Learning to Respond with Your Favorite Stickers: A Framework of Unifying Multi-Modality and User Preference in Multi-Turn Dialog

12

SHEN GAO and XIUYING CHEN, Wangxuan Institute of Computer Technology, Peking University
LI LIU, Inception Institute of Artificial Intelligence
DONGYAN ZHAO, Wangxuan Institute of Computer Technology, Peking University
RUI YAN, Gaoling School of Artificial Intelligence, Renmin University of China and Wangxuan Institute of Computer Technology, Peking University

Stickers with vivid and engaging expressions are becoming increasingly popular in online messaging apps, and some works are dedicated to automatically select sticker response by matching the stickers image with previous utterances. However, existing methods usually focus on measuring the matching degree between the dialog context and sticker image, which ignores the user preference of using stickers. Hence, in this article, we propose to recommend an appropriate sticker to user based on multi-turn dialog context and sticker using history of user. Two main challenges are confronted in this task. One is to model the sticker preference of user based on the previous sticker selection history. Another challenge is to jointly fuse the user preference and the matching between dialog context and candidate sticker into final prediction making. To tackle these challenges, we propose a *Preference Enhanced Sticker Response Selector* (PESRS) model. Specifically, PESRS first employs a convolutional-based sticker image encoder and a self-attention-based multi-turn dialog encoder to obtain the representation of stickers and utterances. Next, deep interaction network is proposed to conduct deep matching between the sticker and each utterance. Then, we model the user preference by using the recently selected stickers as input and use a key-value memory network to store the preference representation. PESRS then learns the short-term and long-term dependency between all interaction results by a fusion network and dynamically fuses the user preference representation into the final sticker selection prediction. Extensive experiments conducted on a large-scale real-world dialog dataset show that our model achieves the state-of-the-art performance for all commonly used metrics. Experiments also verify the effectiveness of each component of PESRS.

CCS Concepts: • **Information systems → Multimedia content creation**; **Retrieval models and ranking**;

Additional Key Words and Phrases: Sticker selection, user modeling, multi-turn dialog

This work was supported by the National Science Foundation of China (NSFC No. 61876196) and the National Key R&D Program of China (2020AAA0105200). Rui Yan is supported as a Young Fellow of Beijing Institute of Artificial Intelligence (BAAI).

Authors' addresses: S. Gao, X. Chen, and D. Zhao, Wangxuan Institute of Computer Technology, Peking University; emails: {shengao, xy-chen, zhaody}@pku.edu.cn; L. Liu, Inception Institute of Artificial Intelligence; email: li-liu1985@inceptioniai.org; R. Yan (corresponding author), Gaoling School of Artificial Intelligence, Renmin University of China and Wangxuan Institute of Computer Technology, Peking University; email: ruiyan@pku.edu.cn.

1046-8188/2021/02-ART12 $15.00
https://doi.org/10.1145/3429980

ACM Reference format:
Shen Gao, Xiuying Chen, Li Liu, Dongyan Zhao, and Rui Yan. 2021. Learning to Respond with Your Favorite Stickers: A Framework of Unifying Multi-Modality and User Preference in Multi-Turn Dialog. *ACM Trans. Inf. Syst.* 39, 2, Article 12 (February 2021), 32 pages.
https://doi.org/10.1145/3429980

1 INTRODUCTION

Images (a.k.a.,graphicons) are another important approach for expressing feelings and emotions in addition to using text in communication. In mobile messaging apps, these images can generally be classified into emojis and stickers. An emoji is a kind of small picture that is already stored in most of the keyboard of the mobile operational systems, i.e., iOS or Android. Emojis are pre-designed by the mobile phone vendor (now it is managed by standards organization) and the number of emoji is limited, and users cannot design emojis by themselves. Different from the inflexible emojis, a sticker is an image or graphicon essentially [14, 24, 32] that users can draw or modify images as a sticker and upload it to the chatting app by themselves. The using of stickers on online chatting usually brings diversity of expressing emotion. Emojis are sometimes used to help reinforce simple emotions in a text message due to their small size, and their variety is limited. Stickers, however, can be regarded as an alternative for text messages, which usually include cartoon characters and are of high definition. They can express much more complex and vivid emotion than emojis. Most messaging apps, such as WeChat, Telegram, WhatsApp, and Slack provide convenient ways for users to download stickers for free or even share self-designed ones. We show a chat window including stickers in Figure 1.

Stickers are becoming more and more popular in online chat. First, sending a sticker with a single click is much more convenient than typing text on the 26-letter keyboard of a small mobile phone screen. Second, there are many implicit or strong emotions that are difficult to express in words but can be captured by stickers with vivid facial expressions and body language. However, the large-scale use of stickers means that it is not always straightforward to think of the sticker that best expresses one's feeling according to the current chatting context. Users need to recall all the stickers they have collected and selected the appropriate one, which is both difficult and time-consuming.

Consequently, much research has focused on recommending appropriate emojis to users according to the chatting context. Existing works such as Reference [79] are mostly based on emoji recommendation, where they predict the probable emoji given the contextual information from multi-turn dialog systems. In contrast, other works [6, 7] recommend emojis based on the text and images posted by a user. As for sticker recommendation, existing works such as Reference [41] and apps like Hike or QQ directly match the text typed by the user to the short text tag assigned to each sticker. However, since there are lots of ways of expressing the same emotion, it is very hard to capture all variants of an utterance as tags.

To overcome the drawbacks, we proposed a sticker response selector (SRS) for sticker selection in our early work [22], where we addressed the task of sticker response selection in multi-turn dialog. We focus on the two main challenges in this work: (1) Since existing image recognition methods are mostly built with real-world images, how to capture the semantic meaning of sticker is challenging. (2) Understanding multi-turn dialog history information is crucial for sticker recommendation, and jointly modeling the candidate sticker with multi-turn dialog is challenging. Herein, we propose a novel sticker recommendation model, namely SRS, for sticker response selection in multi-turn dialog. Specifically, SRS first learns representations of dialog context history using a self-attention mechanism and learns the sticker representation by a

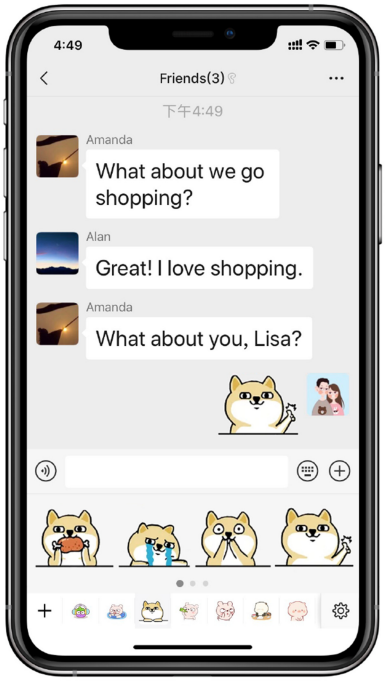


Fig. 1. An example of stickers in a multi-turn dialog. Sticker response selector automatically selects the proper sticker based on multi-turn dialog history.

convolutional neural network (CNN). Next, SRS conducts deep matching between the sticker and each utterance and produces the interaction results for every utterance. Finally, SRS employs a fusion network that consists of a sub-network fusion recurrent neural network (RNN) and fusion transformer to learn the short- and long-term dependency of the utterance interaction results. The final matching score is calculated by an interaction function. To evaluate the performance of our model, we propose a large number of multi-turn dialog datasets associated with stickers from one of the popular messaging apps. Extensive experiments conducted on this dataset show that SRS significantly outperforms the state-of-the-art baseline methods in commonly used metrics.

However, the user's sticker selection depends not only on the matching degree between dialog context and candidate sticker image but also on the user's preference of using sticker. When users decide to use a sticker as their response in multi-turn dialog, they may choose their favorite one from all appropriate stickers as the final response. We assume that a user tends to use the recently used sticker in their dialog history, and the recently used sticker can represent the user's preference of sticker selection. An example is shown in Figure 2. To verify this assumption, we retrieve 10 recently used stickers of each user and calculate the proportion of whether the currently used sticker appeared in these 10 stickers. The result shows that 54.09% of the stickers exist in the 10 recently used sticker set. Hence, we reach to the conclusion that users have strong personal preference when selecting the sticker as their response for the current dialog context. However, in some cases, this also indicates a tendency to re-use stickers but not necessarily a preference.

Motivated by this observation, in this work, we take one step further and improve our previously proposed SRS framework with user preference modeling. Overall, we propose a novel sticker recommendation model that considers the user preference, namely Preference Enhanced Sticker Response Selector (PESRS). Specifically, PESRS first employs a convolutional network to extract

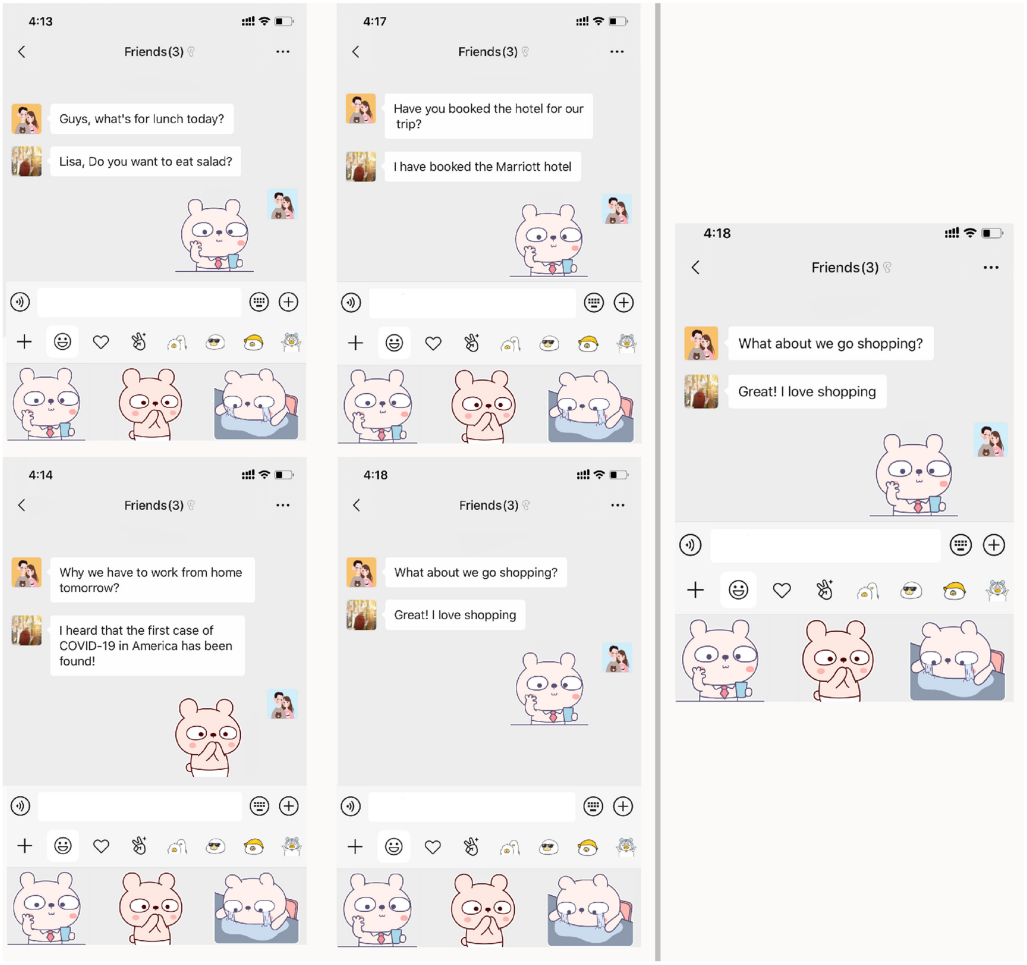


Fig. 2. User's history dialog context and the selected sticker. Four figures in the left history dialog context and the selected sticker, and the right one the current dialog context with the user-selected sticker. A user tends to use the same sticker when the dialog context is semantically similar.

features from the candidate stickers. Then, we retrieve the recent user sticker selections then a user preference modeling module is employed to obtain a user preference representation. Next, we conduct the deep matching between the candidate sticker and each utterance as the same as SRS. Finally, we use a gated fusion method to combine the deep matching result and user preference into final sticker prediction.

