dagger$	$0.835^\dagger$	$0.571^\dagger$	$0.621^\dagger$	$0.460^\ddagger$	$0.678^\dagger$	$0.813^\dagger$	$0.979^\ddagger$	$0.794^\dagger$	$0.804^\dagger$	**0.724**
(3) DGMN$^\diamond$	$0.358^\dagger$	$0.528^\dagger$	$0.818^\dagger$	$0.547^\dagger$	$0.636^\dagger$	$0.424^\dagger$	$0.539^\dagger$	$0.703^\dagger$	$0.956^\dagger$	$0.697^\dagger$	$0.838^\dagger$	$0.563^\dagger$
(3) RSM-DCK$^\diamond$	$0.428^\dagger$	$0.627^\dagger$	$0.947^\dagger$	$0.623^\dagger$	$0.858^\dagger$	$0.438^\dagger$	$0.615^\dagger$	$0.753^\dagger$	$0.972^\dagger$	$0.748^\dagger$	$0.838^\dagger$	$0.633^\dagger$
(3) CSN$^\diamond$	$0.387^\dagger$	$0.560^\dagger$	$0.842^\dagger$	$0.572^\dagger$	$0.654^\dagger$	$0.445^\dagger$	$0.681^\dagger$	$0.807^\dagger$	$0.976^\dagger$	$0.794^\dagger$	$0.846^\dagger$	$0.698^\ddagger$
(3) IMPChat	**0.460**	**0.665**	0.963	**0.651**	0.868	**0.466**	**0.691**	**0.820**	**0.982**	**0.804**	**0.877**	0.706

matches for single turn-dialogue. **Conv-KNRM** [6]: The model uses a CNN kernel-based ranking model to model n-gram soft matches for single-turn dialogue.

(2) Multi-turn dialogue models:

SMN [43]: The model matches a response with each utterance in the context on multi-level. It uses CNN to extract matching information and obtains the final matching score using an RNN. **DAM** [56]: The model constructs multi-level text segment representations with stacked self-attention and then extracts the matched segment pairs with attention across the context and response. **IOI** [38]: The model uses a chain of interactive blocks to conduct semantic interaction between response and utterance in the context many times to obtain deep interactive matching information. **MSN** [46]: The model conducts context selection and filtered irrelevant context. It lets the response candidates interact with each utterance in the context to get multiple matching features.

(3) Persona-based dialogue models:

DIM [11]: The model uses BiLSTM to encode response candidates, user profile, and context, and lets the user profile, context interact with response candidates, respectively, via cross-matching. It utilizes another BiLSTM to aggregate the matching features. **DGMN** [52]: The model fuses information in a document and a context into representations of each other. It performs hierarchical interactions between a response and both document and context. And the importance of each part is dynamically determined. **RSM-DCK** [13]: The model pre-selects document and context. It performs matching between response-context and response-document. It uses Bi-LSTM to aggregate matching features. **CSN** [61]: The model designs a content selection network to explicitly select relevant contents, and filter out the irrelevant parts. It performs context-response matching and document-response matching and aggregates matching features using CNN with LSTM.

4.3 Implementation Details

In our model, we use Word2Vec [24] to initialize the word embedding, which has a size of 200. The Word2Vec is trained on the dataset. In the experiments, all baseline models apply the word embedding. We limit the max sequence length to 50. For the Personalized Style Matching module, we set the number of attentive modules to 3. For the Post-Aware Personalized Preference Matching module, we set the number of hops to 2 and set the hidden size of GRU to 300. We use three-layer CNN with $16[3,3]$, $32[3,3]$, $64[3,3]$ filters, respectively. The 1st and 2nd layer use $[2,2]$ stride and the max-pooling size is $[2,2]$ with $[2,2]$ stride. The 3rd layer uses $[3,3]$ stride, and the max-pooling size is $[3,3]$ with $[3,3]$ stride. We optimize the model using the Adam method with a learning rate set as 5e-4, which is decayed during training. We tune IMPChat and all baseline models on the validation set and evaluate on the test set. For both datasets, we set batch size as 128 and train the model for 10 epochs on two Tesla V100 16G GPUs. Our model is implemented by Pytorch [28], and the code will be released based upon the acceptance of the paper.

