A Generative Framework for Personalized Sticker Retrieval

Anonymous ACL submission

Abstract

Formulating information retrieval as a variant of generative modeling, specifically using autoregressive models to generate relevant identifiers for a given query, has recently attracted considerable attention. However, its application to personalized sticker retrieval remains largely unexplored and presents unique challenges: existing relevance-based generative retrieval methods typically lack personalization, leading to a mismatch between diverse user expectations and the retrieved results. To address this gap, we propose PEARL, a novel generative framework for personalized sticker retrieval, and make two key contributions: (i) To encode user-specific sticker preferences, we design a representation learning model to learn discriminative user representations. It is trained on three prediction tasks that leverage personal information and click history; and (ii) To generate stickers aligned with a user's query intent, we propose a novel intent-aware learning objective that prioritizes stickers associated with higher-ranked intents. Empirical results from both offline evaluations and online tests demonstrate that PEARL significantly outperforms state-of-the-art methods.

1 Introduction

With the rise of instant messaging applications, online chatting has become an integral part of daily communication. Stickers, as expressive visual elements commonly used on platforms such as WeChat and WhatsApp, play a crucial role in conveying emotions and sentiments. As users increasingly rely on stickers to express themselves, personalized sticker retrieval becomes crucial for retrieving stickers that match users' unique communication styles and emotional preferences (Konrad et al., 2020; Chee et al., 2025).

Using generative modeling for sticker retrieval. Generative retrieval (GR) is an emerging paradigm in information retrieval (Tay et al., 2022), where the entire corpus is encoded into model parameters, enabling a single parametric model to directly generate a ranked list of results. Typically, a sequence-to-sequence (Seq2Seq) encoder-decoder architecture is employed to predict the identifiers of documents relevant to a given query. Recent studies have demonstrated impressive performance across various retrieval tasks, e.g., passage retrieval and image retrieval (Zhang et al., 2024, 2018; Tang et al., 2023; Long et al., 2024).

However, directly applying existing relevant-based GR methods to personalized sticker retrieval poses unique challenges: (i) *Different users prefer different stickers.* Personalized sticker retrieval should incorporate user-specific information, e.g., personal portraits and historical preferences, rather than relying solely on query-sticker semantic associations as in existing GR methods. For instance, given the query "Hello", younger users may prefer lively, animated stickers, while older users may favor more restrained or text-based ones. (ii) *A single user's preference for sticker properties varies with intent.* This calls for intent-aware ranking that aligns with the user's preferences across different sticker properties—be it character IP, visual style, or textual content. For example, for the query "Doraemon sleeping", sticker properties related to the Doraemon character should be prioritized. In contrast, for "good morning", textual content extracted via OCR may be more important.

A personalized sticker retriever. Our goal is to develop an effective *PErsonalized-learner for generAtive sticker RetrievaL* (PEARL), that can bridge the gap between diverse user expectations and the relevant stickers retrieved by generative modeling. To this end, we need to resolve two key challenges in terms of encoding and decoding.

First, *How to encode user-specific preferences effectively?* In this work, we consider that user-specific preferences are mainly determined by the

user's age and gender, as well as historical click-through data. In GR, generating document identifiers using dense document representations has been proven effective (Zhou et al., 2022; Li et al., 2024). However, user-specific information has not been adequately considered in existing studies. To address the issue, we first categorize users based on their age and gender into distinct user groups, and then for each user group, we design a discriminative representation learning model that captures the unique characteristics of the user group. Specifically, three tasks, including user click prediction, user intent prediction and user interest prediction, are involved in the representation learning of the user group using data in the history click log: Subsequently, the user group representation is input into the generative model along with the user query for personalized encoding.

Second, *How to decode stickers that align with individual expressive intent?* A sticker typically involves multiple properties, such as character IP, OCR textual content, visual style, entity, and meaning. We first generate a product quantization (PQ) code for each property of a given sticker as its property identifier (Zhou et al., 2022). Accordingly, the objective of the GR model is to generate each property identifier of the corresponding stickers for a given input query. We propose an *intent-aware loss* that reweights the relevance between the input query and different property identifiers based on inferred user intent. To infer user intent, we leverage the chain-of-thought (CoT) reasoning capabilities of large language models (LLMs) (Yu et al., 2023) to determine the intent ranking of the query with respect to each property dimension. The intent-aware loss is designed to ensure that the property identifiers corresponding to higher-ranked intents receive greater attention.