The key to the success of PESRS lies in how to design the user preference modeling module, which should not only identify the user's favorite sticker but also consider the current dialog context. Motivated by this, we first propose an RNN-based position-aware sticker modeling module that encodes the recently used stickers in chronological order. Then, we employ a key-value memory network to store these sticker representations as values and the corresponding dialog context as keys. Finally, we use the current dialog context to query the key-value memory and obtain the dynamic user preference of the current dialog context.

We empirically compare PESRS and SRS on the public dataset[1] proposed by our early work [22]. This is a large-scale real-world Chinese multi-turn dialog dataset, where dialog context is multiple text utterances and the response is a sticker image. Experimental results show that on this dataset, our newly proposed PESRS model can significantly outperform the existing methods. Particularly, PESRS yields 4.8% and 7.1% improvement in terms of *MAP* and R_{10}@1 compared with our early work SRS. In addition to the comprehensive evaluation, we also evaluate our proposed user preference memory by a fine-grained analysis. The analysis reveals how the model leverages the user's recent sticker selection history and provides us insights on why they can achieve big improvement over state-of-the-art methods.

This work is a substantial extension of our previous work reported at WWW 2020. The extension in this article includes the user preference modeling framework for the existing methods, a proposal of a new framework for sticker selection in the multi-turn dialog. Specifically, the contributions of this work include the following:

- We propose a position-aware sticker modeling module that can model the user's sticker selection history.
- We propose a key-value memory network to store the user's recently used stickers and its corresponding dialog context.
- Finally, we use the current dialog context to query the key-value memory and obtain a user preference representation and then fuse the user preference representation into final sticker prediction dynamically.
- Experiments conducted on a large-scale real-world dataset show that our model outperforms all baselines, including state-of-the-art models. Experiments also verify the effectiveness of each module in PESRS as well as its interpretability.

The rest of the article is organized as follows: We summarize related work in Section 2. Section 3 introduces the data collection method and some statistics of our proposed multi-turn dialog sticker selection dataset. We then formulate our research problem in Section 4 and elaborate our approach in Section 5. Section 6 gives the details of our experimental setup, and Section 7 presents the experimental results. Finally, Section 8 concludes the article.

2 RELATED WORK

We outline related work on sticker recommendation, user modeling, visual question answering, visual dialog, and multi-turn response selection.

2.1 Sticker and Emoji Recommendation

Most of the previous works emphasize the use of emojis instead of stickers. For example, References [6, 7] use a multimodal approach to recommend emojis based on the text and images in an Instagram post. Reference [27] proposes a MultiLabel-RandomForest algorithm to predict emojis based on the private instant messages. Reference [87] conducts emoji prediction on social media text (e.g., Sina Weibo and Twitter), and they tackle this task as ranking among all emojis. The total number of unique emojis in their dataset is 50, which is much smaller than the number of stickers. What is more, emojis are limited in variety, while there exists an abundance of different stickers. Reference [91] incorporates the emoji information into the dialog generation task, and they use the emoji classification as an auxiliary task to facilitate the dialog generation to produce utterance with proper emotion. The most similar work to ours is Reference [41], where they

[1] https://github.com/gsh199449/stickerchat.

generate recommended stickers by first predicting the next message the user is likely to send in the chat, and then substituting it with an appropriate sticker.

However, more often than not the implication of the stickers cannot be fully conveyed by text and, in this article, we focus on directly generating sticker recommendations from dialog history.

2.2 User Modeling

User modeling [35, 57, 85, 93, 94] is a hot research topic, especially in recommendation tasks, which models the preference of user based on the user history interaction data. Specifically, in the e-commerce recommendation task [34, 42, 58], the user modeling systems use the purchase history or click records to model the user's intrinsic interest and temporal interest [56, 86]. Most of the research typically utilizes user-item binary relations and assumes a flat preference distribution over items for each user. They neglect the hierarchical discrimination between user intentions and user preferences. Zhu et al. [92] propose a novel key-array memory network with user-intention-item triadic relations, which takes both user intentions and preferences into account for the next-item recommendation. As for the user modeling in the news recommendation task, there is much side information that can be used to obtain a better user preference representation. Wu et al. [74] propose a neural news recommendation approach that can exploit heterogeneous user behaviors, including the search queries and the browsed webpages of the user.

However, to model the user preference of sticker selection, we should not only model the sticker selection history, and the dialog context of each selected sticker should also be considered when modeling the user preference.

2.3 Memory Networks

The memory network proposed by Sukhbaatar et al. [61] generally consists of two components. The first one is a memory matrix to save information (i.e.,memory slots), and the second one is a neural network to read/write the memory slots. The memory network has shown better performance than traditional long-short term memory network in several tasks, such as question answering [18, 48, 55, 61], machine translation [50], text summarization [9, 38], dialog system [11, 75], and recommendation [15, 69, 90]. The reason is that the memory network can store the information in a long time range and has more memory storage units than LSTM that has the single hidden state. Following the memory network, there are many variations of a memory network that have been proposed, i.e., key-value memory network [51] and dynamic memory network [40, 80]. Our method is mainly based on the key-value memory network [51], which employs the user history dialog contexts as the memory keys and the corresponding selected stickers as the memory values. However, there are two main differences between our PESRS model and the previous key-value memory network. First, the user history data are in chronological order, and we should consider the time information when storing them into the memory. To recommend more accurate stickers, the model should not only consider the user preference information stored in the memory but also incorporate the matching result between current dialog context and candidate stickers. The second difference lies in that we propose a dynamic fusion layer that considers both the memory read output and the matching result of the current context. Compared with these methods, we not only implement a key-value memory network but also provide a sticker selection framework that could incorporate the user's preference.

2.4 Visual Question Answering

Sticker recommendation involves the representation of and interaction between images and text, which is related to the Visual Question Answering (VQA) task [19, 25, 46, 54, 59, 60, 68]. Specifically, VQA takes an image and a corresponding natural language question as input and outputs the

answer. It is a classification problem in which candidate answers are restricted to the most common answers appearing in the dataset and requires deep analysis and understanding of images and questions such as image recognition and object localization [26, 49, 76, 81]. Current models can be classified into three main categories: early fusion models, later fusion models, and external knowledge-based models. One state-of-the-art VQA model is Reference [45], which proposes an architecture, positional self-attention with co-attention, that does not require a RNN for video question answering. Reference [29] proposes an image-question-answer synergistic network, where candidate answers are coarsely scored according to their relevance to the image and question pair in the first stage. Then, answers with a high probability of being correct are re-ranked by synergizing with images and questions.

The difference between sticker selection and VQA task is that the sticker selection task focus more on multi-turn multimodal interaction between stickers and utterances.

2.5 Visual Dialog

Visual dialog extends the single turn dialog task [28, 52, 63] in VQA to a multi-turn one, where later questions may be related to former question–answer pairs. To solve this task, Reference [47] transfers knowledge from a pre-trained discriminative network to a generative network with an RNN encoder, using a perceptual loss. Reference [77] combines reinforcement learning and generative adversarial networks (GANs) to generate more humanlike responses to questions, where the GAN helps overcome the relative paucity of training data and the tendency of the typical maximum-likelihood-estimation-based approach to generate overly terse answers. Reference [36] demonstrates a simple symmetric discriminative baseline that can be applied to both predicting an answer as well as predicting a question in the visual dialog.

Unlike visual dialog tasks, in a sticker recommendation system, the candidates are stickers rather than text.

2.6 Multi-turn Response Selection

Multi-turn response selection [8, 17, 44, 65, 82–84] takes a message and utterances in its previous turns as input and selects a response that is natural and relevant to the whole context. In our task, we also need to take previous multi-turn dialog into consideration. Previous works include Reference [88], which uses an RNN to represent context and response and measures their relevance. More recently, Reference [78] matches a response with each utterance in the context on multiple levels of granularity, and the vectors are then combined through an RNN. The final matching score is calculated by the hidden states of the RNN. Reference [89] extends this work by considering the matching with dependency information. More recently, Reference [64] proposes a multi-representation fusion network where the representations can be fused into matching at an early stage, an intermediate stage, or at the last stage.

Traditional multi-turn response selection deals with pure natural language processing, while in our task, we also need to obtain a deep understanding of images.

3 DATASET

In this section, we introduce our multi-turn dialog dataset with sticker as response in detail.

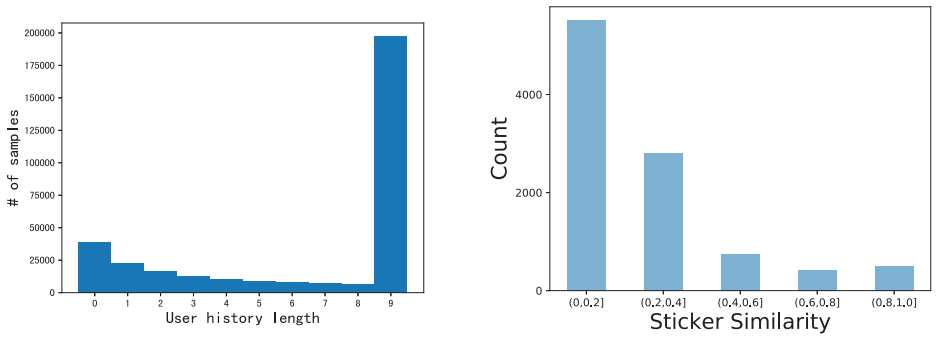
3.1 Data Collection

We collect the large-scale multi-turn dialog dataset with stickers from one of the most popular messaging apps, Telegram.[2] In this app, a large amount of sticker sets are published, and everyone

[2] https://telegram.org/.

Table 1. Statistics of Response Selection Dataset

	Train	Valid	Test
# context-stickers pairs	320,168	10,000	10,000
Avg. words of context utterance	7.54	7.50	7.42
Avg. users participate	5.81	5.81	5.79



(a) The distribution of history length in training dataset. (b) Similarity distribution among all stickers in test dataset.

Fig. 3. Statistics of dataset.

can use the sticker when chatting with a friend or in a chat group. Specifically, we select 20 public chat groups consisting of active members, which are all open groups that everyone can join it without any authorities. The chat history of these groups is collected along with the complete sticker sets. These sticker sets include stickers with similar style. All stickers are resized to a uniform size of 128×128 pixels. We use 20 utterances before the sticker response as the dialog context, and then we filter out irrelevant utterance sentences, such as URL links and attached files. Due to privacy concern, we also filter out user information and anonymize user IDs. To construct negative samples, 9 stickers other than the ground-truth sticker are randomly sampled from the sticker set. After pre-processing, there are 320,168 context–sticker pairs in the training dataset, 10,000 pairs in the validation, and 10,000 pairs in test datasets, respectively. We make sure that there is no overlap between these three datasets and there is no the same dialog context in any two datasets. Two examples are shown in Figure 4. We publish this dataset to communities to facilitate further research on dialog response selection task.

3.2 Statistics and Analysis

In total, there are 3,516 sets of sticker that contain 174,695 stickers. The average number of stickers in a sticker set is 49.64. Each context includes 15.5 utterances on average. The average number of users who participate in the dialog context over each dataset is shown in the third row of Table 1.

Since not all the users have history dialog data, we calculate the percentage of how many data samples in our dataset have history data. There are 290,939 data samples in our training dataset that have at lease one history sticker selection history, and the percentage is 88.12%. We set the maximum of retrieved history data pair (consisting of dialog context and selected sticker) for one data sample to 10, and the average of history length in our training dataset is 6.82. We also plot the distribution of history length in Figure 3(a).

Fig. 4. Example cases in the dataset with different similarity scores.

3.3 Sticker Similarity

Stickers in the same set always share a same style or contain the same cartoon characters. Intuitively, the more similar the candidate stickers are, the more difficult it is to choose the correct sticker from candidates. In other words, the similarity between candidate stickers determines the difficulty of the sticker selection task. To investigate the difficulty of this task, we calculate the average similarity of all the stickers in a specific sticker set by the Structural Similarity Index (SSIM) metric [3, 71]. We first calculate the similarity between the ground-truth sticker and each negative sample and then average the similarity scores. The similarity distribution among test data is shown in Figure 3(b), where the average similarity is 0.258. The examples in Figure 4 are also used to illustrate the similarity of stickers more intuitively, where the left one has a relatively low similarity score and the right one has a high similarity score.

4 PROBLEM FORMULATION

Before presenting our approach for sticker response selection in multi-turn dialog, we first introduce our notations and key concepts. Table 2 lists the main notations we use.