4.4 Experimental Results

Table 3 shows the results. Compared to previous state-of-the-art models, we find that **our model IMPChat achieves the best performance across most of the metrics on the two datasets**. As mentioned in Section 4.1.3, we use three types of evaluation metrics that reflect the model's abilities from different aspects. **First**, regarding R_{10}@1, R_{10}@2 and MRR, our model IMPChat leads to statistically significant improvement than all baseline models on the two datasets (t-test with $p < 0.01$). It demonstrates the strong capability of our IMPChat to distinguish the marginal differences between personalized responses and all other responses. **Second**, regarding nDCG, IMPChat outperforms all baselines on Reddit Corpus while IMPChat is worse than several multi-turn models on the Weibo dataset. We also find that most other personalized

Table 4: Ablation results on the Weibo dataset.

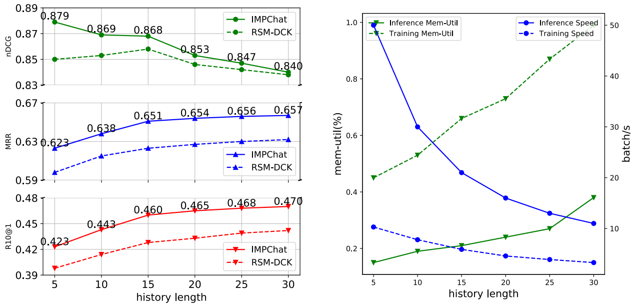
	R_{10}@1	R_{10}@2	R_{10}@5	MRR	nDCG	R_p@1
IMPChat	**0.460**	**0.665**	0.963	**0.651**	0.868	**0.466**
w/o mutil-hop	0.451	0.662	**0.966**	0.646	**0.878**	0.455
w/o style	0.443	0.654	0.962	0.639	0.871	0.447
w/o preference	0.383	0.572	0.939	0.585	0.867	0.391

models perform worse than multi-turn models in terms of nDCG. The potential reasons might be: (1) personalized information increases the marginal distance between personalized responses and other responses. Therefore, the marginal distance between non-personalized (but relevant) responses and other negative responses decreases; (2) more personalized information also brings more noise that undermines matching (also discussed in section 4.5.2). **Last**, regarding R_p@1, we find that most personalized models perform better than multi-turn models, reflecting that personalized information enhances the models' capability to distinguish personalized responses from non-personalized responses. Our IMPChat outperforms all baseline models on Weibo Corpus and baseline models excluding DIM on Reddit Corpus, showing the effectiveness of building implicit user profiles from the dialogue history.

4.5 Discussion

4.5.1 Ablation Study. We respectively remove (1) the multi-hop method, (2) the Personalized Style Matching module, and (3) the Post-Aware Personalized Preference Matching module of IMPChat to investigate their effectiveness. Table 4 shows the results. We can conclude that removing any part of IMPChat will lead to a performance drop. Notably, performance degrades a large margin when removing the Post-Aware Personalized Preference Matching module. This is because, without the module, IMPChat only accesses historical responses so that the conditional matching signals among post-response pairs diminish. Besides, we find that without either multi-hop or Personalized Style Matching module, IMPChat still outperforms all baseline models regarding R_{10}@1, R_{10}@2, MRR and R_p@1 which verifies that either of the two parts indeed brings orthogonal information that the rest parts fail to capture.

4.5.2 How to Choose the History Length? We aim to learn an implicit user profile from the user's dialogue history. Thus, how to choose the length of dialogue history is a crucial problem. We investigate the problem from two aspects: how the history dialogue length affects the model's performance; and the difficulty to train the model or make inferences with more dialogue history. Figure 3 shows the influence of the history length, from which we can find: (1) as aforementioned, R_{10}@1 and MRR measure the model's capability to retrieve a personalized response, while nDCG also considers non-personalized responses. Thus, with more dialogue history, the model's capability of finding the personalized response steadily increases since the longer history brings more personalized information. However, the model's capability to retrieve non-personalized responses decreases. The potential reason for the decline is that long history brings more noise into the implicit profile, which ambiguates the decision boundaries between non-personalized responses and other negative ones; (2) Figure 3b shows that the required computing resources increase with longer dialogue history[5]. In a practical scenario, there is a trade-off regarding how to choose the length of dialogue history. First, training and inference speeds matter in practical use, especially for devices with limited computing resources such as a mobile phone. Second, not always personalized responses can be retrieved before response selection. When no response matches the user's personality, a relevant response is better than a wrong one. We also conduct experiments on RSM-DCK, which has a similar number of parameters to IMPChat. The results also verify our findings.



(a) Effect of history length. **(b) Resource usage and speed.**

Figure 3: The left figure shows the performance variance on different history lengths. The right figure shows the required GPU memory and speed for training and inference.