Experiments and contributions. The effectiveness of PEARL is verified by extensive offline analyses and large-scale online tests. PEARL significantly outperforms state-of-the-art methods, particularly in MRR@10 and Recall@10, with substantial improvements of 15% and 18.3%, and additionally achieves CTR improvements and GSB gains of 7.12% and 5.98% against the online system under the evaluation of human experts.

2 Problem Statement

Task description. Given a textual input query q, the objective of sticker retrieval is to yield a ranked list R of top-k relevant stickers from a large sticker repository $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, where n denotes the total number of stickers in the repository.

As one of the most popular instant messaging platforms, WeChat is a representative application scenario of sticker retrieval (Zhou et al., 2017). During our investigation of sticker retrieval in WeChat, five properties of stickers are considered in this work, including: (i) *OCR textual content* o refers to the text extracted from the sticker using Optical Character Recognition (OCR) technology. (ii) *Character IP* c refers to Intellectual Property (IP) related to the characters depicted on the sticker, which could be a well-known character from a movie, TV show, comic book, video game, or any other form of media. (iii) *Entity* e refers to the specific object, symbol, or concept that is primarily depicted in the sticker. (iv) *Visual style* v refers to the specific artistic style that the sticker's design follows. (v) *Meaning* m refers to the intended message, sentiment, or symbolism that the sticker is designed to convey, which is typically provided by the source of the sticker. A detailed example of these properties is provided in Appendix D.

User-specific personalization in sticker retrieval. User-specific personalization refers to the process of retrieving stickers based on user-specific information beyond general relevance. Generally, the definition of user-specific personalization can vary across different sticker retrieval systems. In this work, based on our investigation in WeChat, we focus primarily on the personalization induced by age a, gender g, and historical interest in character IPs H_c and entities H_e. We further categorize users based on age and gender, denoted as *user groups*, and a user with age a and gender g is allocated into the user group $G_{a,g}$.

Benchmark construction. In this work, we involve two sticker repositories at different scales. (i) *WeChat offline dataset*. We construct the WeChat offline dataset by sampling partial stickers from the WeChat online system. We enlisted human annotators for the annotation of the training and test datasets, as well as the collection of click logs with permission. Refer to Appendix A for detailed elaboration. (ii) *WeChat online dataset*. We also assess retrieval performance on the online large-scale sticker repository with millions of stickers, using the internal platform of WeChat.

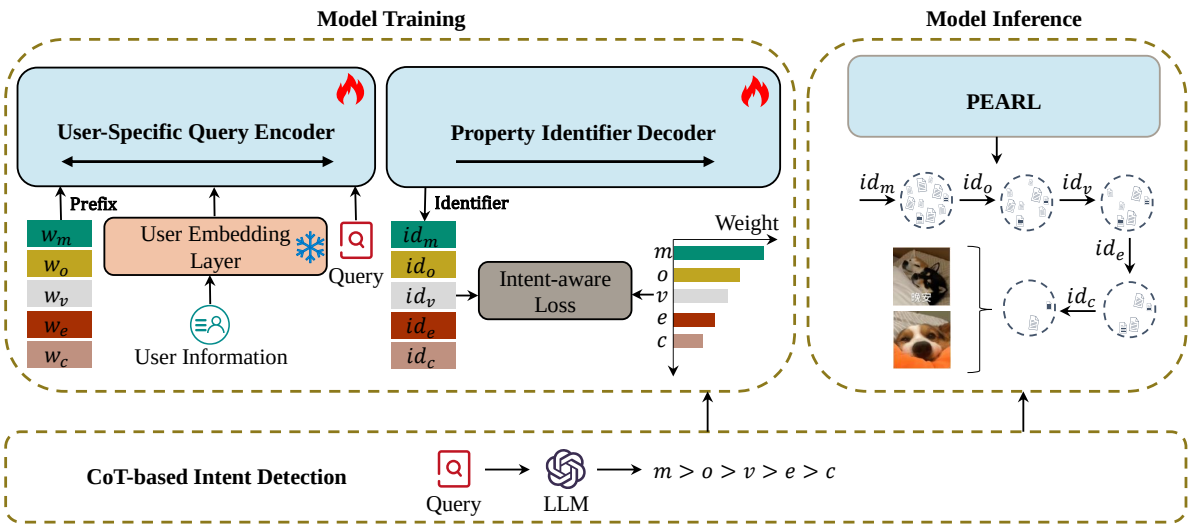

Figure 1: The overview of PEARL.