Similarly to the multi-turn dialog response selection [78, 89], we assume that there is a multi-turn dialog context $s = \{u_1, \ldots, u_{T_u}\}$ and a candidate sticker set $C = \{c_1, \ldots, c_{T_c}\}$, where u_i represents the ith utterance in the multi-turn dialog. In the ith utterance $u_i = \{x_1^i, \ldots, x_{T_x^i}^i\}$, x_j^i represents the jth word in u_i, and T_x^i represents the total number of words in u_i utterance. In dialog context s, c_i represents a sticker image with a binary label y_i, indicating whether c_i is an appropriate response for s. T_u is the number of utterance in the dialog context, and T_c is the number of candidate stickers. For each candidate set, there is only one ground-truth sticker, and the remaining ones are negative samples.

To model the user preference, we use T_h history dialog contexts with user-selected sticker $\{(\hat{s}^1, \hat{c}_1), \ldots, (\hat{s}^{T_h}, \hat{c}_{T_h})\}$, where $\hat{s}^i$ denotes the ith history dialog context and $\hat{c}_i$ denotes the user-selected sticker at the ith history dialog context. In the remainer of the article, we use the word *current* to denotes the dialog context s and sticker c_i, which the model needs to predict the sticker selection, and we use the word *history* to denote the dialog context and sticker that user has generated before. In the kth history, there is a dialog context $\hat{s}^k = \{\hat{u}_i^k, \ldots, \hat{u}_{T_u}^k\}$, which contains up to T_u utterances as the same as current dialog context s, and a user-selected sticker $\hat{c}_k$. For each dialog history, we pad the dialog context where the number of utterances is less than T_u to T_u. Our goal is to learn a ranking model that can produce the correct ranking for each candidate sticker

Table 2. Glossary

Symbol	Description
s	multi-turn dialog context
u_i	ith utterance in s
T_u	number of utterances in dialog context
x_j^i	jth word in ith utterance u_i
T_x^i	number of words in the ith utterance
C	candidate sticker set
c_i	ith candidate sticker in c
T_c	number of stickers in candidate sticker set c
y_i	the selection label of ith sticker c_i
$\hat{s}^k$	kth multi-turn dialog context in history
$\hat{u}_i^k$	ith utterance in kth history context $\hat{s}^k$
$\hat{x}_j^{k,i}$	jth word in ith utterance $\hat{u}_i^k$ of kth history context
$\hat{c}_k$	user-selected sticker of kth history
T_h	number of history dialog context and selected sticker

c_i; that is, can we select the correct sticker among all the other candidates? For the rest of the article, we take the ith candidate sticker c_i as an example to illustrate the details of our model and omit the candidate index i for brevity. In some of the sticker selection scenarios, the stickers in the preceding dialog context may affect the current decision of sticker selection. But in most cases, the sticker selection is influenced by a few utterances before. Thus, in this article, we focus on modeling the text utterances in dialog context. And we will consider the information provided by the stickers in the preceding context in our future work.

5 PESRS MODEL

5.1 Overview

In this section, we propose our PESRS. An overview of PESRS is shown in Figure 5, which can be split into five main parts as follows:

- *Sticker encoder* is a convolutional neural network–(CNN) based image encoding module that learns a sticker representation.
- *Utterance encoder* is a self-attention mechanism-based module encoding each utterance u_i in the multi-turn dialog context s.
- *User preference modeling module* is a key-value memory network that stores the representation of history dialog context and corresponding selected sticker.
- *Deep interaction network* module conducts deep matching between each sticker representation and each utterance, and outputs each interaction result.
- *Fusion network* learns the short-term dependency by the fusion RNN and the long-term dependency by the fusion Transformer and finally outputs the matching score by combining the current interaction results with user preference representation using a gated fusion layer.

5.2 Sticker Encoder

Much research has been conducted to alleviate gradient vanishing [31] and reduce computational costs [30] in image modeling tasks. We utilize one of these models, i.e., the Inception-v3 [62] model,

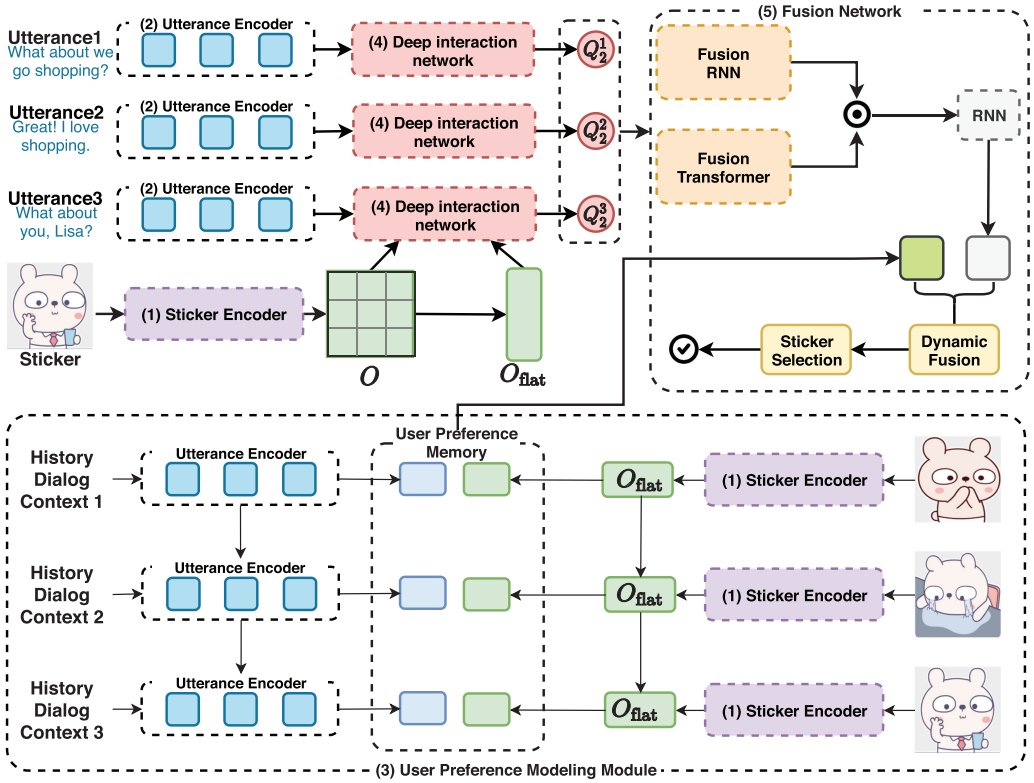


Fig. 5. Overview of PESRS. We divide our model into five ingredients: (1) *Sticker encoder* learns sticker representation by a convolutional neural network; (2) *Utterance encoder* learns representation of each utterance by self-attention-based Transformer; (3) *User preference modeling module* obtains the position-aware history representations and store them into a key-value memory network; (4) *Deep interaction network* conducts deep matching interaction between sticker representation and utterance representation in different levels of granularity; and (5) *Fusion network* combines the long-term and short-term dependency feature between interaction results produced by (4) and the user preference representation produced by (3) into final sticker prediction layer.

rather than plain CNN to encode sticker image:

$$O, O_{\text{flat}} = \text{Inception-v3}(c), \tag{1}$$

where c is the sticker image. The sticker representation is $O \in \mathbb{R}^{p \times p \times d}$, which conserves the two-dimensional information of the sticker and will be used when associating stickers and utterances in Section 5.4. We use the original image representation output of Inception-v3 $O_{\text{flat}} \in \mathbb{R}^d$ as another sticker representation. Most imaging grounded tasks [37, 72, 73] employ the pre-trained image encoding model to produce the image representation. However, existing pre-trained CNN networks including Inception-v3 are mostly built on real-world photos. Thus, directly applying the pre-trained networks on stickers cannot speed up the training process. In this dataset, sticker author give each sticker c an emoji tag that denotes the general emotion of the sticker. Hereby, we propose an auxiliary sticker classification task to help the model converge quickly, which uses O_{flat} to predict which emoji is attached to the corresponding sticker. More specifically, we feed

O_{flat} into a linear classification layer and then use the cross-entropy loss $\mathcal{L}_s$ as the loss function of this classification task.

5.3 Utterance Encoder

To model the semantic meaning of the dialog context, we learn the representation of each utterance u_i. First, we use an embedding matrix e to map a one-hot representation of each word in each utterance u_i to a high-dimensional vector space. We also add the positional embedding to the original word embedding, and we use the $e(x_j^i)$ to denote the embedding representation of word x_j^i. The positional embedding is the same as Transformer [67]. From these embedding representations, we use the attentive module with positional encoding from Transformer [67] to model the interactions between the words in an utterance. Attention mechanisms have become an integral part of compelling sequence modeling in various tasks [5, 16, 21, 45]. In our sticker selection task, we also need to let words fully interact with each other words to model the dependencies of words in the input sentence. The self-attentive module in the Transformer requires three inputs: the query Q, the key K, and the value V. To obtain these three inputs, we use three linear layers with different parameters to project the embedding of dialog context $e(x_j^i)$ into three spaces:

$$Q_j^i = FC(e(x_j^i)), \tag{2}$$

$$K_j^i = FC(e(x_j^i)), \tag{3}$$

$$V_j^i = FC(e(x_j^i)). \tag{4}$$

The self-attentive module then takes each Q_j^i to attend to $K_{\cdot}^i$ and uses these attention distribution $\alpha_{j,\cdot}^i \in \mathbb{R}^{T_x^i}$ as weights to gain the weighted sum of V_j^i, as shown in Equation (6),

$$\alpha_{j,k}^i = \frac{\exp\left(Q_j^i \cdot K_k^i\right)}{\sum_{n=1}^{T_x^i} \exp\left(Q_j^i \cdot K_n^i\right)}, \tag{5}$$

$$\beta_j^i = \sum_{k=1}^{T_x^i} \alpha_{j,k}^i \cdot V_k^i, \tag{6}$$

Next, we add the original word embedding $e(x_j^i)$ on β_j^i as the residential connection layer, shown in Equation (7):

$$\hat{h}_j^i = \text{Dropout}\left(e(x_j^i) + \beta_j^i\right), \tag{7}$$

where $\alpha_{j,k}^i$ denotes the attention weight between the jth word to the kth word in the ith utterance. To prevent vanishing or exploding of gradients, a layer normalization operation [43] is also applied on the output of the feed-forward layers with ReLU activation as shown in Equation (8):

$$h_j^i = \text{norm}\left(max(0, \hat{h}_j^i \cdot W_1 + b_1) \cdot W_2 + b_2 + \hat{h}_j^i\right), \tag{8}$$

where W_1, W_2, b_1, b_2 are all trainable parameters of the feed-forward layer. h_j^i denotes the hidden state of jth word for the ith utterance in the Transformer. We also employ the multi-head attention in our model that conducts these operation multiple times and then concatenate the outputs as the final representation. For brevity, we omit these multi-head operations in our equations.

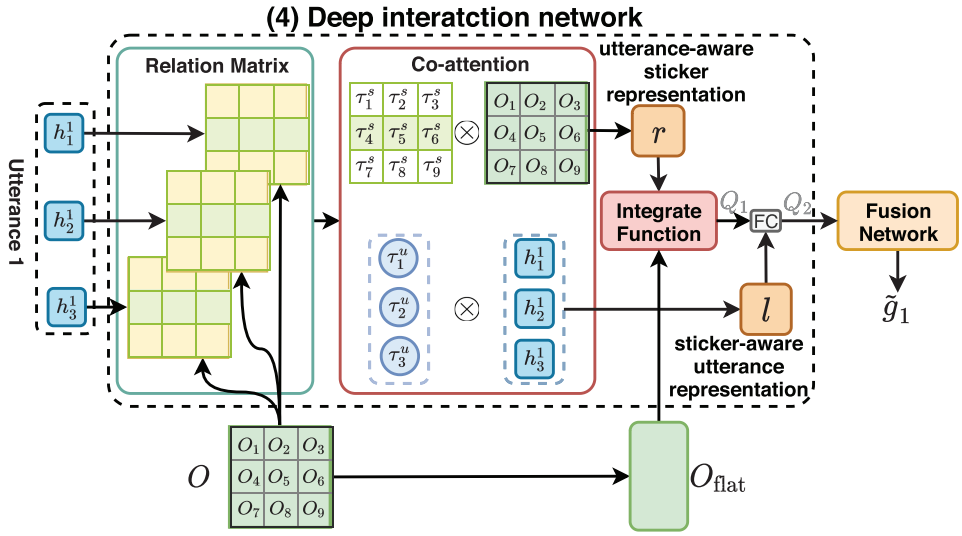


Fig. 6. Framework of deep interaction network.