5 CONCLUSION

In this paper, we propose building implicit user profiles to endow personality to a retrieval-based chatbot. To achieve such a goal, we build a model IMPChat to learn the implicit user profile from two aspects. First, it models the user's personalized language style using the user's historical responses. It then models the user's personalized preferences from a post-aware user profile which contains post-response pairs that are topically related to the current post. Extensive experimental results on two large datasets show that our method outperforms all previous state-of-the-art models, verifying our model's effectiveness for the personalized retrieval-based chatbot.

ACKNOWLEDGEMENT

Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 61872370 and No. 61832017, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, Shandong Provincial Natural Science Foundation under Grant ZR2019ZD06, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. I also wish to acknowledge the support provided and contribution made by Public Policy and Decision-making Research Lab of Renmin University of China.

[5] We set the batch size to 32, and conduct experiments on a single Tesla V100 16G GPU.

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] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[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] Ana Paula Chaves and Marco Aurelio Gerosa. 2020. How Should My Chatbot Interact? A Survey on Social Characteristics in Human–Chatbot Interaction Design. *International Journal of Human–Computer Interaction* (Nov 2020), 1–30. https://doi.org/10.1080/10447318.2020.1841438

[6] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-Hoc Search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 126–134. https://doi.org/10.1145/3159652.3159659

[7] Antonella De Angeli, Graham I Johnson, and Lynne Coventry. 2001. The unfriendly user: exploring social reactions to chatterbots. In *Proceedings of The International Conference on Affective Human Factors Design, London*. 467–474.

[8] Jianfeng Gao, Michel Galley, and Lihong Li. 2019. *Neural Approaches to Conversational AI: Question Answering, Task-oriented Dialogues and Social Chatbots*. Now Foundations and Trends.

[9] James H. Gerlach and Feng-Yang Kuo. 1991. Understanding Human-Computer Interaction for Information Systems Design. *MIS Q.* 15, 4 (1991), 527–549.

[10] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[11] Jia-Chen Gu, Zhen-Hua Ling, Xiaodan Zhu, and Quan Liu. 2019. Dually Interactive Matching Network for Personalized Response Selection in Retrieval-Based Chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 1845–1854.

[12] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*. 2042–2050.

[13] Kai Hua, Zhiyuan Feng, Chongyang Tao, Rui Yan, and Lu Zhang. 2020. Learning to Detect Relevant Contexts and Knowledge for Response Selection in Retrieval-Based Dialogue Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 525–534. https://doi.org/10.1145/3340531.3411967

[14] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, , et al. 2017. *AliMe Assist* : An Intelligent Assistant for Creating an Innovative E-commerce Experience. In *Proceedings of the CIKM 2017*. ACM, 2495–2498.

[15] 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.

[16] 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.

[17] 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.

[18] Yafei Liu, Hongjin Qian, Hengpeng Xu, and Jinmao Wei. 2020. Speaker or Listener? The Role of a Dialog Agent. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 4861–4869. https://doi.org/10.18653/v1/2020.findings-emnlp.437

[19] Ryan Lowe, Nissan Pow, Iulian V. Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A large dataset for research in unstructured multi-turn Dialogue systems. *SIGDIAL 2015 - 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Proceedings of the Conference* (2015), 285–294. https://doi.org/10.18653/v1/w15-4640 arXiv:1506.08909

[20] 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 *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 555–564. https://doi.org/10.1145/3404835.3462828

[21] François Mairesse and Marilyn A. Walker. 2007. PERSONAGE: Personality Generation for Dialogue. In *Proceedings of the ACL 2007*. ACL.

[22] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training Millions of Personalized Dialogue Agents. In *EMNLP*. 2775–2779.

[23] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training Millions of Personalized Dialogue Agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2775–2779. https://doi.org/10.18653/v1/D18-1298

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs.CL]

[25] 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.

[26] Kellie Morrissey and Jurek Kirakowski. 2013. 'Realness' in chatbots: establishing quantifiable criteria. In *International conference on human-computer interaction*. Springer, 87–96.

[27] Alice H. Oh and Alexander I. Rudnicky. 2000. Stochastic Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational Systems - Volume 3*. ACL, 27–32.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[29] Manuel Portela and Carlos Granell-Canut. 2017. A New Friend in Our Smartphone? Observing Interactions with Chatbots in the Search of Emotional Engagement. In *Proceedings of the XVIII International Conference on Human Computer Interaction* (Cancun, Mexico) *(Interacción '17)*. Association for Computing Machinery, New York, NY, USA, Article 48, 7 pages. https://doi.org/10.1145/3123818.3123826

[30] 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 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, Virtual Event, Canada. https://doi.org/10.1145/3404835.3463239

[31] Qiao Qian, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Assigning Personality/Profile to a Chatting Machine for Coherent Conversation Generation.. In *IJCAI*. 4279–4285.