3 Methodology

In this section, we present the proposed PEARL for personalized sticker retrieval in detail.

3.1 Overview

The proposed PEARL framework employs an encoder-decoder generative architecture: the encoder encodes the user-specific information and the query; the decoder decodes property identifiers to retrieve specific stickers. To capture user-specific information, personalized representation learning is proposed to assign a unique dense embedding for each user group. To align the decoding process with user intent, intent-aware loss is proposed, guiding the process of property identifier generation with user intent predicted by LLMs. The overview of PEARL is shown in Figure 1.

3.2 Model Architecture

The architecture of PEARL comprises a user-specific encoder and a property identifier decoder.

User-specific query encoder. The user-specific query encoder maps user-specific information involving the age a and gender g along with the input query $q = \{w_1, w_2, \ldots, w_{|q|}\}$ into a compact hidden state representation, formulated as follows:

$$H_q = \text{Encoder}(w_{a,g}, w_1, w_2, \ldots, w_{|q|}), \quad (1)$$

where H_q denotes the hidden state representation, and $w_{a,g}$ is a user-specific special token added to the vocabulary to represent the specific user group $G_{a,g}$ categorized by age a and gender g. To align the semantic representation of each user-specific token $w_{a,g}$ with user preferences, personalized representation learning is utilized to train the embedding of user-specific tokens, as presented in Section 3.3.

Property identifier decoder. Given the encoded representation H_q, the property identifier decoder is intended for yielding the property identifier of the target stickers. Specifically, the probability of generating the n-th token w_n in the target identifier of the property $p \in \{o, c, e, v, m\}$ is defined as:

$$P(w_n | w_{<n}, q, a, g, p) = \text{Decoder}(w_{<n}, H_q, w_p), \quad (2)$$

where w_p is a special token indicating the identifier start of the property p. The identifier construction is introduced as follows.

Sticker identifier. Since each sticker has multiple properties, we propose representing each sticker with multiple identifiers corresponding to its different properties. For property identifier construction, we apply semantic-based property identifiers through Product Quantization (PQ) (Zhou et al., 2022). For all D-dimensional vectors, PQ first partitions the D-dimensional space into m disjoint subspaces. Subsequently, k-means clustering is independently applied to each subspace to obtain k cluster centroids per group. Each vector is ultimately represented by a sequence of m cluster identifiers, corresponding to the nearest centroids in each subspace. More details on PQ refer to Appendix C. We leverage BERT (Devlin et al., 2019) to encode the property p and then the identifier of each property for a specific sticker is defined as:

$$id_p = \text{PQ}(\text{BERT}(p)), \quad p \in \{o, c, e, v, m\}, \quad (3)$$

where multiple property identifiers id_p with respect

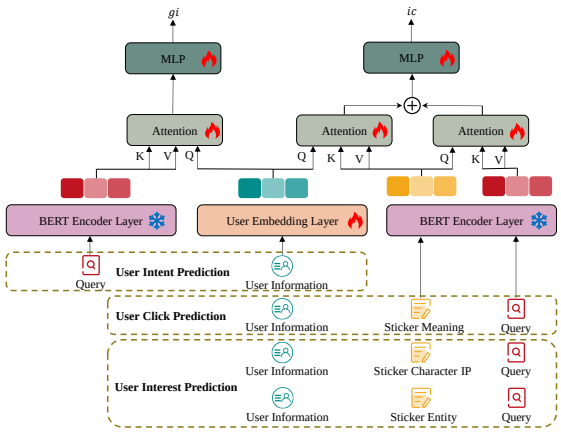

Figure 2: The learning of user-specific representation.

to a specific sticker are treated as new tokens and added to the vocabulary.

At the inference time, the constrained beam search strategy is utilized to limit each generated identifier within a pre-defined candidate set. The order in which different property identifiers are decoded is guided by the intent contained in the query, as in Section 3.4.