5.4 Deep Interaction Network

Now that we obtain the representation of the sticker and each utterance, we can conduct a deep matching between these components to model the bi-directional relationship between the words in dialog context and the sticker patches. On one hand, there are some emotional words in dialog context history that match the expression of the stickers such as "happy" or "sad." On the other hand, specific parts of the sticker can also match these corresponding words such as dancing limbs or streaming eyes. Hence, we employ a bi-directional attention mechanism between a sticker and each utterance, that is, from utterance to sticker and from sticker to utterance, to analyze the cross-dependency between the two components. The interaction is illustrated in Figure 6.

We take the ith utterance as an example and omit the index i for brevity. The two directed attentions are derived from a shared relation matrix, $M \in \mathbb{R}^{(p^2)\times T_u}$, calculated by sticker representation $O \in \mathbb{R}^{p\times p\times d}$ and utterance representation $h \in \mathbb{R}^{T_u\times d}$. The score $M_{kj} \in \mathbb{R}$ in the relation matrix M indicates the relation between the kth sticker representation unit O_k, $k \in [1, p^2]$ and the jth word h_j, $j \in [1, T_u]$ and is computed as:

$$M_{kj} = \sigma(O_k, h_j), \tag{9}$$

$$\sigma(x, y) = w^{\intercal}[x \oplus y \oplus (x \otimes y)], \tag{10}$$

where σ is a trainable scalar function that encodes the relation between two input vectors. $\oplus$ denotes a concatenation operation and $\otimes$ is the element-wise multiplication.

Next, a two-way max pooling operation is conducted on M, i.e.,let $\tau_j^u = \max(M_{:j}) \in \mathbb{R}$ represent the attention weight on the jth utterance word by the sticker representation, corresponding to the "utterance-wise attention." This attention learns to assign high weights to the important words that are closely related to sticker. We then obtain the weighted sum of hidden states as "*sticker-aware utterance representation*" l:

$$l = \sum_{j}^{T_u} \tau_j^u h_j. \tag{11}$$

Similarly, stickerwise attention learns which part of a sticker is most relevant to the utterance. Let $\tau_k^s = \max(M_{k:}) \in \mathbb{R}$ represent the attention weight on the kth unit of the sticker representation. We use this to obtain the weighted sum of O_k, i.e., the "*utterance-aware sticker representation*" r:

$$r = \sum_{k}^{p^2} \tau_k^s O_k . \tag{12}$$

After obtaining the two outputs from the co-attention module, we combine the sticker and utterance representations and finally get the ranking result. We first integrate the utterance-aware sticker representation r with the original sticker representation O_{flat} using an *integrate function*, named *IF*:

$$Q_1 = \text{IF}\left(O_{\text{flat}}, r\right), \tag{13}$$

$$\text{IF}(x, y) = \text{FC}\left(x \oplus y \oplus (x \otimes y) \oplus (x + y)\right), \tag{14}$$

where FC denotes the fully connected (FC) layer, and we use the ReLU [53] as the activation function, where $\oplus$ represents the vector concatenation along the final dimension of the vector and $\otimes$ denotes the elementwise product operation. We add the sticker-aware utterance representation l into Q_1 together and then apply a fully connected layer with ReLU activation:

$$Q_2 = \text{FC}(Q_1 \oplus l). \tag{15}$$

5.5 User Preference Modeling Module

Users have their preference when selecting the sticker as the response of the multi-turn dialog context. Hence, to recommend the sticker more accurately, our model should consider the user's preference when giving the final sticker recommendation. Intuitively, the sticker that selected by the user recently contains the user's preference, and these history data can help our model to build the preference representation. As for constructing the preference modeling module, our motivation is to find the semantically similar dialog contexts in the history data, and then use the corresponding selected stickers of these dialog contexts to facilitate the final sticker prediction of the current dialog context. Hence, we propose the user preference memory and the architecture of this module as shown in Figure 7. The proposed user preference memory unit inherits from memory networks [10, 23, 66, 69] and generally has two steps: (1) memory addressing and (2) memory reading. The user preference memory consists of a set of history multi-turn dialog contexts and selected stickers. Though one action is corresponding to a dialog context, it should attend to the different history contexts (i.e., memory slots) upon the current dialog context. Thus, we address and read the memory unit as follows.

5.5.1 History Encoding. To store the history dialog contexts and selected stickers, we encode them into vector spaces using the same method as used when encoding current dialog context and candidate stickers. Concretely, first, attentive module is employed to encode all the dialog contexts $\{\hat{s}^1, \ldots, \hat{s}^{T_h}\}$:

$$\overline{h}_i^k = \text{mean-pooling}(\text{Transformer}(\hat{u}_i^k)), \tag{16}$$

where $\overline{h}_i^k$ is the vector representation of the ith utterance in the kth history, and the Transformer is the same operation as shown in Equation (2)–Equation (8). Different with the Transformer used in Equation (2), the query, key, and value in Equation (16) are all $\hat{u}_i^k$, where we conduct self-attention

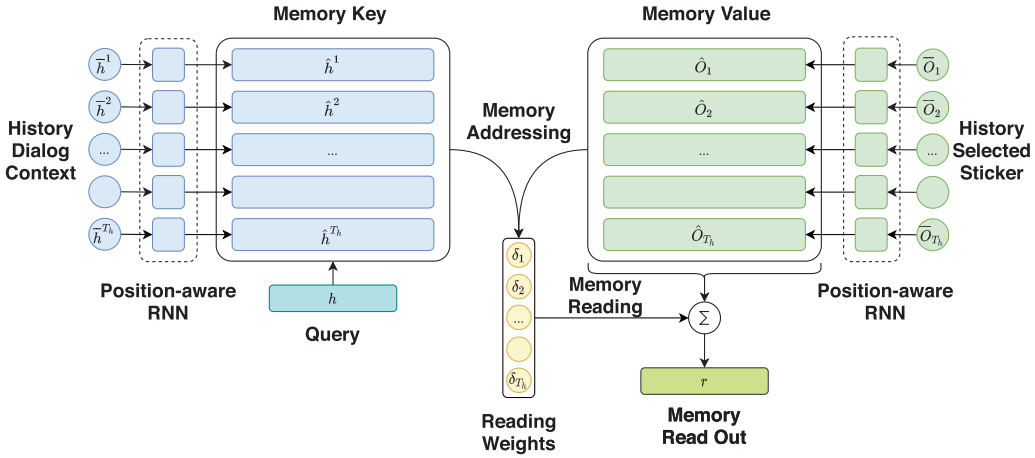


Fig. 7. Framework of user preference modeling module that consists a position-aware RNN and a key-value memory network. We use the history dialog contexts as the keys and the history selected stickers as the values. Finally, we use the representation of the current dialog context as the query the user preference memory.

over all history dialog contexts. Then, we use a max-pooling layer to obtain the vector representation $\overline{h}^k$ of the kth dialog context in history:

$$\overline{h}^k = \max\left(\left\{\overline{h}^k_1, \ldots, \overline{h}^k_{T_h}\right\}\right). \tag{17}$$

Next, we use the same image encoder Inception-v3 in Section 5.2 to encode all the stickers $\{\hat{c}_1, \ldots, \hat{c}_{T_h}\}$ of each history dialog context into vector representations $\{\overline{O}_1, \ldots, \overline{O}_{T_h}\}$:

$$\overline{O}_k = \text{Inception-v3}(\hat{c}_k), \tag{18}$$

where sticker representation $\overline{O}_k \in \mathbb{R}^d$ is a one-dimensional vector, and we drop the output O and use the output O_{flat} as the $\overline{O}_k$.

Intuitively, it is much easier for the user to recall their recently used stickers than the stickers they used a long time ago. Thus, we propose a RNN-based position-aware user history modeling layer that incorporates the position feature into the history data representation, e.g., history dialog context representation $\overline{h}^k$ and history selected sticker representation $\overline{O}_k$. We first concatenate the position of history dialog context as an additional feature to the vector representation of dialog representation $\overline{h}^k$ and sticker representation $\overline{O}_k$. Then, we employ an RNN to encode these representations in chronological order:

$$\hat{h}^k = \text{RNN}(t_k \oplus \overline{h}^k, \hat{h}^{k-1}), \tag{19}$$

$$\hat{O}_k = \text{RNN}(t_k \oplus \overline{O}_k, \hat{O}_{k-1}). \tag{20}$$

Finally, we obtain the position-aware history data representations $\{\hat{h}^1, \ldots, \hat{h}^{T_h}\}$ and $\{\hat{O}_1, \ldots, \hat{O}_{T_h}\}$ and we will introduce how to store them into user preference memory.

5.5.2 Memory Addressing. After obtaining all the vector representations of history sticker and dialog context, we employ a key-value memory network and store them into each key-value slot, as shown in Figure 7. In this memory network, we use the dialog contexts as the keys and use the corresponding selected stickers as the values.

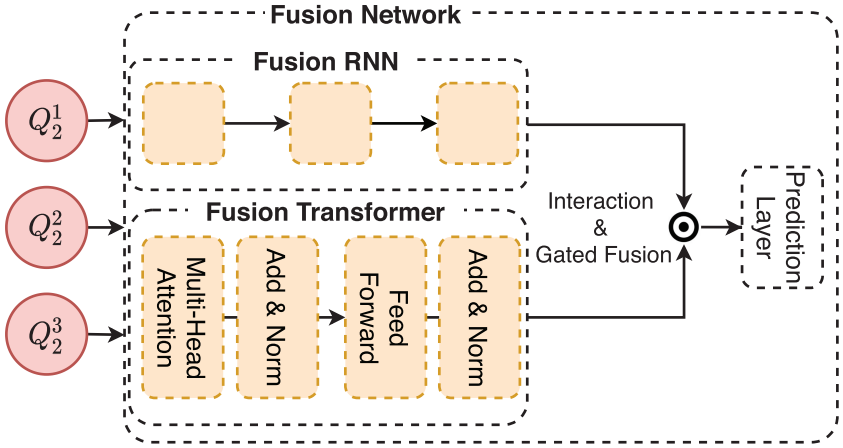


Fig. 8. Framework of fusion network. The black circle in the right of the figure indicates the interaction combination and gated fusion.

First, we construct the query from the current dialog context, which will be used to retrieve the user preference representation from the memory network. We apply a max-pooling layer on the representations of each utterance in the current dialog context:

$$h_m^i = \text{mean-pooling}\left(\left\{h_1^i, \ldots, h_{T_x}^i\right\}\right), \tag{21}$$

$$h = \max\left\{h_m^1, \ldots, h_m^{T_u}\right\}, \tag{22}$$

where h_m^i is the representation of ith utterance in the current dialog context and $h \in \mathbb{R}^d$ is used as the query, and it represents the overall information of the current dialog context. Next, we use h to calculate the read weights over each memory slot:

$$\delta_k = \text{softmax}(hW_\delta \hat{h}^k), \tag{23}$$

where $\delta_k \in [0, 1]$ is the read weight for the kth memory slot and W_δ is a trainable parameter.

5.5.3 Memory Reading. After obtaining the read weights $\{\delta_1, \ldots, \delta_{T_h}\}$ for all the memory slots, we can write the semantic output for preference memory by

$$r = \sum_k^{T_h} \delta_k \hat{O}_k, \tag{24}$$

where r in essence represents a semantic preference representation and will be used when predicting the sticker in current dialog context.

5.6 Fusion Network

Until now, we have obtained the user preference representation and interaction result between each utterance and the candidate sticker. Here, we again include the utterance index i, which has been omitted in previous subsections, and Q_2 now becomes Q_2^i. Since the utterances in a multi-turn dialog context are in chronological order, we employ a *Fusion RNN* and a *Fusion Transformer* to model the short-term and long-term interaction between utterance $\{Q_2^1, \ldots, Q_2^{T_u}\}$. Fusion RNN (shown in the top part of Figure 8) is based on the recurrent network, which can capture short-term dependency over each utterance interaction result. Fusion Transformer (shown in the bottom

part of Figure 8) is based on the self-attention mechanism, which is designed for capturing the important elements and the long-term dependency among all the interaction results.