[32] 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.

[33] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proceedings of the AAAI 2016*.

[34] Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short-Text Conversation. In *Proceedings of the ACL 2015*. ACL, 1577–1586.

[35] Heung-Yeung Shum, Xiao-dong He, and Di Li. 2018. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering* 19, 1 (2018), 10–26.

[36] 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.

[37] Haoyu Song, Wei-Nan Zhang, Jingwen Hu, and Ting Liu. 2020. Generating Persona Consistent Dialogues by Exploiting Natural Language Inference. In *Proceedings of the AAAI 2020, IAAI 2020, EAAI 2020*. AAAI Press, 8878–8885.

[38] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. One Time of Interaction May Not Be Enough: Go Deep with an Interaction-over-Interaction Network for Response Selection in Dialogues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1–11. https://doi.org/10.18653/v1/P19-1001

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, et al. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[40] Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *CoRR* abs/1506.05869 (2015). arXiv:1506.05869

[41] Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A Dataset for Research on Short-Text Conversations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, 935–945. https://www.aclweb.org/anthology/D13-1096

[42] Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2015. Syntax-Based Deep Matching of Short Texts. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[43] Yu Wu, Wei Wu, Chen Xing, Zhoujun Li, and Ming Zhou. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1 (2017), 496–505. https://doi.org/10.18653/v1/P17-1046 arXiv:1612.01627

[44] 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.

[45] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Aug 2017). https://doi.org/10.1145/3077136.3080809

[46] Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop Selector Network for Multi-turn Response Selection in Retrieval-based Chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 111–120. https://doi.org/10.18653/v1/D19-1011

[47] Saizheng 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.

[48] Saizheng 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? *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1 (2018), 2204–2213. https://doi.org/10.18653/v1/p18-1205 arXiv:1801.07243

[49] 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.

[50] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling Multi-turn Conversation with Deep Utterance Aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*. 3740–3752.

[51] Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling Multi-turn Conversation with Deep Utterance Aggregation. 61733011 (2018). arXiv:1806.09102 http://arxiv.org/abs/1806.09102

[52] Xueliang Zhao, Chongyang Tao, Wei Wu, Can Xu, Dongyan Zhao, and Rui Yan. 2019. A Document-grounded Matching Network for Response Selection in Retrieval-based Chatbots. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 5443–5449.

[53] Peixiang Zhong, Chen Zhang, Hao Wang, Yong Liu, and Chunyan Miao. 2020. Towards Persona-Based Empathetic Conversational Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6556–6566.

[54] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics* 46, 1 (2020), 53–93.

[55] Xiangyang Zhou, Daxiang Dong, Hua Wu, Shiqi Zhao, Dianhai Yu, Hao Tian, Xuan Liu, and Rui Yan. 2016. Multi-view Response Selection for Human-Computer Conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 372–381. https://doi.org/10.18653/v1/D16-1036

[56] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-Turn Response Selection for Chatbots with Deep Attention Matching Network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1118–1127. https://doi.org/10.18653/v1/P18-1103

[57] Yujia Zhou, Zhicheng Dou, Bingzheng Wei, Ruobing Xie, and Ji-Rong Wen. 2021. Group based Personalized Search by Integrating Search Behaviour and Friend Network. In *SIGIR 2021*. ACM, 92–101.

[58] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Encoding History with Context-aware Representation Learning for Personalized Search. In *SIGIR*. ACM, 1111–1120.

[59] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Enhancing Re-finding Behavior with External Memories for Personalized Search. In *WSDM 2020*.

[60] 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

[61] Yutao Zhu, Jian-Yun Nie, Kun Zhou, Pan Du, and Zhicheng Dou. 2021. Content Selection Network for Document-Grounded Retrieval-Based Chatbots. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12656)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer, 755–769. https://doi.org/10.1007/978-3-030-72113-8_50

[62] Yutao Zhu, Jian-Yun Nie, Kun Zhou, Pan Du, Hao Jiang, and Zhicheng Dou. 2021. Proactive Retrieval-based Chatbots based on Relevant Knowledge and Goals. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2000–2004. https://doi.org/10.1145/3404835.3463011

[63] Yutao Zhu, Ruihua Song, Zhicheng Dou, Jian-Yun Nie, and Jin Zhou. 2020. ScriptWriter: Narrative-Guided Script Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 8647–8657. https://doi.org/10.18653/v1/2020.acl-main.765