3.3 Personalized representation learning

As shown in Figure 2, we leverage additional data from user click logs for personalized representation learning, trained with three discriminative tasks. The training data for personalized representation learning is sampled from the user click logs dumped from the online sticker search system. Apart from the metadata of stickers, i.e., $\{o, c, e, v, m\}$ and the user-specific information $\{a, g, H_c, H_e\}$, user logs additionally involve the input query q and the click behavior ic (is clicked) which indicates whether the user clicks the sticker.

For the description of three tasks, we first ouline the used attention mechanism. Given the input hidden state $h^q, h^k, h^v \in \mathbb{R}^d$, the attention mechanism $A(\mathbf{h}^q, \mathbf{h}^k, \mathbf{h}^v)$ can be formulated as:

$$A(\cdot) = \text{softmax}\left(\frac{W^q\mathbf{h}^q \cdot W^k\mathbf{h}^k}{\sqrt{d}}\right) W^v\mathbf{h}^v, \quad (4)$$

where $W^{(\cdot)} \in \mathbb{R}^d$ are trainable projection matrices.

User click prediction. The core idea is to predict whether the user will click a specific sticker after sending the query. This task directly captures the understanding of users in terms of the relevance of the query and the meaning of stickers, formulated as a binary classification task:

$$h_q = A(\text{BERT}(q), \text{BERT}(m), \text{BERT}(m)), \quad (5)$$

$$h_u = A(\text{UE}(w_{a,g}), \text{BERT}(m), \text{BERT}(m)), \quad (6)$$

$$ic = \text{MLP}(\text{concat}(h_q, h_u)), \quad (7)$$

where UE denotes the user embedding layer.

User intent prediction. The core idea is to predict the intent preference of users hidden in the input query. LLMs are employed to obtain the golden intent gi for a query q, and the prompting strategy is explained in Appendix B in detail. This task is formulated as a multi-label classification task:

$$h_i = A(\text{UE}(w_{a,g}), \text{BERT}(q), \text{BERT}(q)), \quad (8)$$

$$gi = \text{MLP}(h_i), \quad (9)$$

where UE denotes the user embedding layer.

User interest prediction. The core idea is to predict whether a user will be interested in a specific sticker based on the user's historical click behavior. Distinct from the query-meaning relevance, user interest is typically influenced by the character IP and the entity in the sticker. This task is motivated by the phenomenon that younger individuals tend to favor lively and trendy stickers, while older individuals lean towards more conservative and accessible options (Konrad et al., 2020). For the character IP interest c, the task can be formulated as follows:

$$h_q = A(\text{BERT}(q), \text{BERT}(c), \text{BERT}(c)), \quad (10)$$

$$h_u = A(\text{UE}(w_{a,g}), \text{BERT}(c), \text{BERT}(c)), \quad (11)$$

$$ic = \text{MLP}(\text{concat}(h_q, h_u)), \quad (12)$$

where UE denotes the user embedding layer. A similar user interest prediction task is constructed for the entity e.

Learning. The user embedding of $w_{a,g}$ is learned by jointly optimizing the aforementioned three modules with maximum likelihood estimation (MLE). The learned embedding of the special token $w_{a,g}$ is retained frozen for subsequent application in the generative retrieval framework.

3.4 Intent-aware model training

CoT-based intent detection. Given the input query q, we utilize the CoT capability of LLMs to determine the intent ranking with respect to each property dimension. Specifically, (i) we first prompt the LLM to perform the intent detection task by providing the introduction of different properties in $\{o, c, e, v, m\}$ with some examples. (ii) we then construct a question-answer pair that formats the

LLM output: In the question part, we provide a specific query example. In the answer part, we provide the reasoning process that iteratively prioritizes and explains the intent with the highest probability from the intent remaining set, discarding each selected intent until none remain. A specific prompt applied in our implementation is provided in Appendix B.

By prompting LLMs in the CoT manner, a ranked list of intended properties $\mathcal{R}$ can be yielded for each query. The intent detection strategy is applied to queries in both the test set and the training set, aiming to enhance the consistency between training and inference of GR models.