5.6.1 Fusion RNN. Fusion RNN first reads the interaction results for each utterance $\{Q_2^1, \ldots, Q_2^{T_u}\}$ and then transforms into a sequence of hidden states. In this article, we employ the gated recurrent unit (GRU) [12] as the cell of fusion RNN, which is popular in sequential modeling [20, 64, 78]:

$$g_i = \text{RNN}\left(Q_2^i, g_{i-1}\right), \tag{25}$$

where g_i is the hidden state of the fusion RNN and g_0 is the initial state of RNN, which is initialized randomly. Finally, we obtain the sequence of hidden states $\{g_1, \ldots, g_{T_u}\}$. One can replace GRU with similar algorithms such as Long-Short Term Memory network (LSTM) [33]. We leave the study as future work.

5.6.2 Fusion Transformer. To model the long-term dependency and capture the salience utterance from the context, we employ the self-attention mechanism introduced in Equations (5)–(8). Concretely, given $\{Q_2^1, \ldots, Q_2^{T_u}\}$, we first employ three linear projection layers with different parameters to project the input sequence into three different spaces:

$$\mathcal{Q}^i = \text{FC}(Q_2^i), \tag{26}$$

$$\mathcal{K}^i = \text{FC}(Q_2^i), \tag{27}$$

$$\mathcal{V}^i = \text{FC}(Q_2^i). \tag{28}$$

Then we feed these three matrices into the self-attention algorithm illustrated in Equations (5)–(8). Finally, we obtain the long-term interaction result $\{\hat{g}_1, \ldots, \hat{g}_{T_u}\}$.

5.6.3 Long Short Interaction Combination. To combine the interaction representation generated by fusion RNN and fusion Transformer, we employ the SUMULTI function proposed in Reference [70] to combine these representations, which has been proven effective in various tasks:

$$\overline{g}_i = \text{ReLU}\left(\mathcal{W}^s \begin{bmatrix} (\hat{g}_i - g_i) \otimes (\hat{g}_i - g_i) \\ \hat{g}_i \otimes g_i \end{bmatrix} + \mathbf{b}^s\right), \tag{29}$$

where $\otimes$ is the element-wise product. The new interaction sequence $\{\overline{g}_1, \ldots, \overline{g}_{T_u}\}$ is then boiled down to a matching vector $\tilde{g}_{T_u}$ by another GRU-based RNN:

$$\tilde{g}_i = \text{RNN}(\tilde{g}_{i-1}, \overline{g}_i). \tag{30}$$

We use the final hidden state $\tilde{g}_{T_u}$ as the representation of the overall interaction result between the whole utterance context and the candidate sticker.

5.6.4 Gated Fusion. In the final prediction, our model combines the current dialog context interaction result and user preference representation to predict the final result. However, in each case, the information required for current dialog context interaction and user preference representation is not necessarily the same. If the current dialog context is very similar to the history dialog context, then the historical information should play a greater role in prediction. To incorporate the user preference information into final sticker prediction, we employ a gated fusion that dynamically fuses the current context interaction result and user preference representation together by using a gate f_g. To dynamically fuse these two information sources, we calculate a gate $f_g \in [0, 1]$ that decide which part should the model concentrates on when making the final sticker selection decision:

$$f_g = \sigma(\text{FC}([r \oplus \tilde{g}_{T_u}])), \tag{31}$$

where σ is the sigmoid function and $\oplus$ denotes the vector concatenation operation. Next, we apply a weighted sum operation using the gate f_g on current context interaction result $\tilde{g}_{T_u}$ and user preference representation r, as shown in Equation (32). Finally, we apply a fully connected layer to produce the matching score $\hat{y}$ of the candidate sticker:

$$\hat{y} = \sigma(\text{FC}(f_g * \tilde{g}_{T_u} + (1 - f_g) * r)), \tag{32}$$

where $\hat{y} \in (0, 1)$ is the matching score of the candidate sticker.

5.7 Learning

Recall that we have a candidate sticker set $C = \{c_1, \ldots c_{T_c}\}$ that contains multiple negative samples and one ground-truth sticker. We use hinge loss as our objective function:

$$\mathcal{L} = \sum^{N} \max\left(0, \hat{y}_{\text{negative}} - \hat{y}_{\text{positive}} + \text{margin}\right), \tag{33}$$

where $\hat{y}_{\text{negative}}$ and $\hat{y}_{\text{positive}}$ corresponds to the predicted labels of the negative sample and ground-truth sticker, respectively. The margin is the margin rescaling in hinge loss. The gradient descent method is employed to update all the parameters in our model to minimize this loss function.

6 EXPERIMENTAL SETUP

6.1 Research Questions

We list nine research questions that guide the experiments:

- **RQ1** (See Section 7.1): What is the overall performance of PESRS compared with all baselines?
- **RQ2** (See Section 7.2): What is the effect of each module in PESRS?
- **RQ3** (See Section 7.3): How does the performance change when the number of utterances changes?
- **RQ4** (See Section 7.4): Can co-attention mechanism successfully capture the salient part on the sticker image and the important words in dialog context?
- **RQ5** (See Section 7.5): What is the influence of the similarity between candidate stickers?
- **RQ6** (See Section 7.6): What is the influence of the parameter settings?
- **RQ7** (See Section 7.7): What is the influence of the user history length?
- **RQ8** (See Section 7.8): What is the performance of using the user's most selected sticker as the response?
- **RQ9** (See Section 7.9): Can sticker encoder capture the semantic meaning of sticker?

6.2 Comparison Methods

We first conduct an ablation study to prove the effectiveness of each component in PESRS as shown in Table 3. Specifically, we remove each key part of our PESRS to create ablation models and then evaluate the performance of these models.

Next, to evaluate the performance of our model, we compare it with the following baselines. Note that, we adapt VQA and multi-turn response selection models to the sticker response selection task by changing their input text encoder to image encoder. Since we incorporate the user history data into our model, we also compare with the user modeling method that has been widely used in the recommendation tasks.

(1) **SMN:** Reference [78] proposes a sequential matching network to address response selection for the multi-turn conversation problem. SMN first matches a response with each utterance in the context. Then vectors are accumulated in chronological order through an RNN. The final matching score is calculated with RNN.

Table 3. Ablation Models for Comparison

Acronym	Gloss
PESRS w/o Classify	PESRS w/o emoji classification task
PESRS w/o DIN	PESRS w/o **D**eep **I**nteraction **N**etwork
PESRS w/o FR	PESRS w/o **F**usion **R**NN
PESRS FR2T	Change the **F**usion **R**NN in PESRS to Transformer with positional encoding
PESRS w/o UPM	PESRS w/o **U**ser **P**reference **M**emory
PESRS w/o TAR	PESRS w/o **T**ime-**A**ware **R**NN

(2) **DAM:** Reference [89] extends the transformer model [67] to the multi-turn response selection task, where representations of text segments are constructed using stacked self-attention. Then, truly matched segment pairs are extracted across context and response.

(3) **MRFN:** Reference [64] proposes a multi-representation fusion network that consists of multiple dialog utterance representation methods and generates multiple fine-grained utterance representations. Next, they argue that these representations can be fused into final response candidate matching at an early stage, at the intermediate stage or the last stage. They evaluate all stages and find fusion at the last stage yields the best performance. This is the state-of-the-art model on the multi-turn response selection task.

(4) **Synergistic:** Reference [29] devises a novel synergistic network on VQA task. First, candidate answers are coarsely scored according to their relevance to the image-question pair. Afterward, answers with high probabilities of being correct are re-ranked by synergizing with image and question. This model achieves the state-of-the-art performance on the Visual Dialog v1.0 dataset [13].

(5) **PSAC:** Reference [45] proposes the positional self-attention with co-attention architecture on VQA task, which does not require RNNs for video question answering. We replace the output probability on the vocabulary size with the probability on candidate sticker set.

(6) **SRS:** We propose the first sticker selection method consists of the sticker and dialog context encoding module, deep matching network and information fusion layer in our previous work [22]. This method achieves the state-of-the-art performance on the multi-turn dialog-based sticker selection dataset.

(7) **LSTUR:** Reference [2] proposes a long- and short-term user modeling method to represent the long- and short-term user preference and then apply this method to the news recommendation task. Experiments on a real-world dataset demonstrate their approach can effectively improve the performance of neural news recommendation method. To adapt this method on our sticker selection task, we replace their news encoding network with the sticker image encoding network, Inception-v3, as the same as we used in our model. Since there are countless users in our task, we cannot obtain a static user embedding as they used in their model. For fair, comparison, we replace the user embedding in their model to current dialog context.

For the first three multi-turn response selection baselines, we replace the candidate utterance embedding RNN or Transformer network with the image encoding CNN network Inception-v3, which is the same as used in our proposed model. This Inception-v3 network is initialized using a pre-trained model[3] for all baselines and PESRS.

[3]https://github.com/tensorflow/models/tree/master/research/slim.

Table 4. RQ1: Automatic Evaluation Comparison

	MAP	R_{10}@1	R_{10}@2	R_{10}@5
Visual Q&A methods				
Synergistic	0.593	0.438	0.569	0.798
PSAC	0.662	0.533	0.641	0.836
Multi-turn response selection methods				
SMN	0.524	0.357	0.488	0.737
DAM	0.620	0.474	0.601	0.813
MRFN	0.684	0.557	0.672	0.853
LSTUR	0.689	0.558	0.68	0.874
SRS	0.709	0.590	0.703	0.872
PESRS	**0.743**	**0.632**$^{\blacktriangle}$	**0.740**$^{\blacktriangle}$	**0.897**

Significant differences are with respect to MRFN.

6.3 Evaluation Metrics

Following References [64, 89], we employ recall at position k in n candidates $R_n@k$ as an evaluation metric, which measures if the positive response is ranked in the top k positions of n candidates. Following Reference [89], we also employ mean average precision (MAP) [4] as an evaluation metric. The statistical significance of differences observed between the performance of two runs is tested using a two-tailed paired t-test and is denoted using $^{\blacktriangle}$ (or $^{\blacktriangledown}$) for strong significance at $\alpha = 0.01$.

6.4 Implementation Details

We implement our experiments using TensorFlow [1] on an NVIDIA GTX 2080Ti GPU. If the number of words in an utterance is less than 30, then we pad zeros; otherwise, the first 30 words are kept. The word embedding dimension is set to 100 and the number of hidden units is 100. The batch size is set to 32. 9 negative samples are randomly sampled from the sticker set containing the ground-truth sticker, and we finally obtain 10 candidate stickers for the model to select. We initialize all the parameters randomly using a Gaussian distribution in $[-0.02, 0.02]$. We use Adam optimizer [39] as our optimizing algorithm, and the learning rate is 1×10^{-4}.

7 EXPERIMENTAL RESULT

7.1 Overall Performance

For research question **RQ1**, we examine the performance of our model and baselines in terms of each evaluation metric, as shown in Table 4. First, the performance of the multi-turn response selection models is generally consistent with their performances on text response selection datasets. SMN [78], an earlier work on multi-turn response selection task with a simple structure, obtains the worst performance on both sticker response and text response selection. DAM [89] improves the SMN model and gets better performance. MRFN [64] is the state-of-the-art text response selection model and achieves the best performance among baselines in our task as well. Second, VQA models perform generally worse than multi-turn response selection models, since the interaction between multi-turn utterances and sticker is important, which is not taken into account by VQA models. Third, our previously proposed SRS achieves better performance with 3.36%, 5.92%, and 3.72% improvements in MAP, R_{10}@1, and R_{10}@2, respectively, over the state-of-the-art multi-turn selection model, i.e., MRFN, and with 6.80%, 10.69%, and 8.74% significant increases (with p-value <0.05) over the state-of-the-art visual dialog model, PSAC. Finally, comparing with our previously proposed sticker selection method SRS, our newly proposed model PESRS that incorporates the

Table 5. RQ2: Evaluation of Different Ablation Models

	MAP	R_{10}@1	R_{10}@2	R_{10}@5
PESRS w/o Classify	0.714	0.598	0.707	0.866
PESRS w/o DIN	0.728	0.612	0.725	0.888
PESRS w/o FR	0.727	0.609	0.725	0.886
PESRS FR2T	0.725	0.610	0.719	0.881
PESRS w/o UPM	0.709	0.590	0.703	0.872
PESRS w/o TAR	0.710	0.589	0.706	0.873
PESRS	**0.743**	**0.632**▲	**0.740**▲	**0.897**

user preference information achieves the state-of-the-art performance with 4.8%, 7.1%, and 5.3% improvements in *MAP*, R_{10}@1, and R_{10}@2, respectively, over our previous method SRS, which is just based on the multi-modal matching between utterance and sticker image. That demonstrates the superiority of incorporating the user preference information into sticker selection model.