Model training: indexing. The target is to memorize the information about each specific sticker. In this phase, the metadata within each sticker is indexed into the model parameters by mapping each property content to the property identifier, i.e.,

$$\mathcal{L}_I = -\sum_{i=1}^{n} \sum_{p \in \{o,c,e,v,m\}} \log(P_\theta(id_{p_i}|w_p, p_i)), \tag{13}$$

where n denotes the number of stickers in the corpus and w_p is a special prefix token indicating which property identifier to generate.

Model training: retrieval. Labeled training data involving user-query-sticker triplets is further utilized for the integration of personalized user information. After acquiring the ranked list of intended properties $\mathcal{R}$ for queries in the training set, we propose an intent-aware loss to reweight the relevance between the input query and different property dimensions. The core idea is to prioritize stickers with higher-ranked intents. Suppose each user-query-sticker triplet contained in the training dataset $\mathcal{T}$ is $\tau = (G_{a,g}, q, s_i)$, the optimization objective can be formulated as:

$$\mathcal{L}_R = -\sum_{\tau \in \mathcal{T}} \sum_{p \in \mathcal{R}} d_p \log(P_\theta(id_{p_i}|w_p, w_{a,g}, q)), \tag{14}$$

where w_p is a special prefix token indicating which property identifier to generate. The decay weight d_p is defined as:

$$d_p = \frac{1}{\log_2(\text{rank}(p) + 1)}, \tag{15}$$

where rank$(.)$ returns the intent rank within $\mathcal{R}$.

The GR model is learned by jointly optimizing the indexing loss and the retrieval loss, and the total loss $\mathcal{L}_T$ can be formulated as follows:

$$\mathcal{L}_T = \mathcal{L}_I + \mathcal{L}_R. \tag{16}$$

Model inference. Given a test query q, the model inference phase is guided by the ranked list of intended properties $\mathcal{R}$. (i) First, we construct an initial prefix tree for each intent, i.e., T_o, T_c, T_e, T_v, T_m, using property identifiers that span across all stickers. (ii) When processing the i-th intent p in the intent list $\mathcal{R}$, we perform constrained beam search during decoding on the prefix tree T_p to obtain a series of property identifiers, which correspond to a collection of stickers $\mathcal{S}_i$. (iii) We filter $\mathcal{S}_i$ by removing the stickers which do not appear in $\mathcal{S}_{i-1}$. (iv) This process is iteratively repeated until all intents in $\mathcal{R}$ have been processed, resulting in the final collection of target stickers $\mathcal{S}_{|\mathcal{R}|}$. With intent aware, the model inference process is performed in a funnel-like manner, transitioning from a coarse-grained to a fine-grained focus.

4 Experimental Settings

Implementation details. BERT corresponds to the pre-trained `bert-base-chinese`[1]. We adopt `bart-large`[2] as the encoder-decoder backbone of PEARL. We employ `deepseek-chat`[3] for CoT-based intent detection. For PQ, the number of subspaces m is 8, and the number of clusters k is 256. During inference, we set the beam size to 10 and maximum decoding steps to 15. Refer to Appendix G for more implementation details.

Evaluaion metrics. We adopt two evaluation metrics: (i) *Mean reciprocal rank (MRR@k)* measures the relative ranking position of positive stickers. We use MRR@{1,5,10,20} in our settings. (ii) *Recall@k* measures whether positive stickers are ranked in the top-k candidate list. We use Recall@{1,5,10,20} in our settings.

Baseline methods. We compare PEARL's retrieval effectiveness with four categories of representative methods: (i) *Popularity-based methods*: Global Popularity (GPop) that returns the most popular stickers globally and User Group Popularity (UPop) that independently returns the most popular stickers for each user group. The popularity is obtained from the online click log statistics of the WeChat system. (ii) *Traditional retrievers*: BM25 (Steck, 2011), DPR (Karpukhin et al., 2020) and ANCE (Xiong et al., 2020).