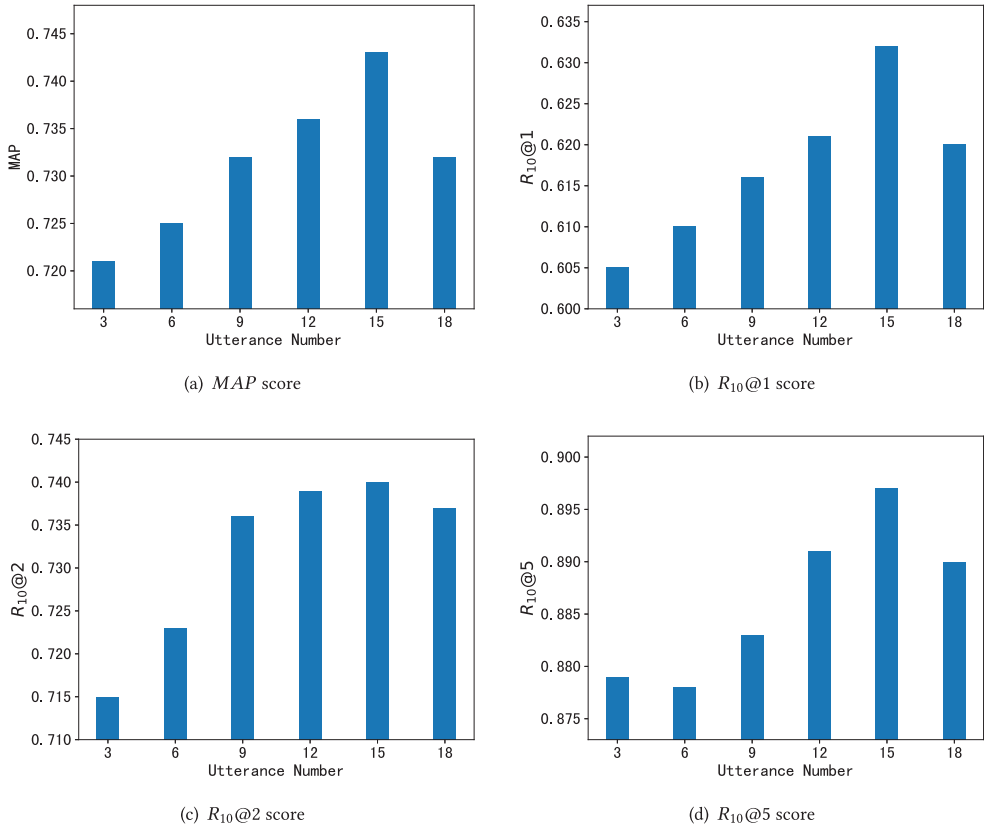
7.2 Ablation Study

For research question **RQ2**, we conduct ablation tests on the use of the sticker classification loss (introduced in Section 5.2), the deep interaction network (introduced in Section 5.4), the fusion RNN (introduced in Section 5.6.1), the user preference memory without position aware RNN (introduced in Section 5.5) and the full user preference memory (introduced in Section 5.5.1), respectively. The evaluation results are shown in Table 5. The performances of all ablation models are worse than that of PESRS under all metrics, which demonstrates the necessity of each component in PESRS. We also find that the sticker classification makes contribution to the overall performance. And this additional task can also speed up the training process, and help our model to converge quickly. We use 21 hours to train the PESRS until convergence, and we use 35 hours for training PESRS w/o Classify. The fusion RNN brings a significant contribution (with p-value <0.05), improving the *MAP* and R_{10}@1 scores by 2.2% and 3.8%, respectively. We also change the fusion RNN to a Transformer with positional encoding, which leads to a decrease of the performance that verifies the effectiveness of fusion RNN. The deep interaction network also plays an important part. Without this module, the interaction between the sticker and utterance are hindered, leading to a 3.3% drop in R_{10}@1. Particularly, since the user preference memory capture the preference of user's sticker selection, we can see that when the user preference memory is removed from the model, the model suffers from dramatic performance drop in terms of all metrics. And the position-aware user history encoding RNN also makes contribution to the PESRS model, improving the *MAP* and R_{10}@1 scores by 4.6% and 7.3%, respectively.

7.3 Analysis of Number of Utterances

For research question **RQ3**, in addition to comparing with various baselines, we also evaluate our model when reading different number of utterances to study how the performance relates to number of context turns.

Figure 9 shows how the performance of the PESRS changes with respect to different numbers of utterances turns. In this experiment, we change the numbers of utterances turns in both current dialog context and history dialog contexts. We observe a similar trend for PESRS on the first three evaluation metrics *MAP*, R_{10}@1, and R_{10}@2: They first increase until the utterance number reaches 15 and then fluctuate as the utterance number continues to increase. There are two possible reasons for this phenomena. The first reason might be that, when the information in the



(a) *MAP* score (b) R_{10}@1 score

(c) R_{10}@2 score (d) R_{10}@5 score

Fig. 9. RQ3: Performance of PRSRS on all metrics when reading different number of utterances.

utterances is limited, the model can capture the features well, and thus when the amount of information increases, the performance gets better. However, the capacity of the model is limited, and when the amount of information reaches its upper bound, it gets confused by this overwhelming information. The second reason might be of the usefulness of the utterance context. Utterances that occur too early before the sticker response may be irrelevant to the sticker and bring unnecessary noise. As for the last metric, the above observations do not preserve. The R_{10}@5 scores fluctuate when the utterance number is below 15, and drop when the utterance number increases. The reason might be that R_{10}@5 is not a strict metric, and it is easy to collect this right sticker in the set of half of the whole candidates. Thus, the growth of the information given to PESRS does not help it perform better but the noise it brings harms the performance. However, though the number of utterances changes from 3 to 18, the overall performance of PESRS generally remains at a high level, which proves the robustness of our model.

7.4 Analysis of Attention Distribution in Interaction Process

Next, we turn to address **RQ4**. We also show three cases with the dialog context in Figure 10. There are four stickers under each dialog context, one is the selected sticker by our model and other three stickers are random selected candidate stickers. As a main component, the deep interaction network comprises a bi-directional attention mechanism between the utterance and the sticker, where each word in the utterance and each unit in the sticker representation have a similarity score in

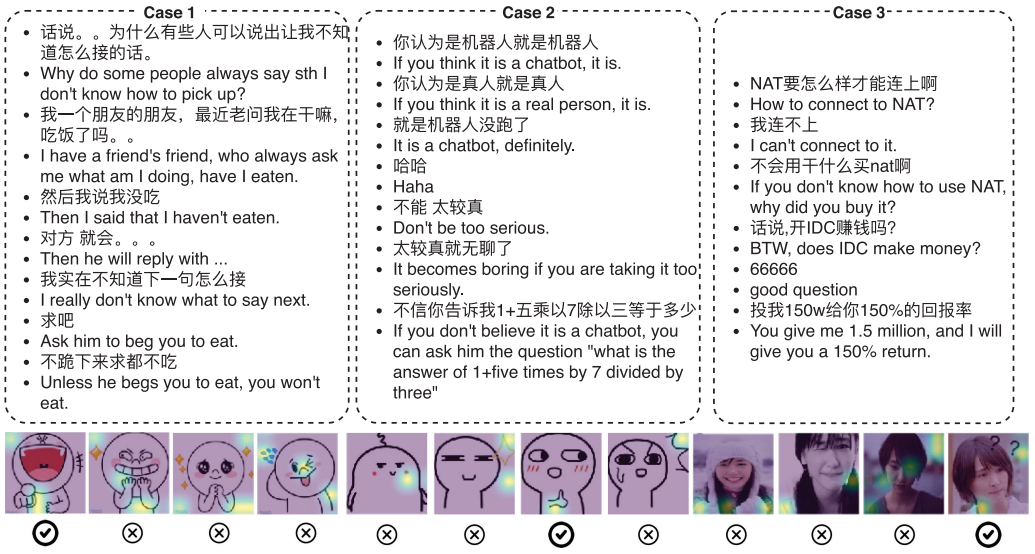


Fig. 10. RQ4: Examples of sticker selection results produced by SRS. We show the selected sticker and three random selected candidate stickers with the attention heat map. The lighter the area on image is, the higher attention weight it gets. The first two cases are collected from a chitchat group, and the third one is collected from a VPN custom service group.

the co-attention matrix. To visualize the sticker selection process and to demonstrate the interpretability of deep interaction network, we visualize the stickerwise attention τ^s (Equation (12)) on the original sticker image and show some examples in Figure 10. The lighter the area is, the higher attention it gets.

Facial expressions are an important part in sticker images. Hence, we select several stickers with vivid facial expression in Figure 10. Take the fourth sticker in Case 1, for example, where the character has a wink eye and a smiling mouth. The highlights are accurately placed on the character's eye, indicating that the representation of this sticker is highly dependent on this part. Another example is the last sticker of Case 3: There are two question marks on the top right corner of the sticker image, which indicates that the girl is very suspicious of this. In addition to facial expression, the characters gestures can also represent emotions. Take the third sticker in Case 2, for example: The character in this sticker gives a thumbs up representing support and we can find that the attention lies on his hand, indicating that the model learns the key point of his body language.

Furthermore, we randomly select three utterances from the test dataset, and we also visualize the attention distribution over the words in an utterance, as shown in Figure 11. We use the weight τ_j^u for the jth word (calculated in Equation (11)) as the attention weight. We can find that the attention module always gives a higher attention weight on the salience word, such as the "easy method," "make a lot of money," and "use Chine Mobile."

7.5 Influence of Similarity between Candidates

In this section, we turn to **RQ5** to investigate the influence of the similarities between candidates. The candidate stickers are sampled from the same set, and stickers in a set usually have a similar style. Thus, it is natural to ask: Can our model identify the correct sticker from a set of similar candidates? What is the influence of the similarity between candidate stickers? Hence, we use the

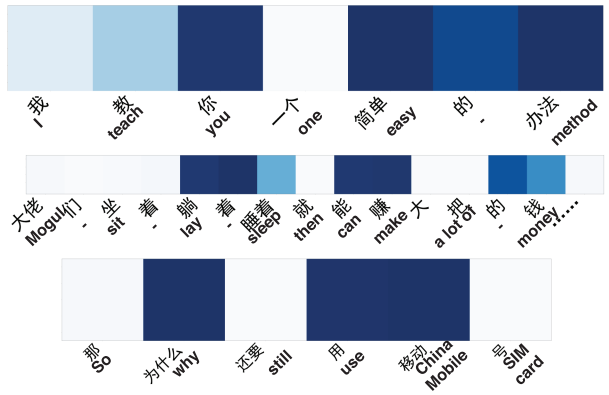


Fig. 11. RQ4: Examples of the attention weights of the dialog utterance. We translate Chinese to English word by word. The darker the area, the higher weight the word gets.

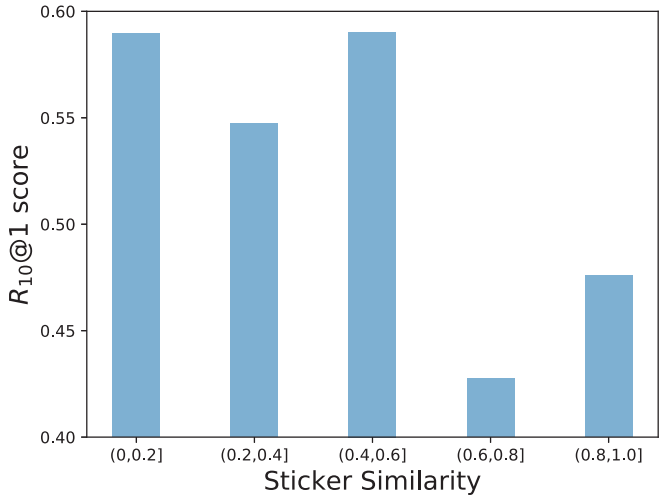


Fig. 12. RQ5: Performance of SRS on groups of different candidate similarity.

SSIM metric [3, 71] to calculate the average similarity among all candidates in a test sample and then aggregate all test samples into five groups according to their average similarities. We calculate the R_{10}@1 of each group of samples, as shown in Figure 12. The x-axis is the average similarity between candidate stickers and the y-axis is the R_{10}@1 score.

Not surprisingly, our model gains the best performance when the average similarity of the candidate group is low and its performance drops as similarity increases. However, we can also see that, though similarity varies from minimum to maximum, the overall performance can overall stay at high level. R_{10}@1 scores of all five groups are above 0.42, and the highest score reaches 0.59. That is, our model is highly robust and can keep giving reasonable sticker responses.