[1]https://huggingface.co/google-bert/bert-base-chinese
[2]https://huggingface.co/facebook/bart-large
[3]https://www.deepseek.com/

	Model	MRR				Recall			
		@1	@5	@10	@20	@1	@5	@10	@20
Popularity-based	GPop	0.0029	0.0069	0.0069	0.0069	0.0002	0.0012	0.0012	0.0012
	UPop	0.0231	0.0308	0.0315	0.0319	0.0024	0.0055	0.0061	0.0067
Tradtional	BM25	0.0519	0.0719	0.0783	0.0826	0.0049	0.0195	0.0282	0.0486
	DPR	0.0778	0.1175	0.1314	0.1385	0.0087	0.0256	0.0486	0.0705
	ANCE	0.0823	0.1293	0.1454	0.1478	0.0172	0.0328	0.0592	0.0793
Cross-modal	CN-CLIP	0.0375	0.078	0.0798	0.08	0.0046	0.0198	0.0223	0.0228
	StickerCLIP	0.0528	0.0821	0.0842	0.0892	0.0052	0.0203	0.0235	0.0248
	PerSRV	0.1061	0.1328	0.1401	0.1496	0.0129	0.0318	0.0476	0.0617
Generative	DSI	0.0029	0.0079	0.0079	0.0079	0.0002	0.001	0.0011	0.001
	DSI-QG	0.0000	0.0033	0.0048	0.0062	0.0	0.0018	0.0028	0.0084
	GENRE	0.0317	0.0512	0.0539	0.0543	0.0039	0.0104	0.0143	0.0152
	MINDER	0.1327	0.1699	0.1804	0.1987	0.0167	0.0492	0.0594	0.0703
	PEARL	**0.1547***	**0.1839***	**0.2074***	**0.2143***	**0.0288***	**0.0582***	**0.0732***	**0.0835***

Table 1: Retrieval performance of PEARL and the baselines on the WeChat offline dataset. $*$ indicates statistically significant improvements over the best performing baseline MINDER ($p \leq 0.05$).

(iii) *Cross-modal retrievers*: CN-CLIP (Yang et al., 2022), StickerCLIP (Zhao et al., 2023), and PerSRV (Chee et al., 2025). (iv) *Generative retrievers*: DSI (Tay et al., 2022), DSI-QG (Zhuang et al., 2022), GENRE (De Cao et al., 2020), Ultron (Zhou et al., 2022) and MINDER (Li et al., 2023).

Model variants. To validate the effectiveness of each components in PEARL, we implement the following variants to facilitate ablation studies: (i) PEARL$_{-UE}$ removes the user embedding from the framework and ignores variability in queries from different user groups. (ii) PEARL$_{click}$ only employs the task of user click prediction in Section 3.3 to train the user embedding. (iii) PEARL$_{intent}$ only employs the task of user intent prediction in Section 3.3 to train the user embedding. (iv) PEARL$_{interest}$ only employs the task of user interest prediction in Section 3.3 to train the user embedding. (v) PEARL$_{-IAL}$ removes the intent-aware loss in Section 3.4 during the model training phase. (vi) PEARL$_{-IG}$ removes the intent-guided docid decoding process in Section 3.4 during the model inference phase and considers the intent of the user query to be equivalent.

5 Experimental Results

5.1 Main results

Table 1 shows the comparison of PEARL and baselines on the WeChat dataset.

Popularity-based methods. We find that: (i) UPop, which independently returns the most popular stickers for each user group, exhibits superior retrieval capability than GPop, which neglects the differences between user groups. The phenomenon highlights the importance of preference differences among different user groups. (ii) PEARL significantly outperforms popularity-based methods. The underlying reason is that popularity-based methods focus exclusively on the popularity of stickers while neglecting the relevance between queries and stickers.

Traditional retrievers. When it comes to traditional retrievers including BM25, DPR and ANCE, PEARL outperforms all traditional retrievers in terms of retrieval performance. The underlying reason might be that PEARL models user preferences into generative models instead of simply relying on relevance between queries and stickers.

Cross-modal retrievers. We can conclude as follows: (i) Although a new image modality is introduced, cross-modal retrievers do not demonstrate the anticipated improvement in retrieval performance. In fact, the performance of cross-modal retrievers lags behind that of text-based dense retrievers. The underlying reason might be that the image modality of stickers tends to be diverse and expressive, hence posing significant challenges and difficulties for modal alignment. (ii) PEARL and PerSRV both model user preference for stickers, and PEARL exhibits superior retrieval performance. We attribute the phenomenon to the fact that apart from modeling user preference for stickers, PEARL further mines user intent behind queries, leading to more specific personalization.

Generative retrievers. When we look at generative retrievers, we can find that: (i) Approaches applying multi-view docids, including MINDER and PEARL, significantly outperforms other methods utilizing either naive string docids (DSI and DSI-QG) or meaning-based single-view docids (GENRE). (ii) PEARL outperforms all other generative baselines. The underlying reason might be

Model	MRR@10	Recall@10
PEARL	0.2074	0.0732
w/o personalized user embedding		
PEARL$_{-UE}$	0.1497	0.0463
PEARL$_{click}$	0.1639	0.0585
PEARL$_{intent}$	0.1563	0.0518
PEARL$_{interest}$	0.1838	0.0614
w/o intent-aware loss		
PEARL$_{-IAL}$	0.1863	0.0638
w/o intent guidance		
PEARL$_{-IG}$	0.1782	0.0575

Table 2: Ablation study on the WeChat offline dataset.

that the personalized representation learning and the intent-aware model training are devised tailor for personalized sticker retrieval.

5.2 Ablation studies

To further validate the effectiveness of each module in PEARL, we conduct ablation studies and report the retrieval performance of model variants in Table 2. The following conclusions can be drawn: (i) The proposed personalized user embedding demonstrates the most significant contribution to retrieval effectiveness, followed by intent guidance during the inference phase, and subsequently by the incorporation of intent-aware loss during the training phase. This highlights that sticker retrieval is an expressive and fuzzy retrieval task which relies on not only the relevance relationship between queries and stickers but also the user preference. (ii) The user interest prediction task contributes most to personalized representation learning. This phenomenon illustrates that user preference for stickers primarily focuses on the preference for Character IPs and entities.

5.3 Efficiency analysis

We compare the efficiency of DPR, MINDER, and PEARL. Note that the intent list of queries is pre-computed in PEARL. Refer to Appendix G for more details. As depicted in Table 3, (i) Generative retrievers, i.e., MINDER and DPR, have a significant reduction of memory footprint and inference time compared to the dense retrieval model DPR. The reduction of memory footprint primarily lies in the elimination of the explicit document index, and the inference time decreases since the heavy retrieval process over the large-scale dense index is replaced with a light generative process over the prefix tree. (ii) Compared to MINDER, PEARL requires longer inference time due to the addition of the intent-aware funnel-like decoding process. However, we believe that such an efficiency sacrifice is worthwhile, as PEARL achieves significant effectiveness gains compared to MINDER according to Table 1.

Model	Memory	Parameter	Time
DPR	3.6G	110M	179ms
MINDER	1.6G	406M	112ms
PEARL	1.6G	406M	124ms

Table 3: Comparisons on the memory, the number of model parameters and inference time per query.

5.4 Online tests

User preferences for stickers are highly subjective, hence the annotation of the golden truth data is usually incomplete in the sticker retrieval task. To this end, we conduct an online test to further verify the effectiveness of our method. It is worth noting that, due to privacy issues, the online WeChat system we compare is a variant that turns off personalization at the individual user's granularity.

Evaluation. We compare PEARL to online WeChat systems at both the sticker and the session level for a more holistic and fair assessment.

For the sticker-level assessment, we assess PEARL and online systems with the Team-Draft Interleaving (TDI) process (Schuth et al., 2015). The specific procedures are as follows: (i) At the start of each query session, a fair Bernoulli trial decides which system—PEARL or the online system—drafted the first sticker. (ii) The active drafter appended its next unseen sticker to the interleaved list, after which drafting control immediately passed to the other system. (iii) Drafting continued in strict alternation until both original top-10 lists were exhausted, resulting in a 20-item interleaved ranking. (iv) Every position in the final list was annotated with a binary ownership label, thereby enabling later attribution of each user click to its originating system. The procedure preserved each model's internal order, and the ownership of returned stickers is completely blind to users to ensure the fairness of comparison. Twenty human experts of different ages and genders are chosen to enter queries and perform clicking behavior, leading to 1,000 valid queries. The evaluation metrics in the sticker-level assessment are two-fold: ΔCTR and ΔACP, refer to Appendix E for a de-

ΔCTR ↑	ΔACP ↓	ΔGSB ↑
+7.12%	-0.19	+5.98%

Table 4: Comparison with the online WeChat system.

tailed introduction of the metrics.