7.6 Robustness of Parameter Setting

In this section, we turn to address **RQ6** to investigate the robustness of parameter setting. We train PESRS model in different parameter setting as shown in Figure 13. The hidden size of the RNN, CNN and the dense layer in our model is tuned from 50 to 200, and we use the MAP and $R_n@k$ to

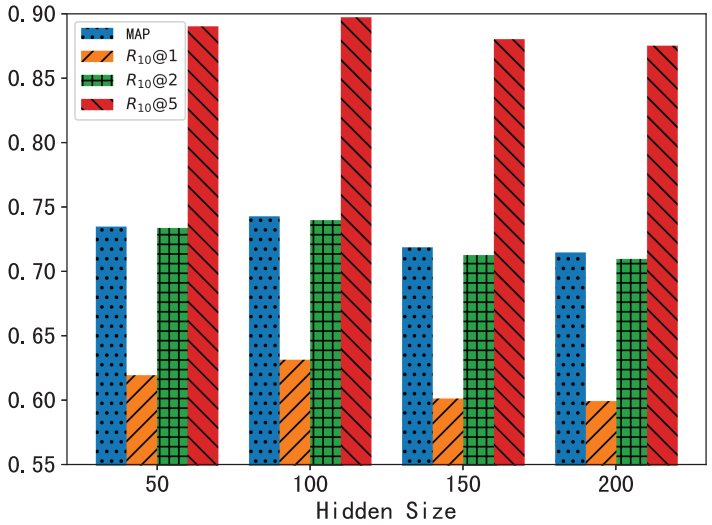


Fig. 13. RQ6: Performance of PESRS with different parameter settings.

evaluate each model. As the hidden size grows larger from 50 to 100, the performance rises, too. The increment of hidden size improves the MAP and R_{10}@1 scores by 1.1% and 1.9%. When the hidden size continuously goes larger from 100 to 200, the performance is declined slightly. The increment of hidden size leads to a 3.9% and 5.3% drop in terms of MAP and R_{10}@1, respectively. Nonetheless, we can find that each metric maintained at a stable interval, which demonstrates that our PESRS is robust in terms of the parameter size.

7.7 Influence of User History Length

Next, we address **RQ7**, which focuses on the influence of using different lengths of user history. We feed different lengths of user sticker selection history to the model, and we show the model performance of different lengths in Figure 14. From this figure, we can see that the model performs worse when we just feed only 2 user stickers into the selection history. The sticker selection prediction performance of the model rises sharply as the history length increases. This indicates that it requires a large amount of user behavior patterns to model the preference of the user. And the growth of user behavior sequence helps PESRS to better capture sticker selection patterns according to the dialog context.

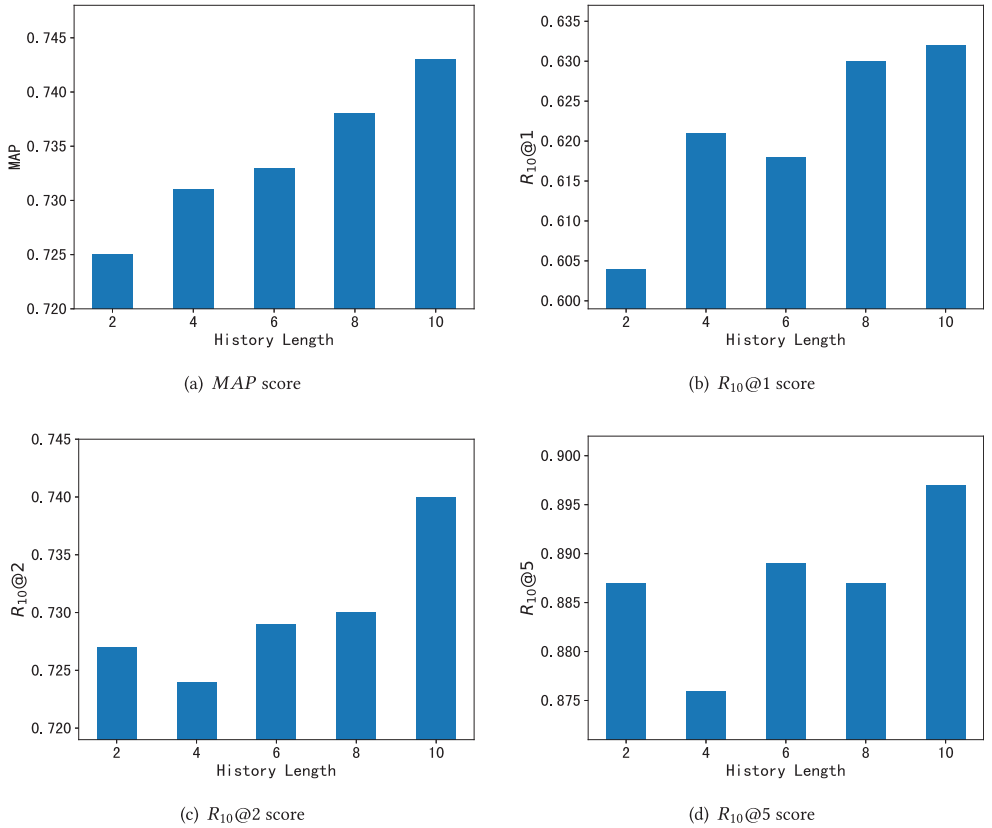
7.8 Analysis of User Preference Memory

Next, we turn to **RQ8** to investigate the effectiveness of user preference modeling module. We propose a simple heuristic method and two variations of our user preference memory module.

To verify the necessity of using user preference modeling network, we use a simple heuristic method (*MostSelected*) that just uses the most selected sticker by user as the sticker prediction of current dialog context. This method does not consider the semantic matching degree of previous dialog context and current dialog context. Consequently, the predicted sticker of this heuristic method is not flexible.

The first variation (*AverageMem*) is to simply apply an average-pooling layer on all the previous selected sticker representations by the corresponding user:

$$r = \sum_{k}^{T_h} \hat{O}_k. \tag{34}$$



(a) *MAP* score (b) R_{10}@1 score

(c) R_{10}@2 score (d) R_{10}@5 score

Fig. 14. RQ7: Performance of PESRS with different user history length.

Then we use this as the user preference representation and feed r to the final gated fusion layer, as shown in Equation (31).

The second variation (*WeightedMem*) is to remove the key addressing process and directly apply an attention-then-weighted method on all the user previous selected stickers. This variation can be split into two steps: (1) calculate attention weights and (2) weighted sum stickers. We use the query vector h (shown in Equation (22)) to calculate attention weights of each user previously selected stickers $\{\hat{O}_1, \ldots, \hat{O}_{T_h}\}$, and the query vector h is the same as used in our proposed user preference memory module:

$$\delta_k = \text{softmax}(hW_\delta \hat{O}_k), \tag{35}$$

where $\delta_k \in [0, 1]$ is the attention weight for the kth selected sticker. Then we apply the weighted-sum on all the user previously selected sticker representations:

$$r = \sum_{k}^{T_h} \delta_k \hat{O}_k. \tag{36}$$

Finally, we feed this preference representation r into final gated fusion layer (Equation (31)). Note that the above two variations exclude the histories of dialogue contexts, and we employ these experiments to verify the effectiveness of incorporating histories of dialogue contexts.

Table 6. RQ8: Performance of Two Variations User Preference Memory Module

	MAP	R_{10}@1	R_{10}@2	R_{10}@5
SRS	0.709	0.590	0.703	0.872
MostSelected	0.545	0.419	0.490	0.679
AverageMem	0.701	0.573	0.701	0.870
WeightedMem	0.694	0.565	0.689	0.866
PESRS	**0.743**	**0.632**▲	**0.740**▲	**0.897**

We conduct the experiments on these variations and compare with our proposed PESRS and SRS, as shown in Table 6. From this table, we can find that *MostSelected* performs the worst among all the methods. That demonstrates the necessity of exploring a learning-based method to leverage the user history data to recommend the proper sticker. By comparing *AverageMem* with the SRS, which does not incorporate the user's history, we find that although *AverageMem* and *WeightedMem* leverages the user's history information, it cannot take advantages from these data to boost the performance of sticker selection. The reason is that these methods cannot model the relationship between current dialog context and previous history data, thus it cannot determine which history data may be helpful for the current context.

7.9 Sticker Classification and Emotion Diversity

Finally, we turn to **RQ9**. In this dataset, the sticker authors give each sticker an emoji label that indicates the approximate emotion of the sticker. However, this label is not a mandatory field when creating a sticker set in this online chatting platform. Some authors use random emoji or one emoji label for all the stickers in the sticker set. Thus, we cannot incorporate the emoji label and tackle the sticker selection task as an emoji classification task. We randomly sample 20 sticker sets and employ human annotators to check whether the emoji label in sticker set is correct, and we find that there are 2 sticker set of them have wrong emoji labels for the stickers. Since we introduce the auxiliary sticker classification (introduced in Section 5.2) to help the model for accelerating convergence of the model training, we also report the sticker classification performance in this article. Note that, since the emoji label of the sticker may not be correct, therefore, the classification performance is *not accurate*, the results are for reference only. The results of the sticker classification are 65.74%, 50.75%, 47.02%, and 61.20% for accuracy, F1, recall and precision, respectively. These results indicate that the sticker encoder can capture the semantic meanings of the sticker image.

To illustrate the diversity of the emotion expressed by the stickers, we use the emoji label as the indicator of the emotion and plot the distribution of the emoji label of stickers. In Figure 15, we only show the top 50 emoji labels used in all the sticker set in our training dataset, and the total number of unique emoji label is 893. From Figure 15, we can find that there are many stickers with the emoji label 😂 and 😏. The reason is that some of the sticker authors assign 😂 or 😏 as the emoji label to all the stickers in their sticker set, as we mentioned before (some authors use random emoji or one emoji label for all the stickers in the sticker set).

8 CONCLUSION

In our previous work, we propose the task of multi-turn sticker response selection, which recommends an appropriate sticker based on multi-turn dialog context history without relying on external knowledge. However, this method only focuses on measuring the matching degree between the dialog context and sticker image, which ignores the user preference of using stickers.

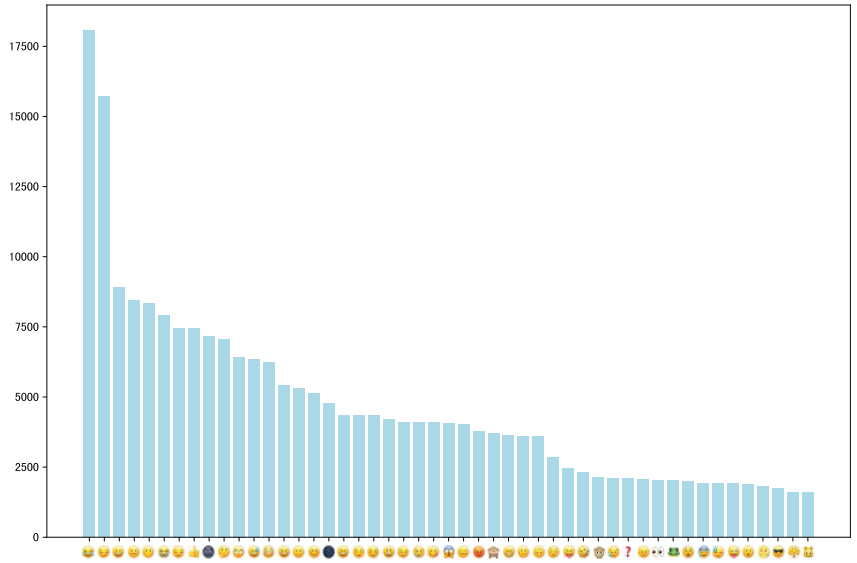


Fig. 15. RQ9: Number of the used emoji labels over stickers in training dataset (top 50 emojis of 893 unique emojis in total).

Hence, in this article, we propose the PESRS to recommend an appropriate sticker to user based on multi-turn dialog context and sticker using history of user. Specifically, PESRS first learns the representation of each utterance using a self-attention mechanism, and learns sticker representation by CNN. Second, a deep interaction network is employed to fully model the dependency between the sticker and utterances. The deep interaction network consists of a co-attention matrix that calculates the attention between each word in an utterance and each unit in a sticker representation. Third, a bi-directional attention is used to obtain utterance-aware sticker representation and sticker-aware utterance representations. Next, we retrieve the recent user sticker selections, and then propose a user preference modeling module that consists a position-aware history encoding network and a key-value-based memory network to generate the user preference representation dynamically according to current dialog context. Then, a fusion network models the short-term and long-term relationship between interaction results, and a gated fusion layer is applied to fuse the current dialog interaction results and user preference representation dynamically. Finally, a fully connected layer is applied to obtain the final sticker prediction using the output of gated fusion layer. Our model outperforms state-of-the-art methods including our previous method SRS in all metrics and the experimental results also demonstrate the effectiveness of each module in our model. In the near future, we aim to propose a personalized sticker response selection system.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. We also thank Anna Hennig in Inception Institute of Artificial Intelligence for her help on this article.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI*, Vol. 16. 265–283.

[2] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long- and short-term user representations. In *ACL*.

[3] Alireza Avanaki. 2008. Exact histogram specification optimized for structural similarity. *arXiv:0901.0065*. Retrieved from https://arxiv.org/abs/0901.0065.
[4] Ricardo Baeza-Yates, Berthier de Araújo Neto Ribeiro, et al. 2011. *Modern Information Retrieval*. ACM Press, New York, NY.
[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[6] Francesco Barbieri, Miguel Ballesteros, Francesco Ronzano, and Horacio Saggion. 2018. Multimodal emoji prediction. In *NAACL*. 679–686.
[7] Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. 2017. Are emojis predictable? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, 105–111.
[8] Zhangming Chan, Juntao Li, Xiaopeng Yang, Xiuying Chen, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Modeling personalization in continuous space for response generation via augmented wasserstein autoencoders. In *EMNLP*.
[9] Xiuying Chen, Zhangming Chan, Shen Gao, Meng-Hsuan Yu, Dongyan Zhao, and Rui Yan. 2019. Learning towards abstractive timeline summarization. In *IJCAI*.
[10] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *WSDM*.
[11] Eric Chu, Prashanth Vijayaraghavan, and Deb Roy. 2018. Learning personas from dialogue with attentive memory networks. In *EMNLP*.
[12] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop*.
[13] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra. 2017. Visual dialog. In *CVPR*. 1080–1089.
[14] Gabriele de Seta. 2018. Biaoqing: The circulation of emoticons, emoji, stickers, and custom images on Chinese digital media platforms. *First Monday* 23, 9–3 (2018).
[15] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *SIGIR*.
[16] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 889–898.
[17] Jiazhan Feng, Chongyang Tao, Wei Wu, Yansong Feng, Dongyan Zhao, and Rui Yan. 2019. Learning a matching model with co-teaching for multi-turn response selection in retrieval-based dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 3805–3815.
[18] Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. 2018. Motion-appearance co-memory networks for video question answering. In *CVPR*.
[19] Peng Gao, Haoxuan You, Zhanpeng Zhang, Xiaogang Wang, and Hongsheng Li. 2019. Multi-modality latent interaction network for visual question answering. In *ICCV*.
[20] Shen Gao, Xiuying Chen, Piji Li, Zhangming Chan, Dongyan Zhao, and Rui Yan. 2019. How to write summaries with patterns? Learning towards abstractive summarization through prototype editing. 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, 3741–3751.
[21] Shen Gao, Xiuying Chen, Piji Li, Zhaochun Ren, Lidong Bing, Dongyan Zhao, and Rui Yan. 2019. Abstractive text summarization by incorporating reader comments. In *AAAI*. 6399–6406.
[22] Shen Gao, Xiuying Chen, Chang Liu, Li Liu, Dongyan Zhao, and Rui Yan. 2020. Learning to respond with stickers: A framework of unifying multi-modality in multi-turn dialog. In *WWW*. Association for Computing Machinery, New York, NY.
[23] Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in E-commerce question-answering. In WSDM. Association for Computing Machinery, New York, NY, 429–437.
[24] Jing Ge and Susan C. Herring. 2018. Communicative functions of emoji sequences on Sina Weibo. *First Monday* 23, 11–5 (2018).
[25] Ankit Goyal, Jian Wang, and Jia Deng. 2018. Think visually: Question answering through virtual imagery. In *ACL (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 2598–2608.
[26] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*. 6904–6913.
[27] Gaël Guibon, Magalie Ochs, and Patrice Bellot. 2018. Emoji recommendation in private instant messages. In*SAC* (2018).
[28] Dan Guo, Hui Wang, and Meng Wang. 2019. Dual visual attention network for visual dialog. In *IJCAI*.

[29] D. Guo, C. Xu, and D. Tao. 2019. Image-question-answer synergistic network for visual dialog. In *CVPR'19*. 10426–10435.
[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*. 1026–1034.
[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.
[32] Susan C. Herring and Ashley Dainas. 2017. "Nice picture comment!" Graphicons in Facebook comment threads. In *HICSS*.
[33] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.
[34] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *WSDM*.
[35] Xiaowen Huang, Quan Fang, Shengsheng Qian, Jitao Sang, Yan Li, and Changsheng Xu. 2019. Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In *MM*.
[36] Unnat Jain, Svetlana Lazebnik, and Alexander G. Schwing. 2018. Two can play this game: Visual dialog with discriminative question generation and answering. In *ICCV*. 5754–5763.
[37] Baoyu Jing, Pengtao Xie, and Eric Xing. 2018. On the automatic generation of medical imaging reports. In *ACL*. Association for Computational Linguistics, Melbourne, Australia, 2577–2586.
[38] Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *NAACL*.
[39] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*, Vol. abs/1412.6980.
[40] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. *arXiv*: 1506.07285. Retrieved from https://arxiv.org/abs/1506.07285.
[41] Abhishek Laddha, Mohamed Hanoosh, and Debdoot Mukherjee. 2019. Understanding chat messages for sticker recommendation in hike messenger. *arXiv:1902.02704*. Retrieved from https://arxiv.org/abs/1902.02704.
[42] Chenyi Lei, Shouling Ji, and Zhao Li. 2019. TiSSA: A time slice self-attention approach for modeling sequential user behaviors. In *WWW*.
[43] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv:1607.06450*. Retrieved from https://arxiv.org/abs/1607.06450.
[44] Juntao Li, Lisong Qiu, Bo Tang, Dongmin Chen, Dongyan Zhao, and Rui Yan. 2019. Insufficient data can also rock! Learning to converse using smaller data with augmentation. In *AAAI*.
[45] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. 2019. Beyond RNNs: Positional self-attention with co-attention for video question answering. In *AAAI*.
[46] Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. 2018. Visual question generation as dual task of visual question answering. In *CVPR*.
[47] Jiasen Lu, Anitha Kannan, Jianwei Yang, Devi Parikh, and Dhruv Batra. 2017. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *NIPS*. 314–324.
[48] Chao Ma, Chunhua Shen, Anthony Dick, Qi Wu, Peng Wang, Anton van den Hengel, and Ian Reid. 2018. Visual question answering with memory-augmented networks. In *CVPR*.
[49] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. 2015. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*. 1–9.
[50] Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *ACL*.
[51] Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. arxiv:1606.03126. Retrieved from https://arxiv.org/abs/1606.03126.
[52] Vishvak Murahari, Prithvijit Chattopadhyay, Dhruv Batra, Devi Parikh, and Abhishek Das. 2019. Improving generative visual dialog by answering diverse questions. In *EMNLP*.
[53] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.
[54] H. Noh, T. Kim, J. Mun, and B. Han. 2019. Transfer learning via unsupervised task discovery for visual question answering. In *CVPR*. 8377–8386.
[55] Juan Pavez, Héctor Allende, and Héctor Allende-Cid. 2018. Working memory networks: Augmenting memory networks with a relational reasoning module. In *ACL*.
[56] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *KDD*.
[57] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, and Kun Gai. 2019. Lifelong sequential modeling with personalized memorization for user response prediction. In *SIGIR*.

[58] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A repeat aware neural recommendation machine for session-based recommendation. In *AAAI*.
[59] F. Sha, H. Hu, and W. Chao. 2018. Cross-dataset adaptation for visual question answering. In *CVPR*. 5716–5725.
[60] Zhou Su, Chen Zhu, Yinpeng Dong, Dongqi Cai, Yurong Chen, and Jianguo Li. 2018. Learning visual knowledge memory networks for visual question answering. In *CVPR*.
[61] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *NIPS*.
[62] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*. 2818–2826.
[63] Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018. Get the point of my utterance! Learning towards effective responses with multi-head attention mechanism. In *IJCAI*.
[64] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *WSDM*. ACM, 267–275.
[65] 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 *ACL*. Association for Computational Linguistics, Florence, Italy, 1–11.
[66] Zhiqiang Tao, Sheng Li, Zhaowen Wang, Chen Fang, Longqi Yang, Handong Zhao, and Yun Fu. 2019. Log2Intent: Towards interpretable user modeling via recurrent semantics memory unit. In *KDD*.
[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.
[68] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2017. Explicit knowledge-based reasoning for visual question answering. In *IJCAI*. 1290–1296.
[69] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural memory streaming recommender networks with adversarial training. In *KDD*.
[70] Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *ICLR*.
[71] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, Eero P. Simoncelli, et al. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* 13, 4 (2004), 600–612.
[72] Chenfei Wu, Jinlai Liu, Xiaojie Wang, and Xuan Dong. 2018. Chain of reasoning for visual question answering. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 275–285.
[73] Chenfei Wu, Jinlai Liu, Xiaojie Wang, and Xuan Dong. 2018. Object-difference attention: A simple relational attention for visual question answering. In MM. Association for Computing Machinery, New York, NY, 519–527.
[74] Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with heterogeneous user behavior. In *EMNLP*.
[75] Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. In *ICLR*.
[76] Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2016. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *CVPR*. 4622–4630.
[77] Qi Wu, Peng Wang, Chunhua Shen, Ian Reid, and Anton van den Hengel. 2018. Are you talking to me? reasoned visual dialog generation through adversarial learning. In *ICCV*. 6106–6115.
[78] 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 *ACL*. Association for Computational Linguistics, Vancouver, Canada, 496–505.
[79] Ruobing Xie, Zhiyuan Liu, Rui Yan, and Maosong Sun. 2016. Neural emoji recommendation in dialogue systems. *arXiv:1612.04609*. Retrieved from https://arxiv.org/abs/1612.04609.
[80] Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. arXiv:1603.01417. Retrieved from https://arxiv.org/1603.01417.
[81] Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *ICML*. 2397–2406.
[82] Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In SIGIR. Association for Computing Machinery, New York, NY, 55–64.
[83] Rui Yan and Dongyan Zhao. 2018. Coupled context modeling for deep chit-chat: Towards conversations between human and computer. In KDD. Association for Computing Machinery, New York, NY, 2574–2583.
[84] Rui Yan, Dongyan Zhao, and Weinan E. 2017. Joint learning of response ranking and next utterance suggestion in human-computer conversation system. In SIGIR. Association for Computing Machinery, New York, NY, 685–694.
[85] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. 2017. Multi-site user behavior modeling and its application in video recommendation. In *SIGIR*.

[86] Zeping Yu, Jianxun Lian, Ahmad Mahmoody, Gongshen Liu, and Xing Xie. 2019. Adaptive user modeling with long and short-term preferences for personalized recommendation. In *IJCAI*.
[87] Guoshuai Zhao, Zhidan Liu, Yulu Chao, and Xueming Qian. 2020. CAPER: Context-aware personalized emoji recommendation. *IEEE Trans. Knowl. Data Eng.*
[88] 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 *EMNLP*. Association for Computational Linguistics, 372–381.
[89] 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 *ACL*, Vol. 1. 1118–1127.
[90] Xiao Zhou, Cecilia Mascolo, and Zhongxiang Zhao. 2019. Topic-enhanced memory networks for personalised point-of-interest recommendation. In *KDD*.
[91] Xianda Zhou and William Yang Wang. 2018. MojiTalk: Generating emotional responses at scale. In *ACL*.
[92] Nengjun Zhu, Jian Cao, Yanchi Liu, Yang Yang, Haochao Ying, and Hui Xiong. 2020. Sequential modeling of hierarchical user intention and preference for next-item recommendation. In *WSDM*.
[93] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to do next: Modeling user behaviors by time-LSTM. In *IJCAI*.
[94] Konrad Żołna and Bartłomiej Romański. 2017. User modeling using LSTM networks. In *AAAI*.

Received May 2020; revised September 2020; accepted October 2020