For the session-level assessment, we show the exposure session returned by PEARL and the online system containing the top-10 stickers, without allowing the user to know which model the exposure page was derived from. We subsequently ask the users to make an overall assessment of the preference for the exposure sessions, which is limited to three responses: *preferring the left exposure session*, *preferring the right exposure session*, and *preferring both equally*. Here, we measure the relative gain with ΔGSB, refer to Appendix E for a detailed introduction of the metric. Twenty human experts of different ages and genders are chosen to enter queries and assess preference for exposure sessions, resulting in 1,000 valid queries.

Experimental results. As depicted in Table 4, compared to the results returned by the online system, PEARL increases the click-through-rate by 7.12% and decrease the average-click-position by 0.19 in the sticker-level human expert evaluation. Furthermore, we can also find that PEARL has achieved significant positive gains in terms of session-level assessment.

Case study. Figure 3 shows the list of the top-5 stickers returned by the online system and PEARL, and the statistics of these users' clicking behavior. Our method returns stickers that are more clicked for the user query "Bye-bye" by female users aged 20–30. More cases refer to Appendix F.

6 Related work

Sticker retrieval. Stickers have gained significant popularity due to their ability to convey emotions, reactions, and nuanced intentions that are difficult to express through plain text (Zhao et al., 2023). To retrieve satisfactory stickers for users, Liang et al. (2024) proposed a framework dubbed Int-RA based on the learning of intention and the cross-modal relationships between conversation context and stickers. Zhao et al. (2023) first adapted the CLIP (Radford et al., 2021) model tailored for the domain of emotive stickers. Most recently, PerSRV (Chee et al., 2025) first focused on personalized sticker retrieval and introduced user preference modeling by style-based personalized ranking. Despite previous

Figure 3: Case study on retrieved results of online system and PEARL.

efforts, personalized sticker retrieval has not benefited from generative models, which have triggered transformative shifts in various areas.

Generative retrieval. Generative retrieval (GR) is a new retrieval paradigm in which a single consolidated model is employed to enable the direct generation of relevant docids from queries. To achieve this, two primary procedures are involved (Tay et al., 2022; Chen et al., 2022; Bevilacqua et al., 2022), i.e., the indexing process and the retrieval process. The indexing process learns the relationship between documents and the corresponding docids. The retrieval process maps queries to relevant docids. To model personalized user preference in generative retrieval, Wu et al. (2024) proposed an efficient hierarchical encoding-decoding generative retrieval method for large-scale personalized E-commerce search systems. Distinct from personalized E-commerce search, which typically involves specific items, the task of personalized sticker retrieval primarily focuses on the abstract expressive intent of stickers and user preference for Character IP and sticker style. The fundamental characteristics of stickers highlight that Personalized generative retrieval tailored for stickers is a non-trivial challenge worth exploring.

7 Conclusion

In this paper, we focus on personalized sticker retrieval with the promising generative retrieval paradigm. Since the sticker retrieval task highly calls for user personalization beyond relance relationships, we propose PEARL, a novel generative framework with user-specific information encoding and intent-aware sticker decoding. Empirical results from both offline evaluations and online experiments indicate the superiority of PEARL.

Limitations

The limitations of this work can be concluded as follows: (i) Given the importance of individual privacy, our focus is primarily on personalization at the level of user groups. This approach, however, offers a relatively coarse granularity that does not allow for the customization of sticker search and recommendations based on each individual's specific sticker preferences. (ii) For search efficiency considerations, we model only the textual information in PEARL without modeling the information of image modality. The introduction of image modality has the potential to further enhance the retrieval. (iii) The generative framework PEARL is coupled to the scenario of sticker retrieval, hence leading to restricted method generalizability. (iv) The application of LLMs for intent detection increases economic costs, restricting the large-scale industry applications.

Ethical Considerations

In this paper, all the models used in our experiment are publicly released. For datasets, we construct offline datasets based on the open-source dataset and extra manual annotation. We invite human annotators for manual annotation and pay the annotators a salary that is in line with the local pay scale. In this process, user privacy is protected, and no personal information is contained in the dataset. Additionally, the methods we propose aim to enhance the effectiveness and personalization of sticker retrieval and do not encourage or induce the model to produce any harmful information or leakage of user privacy. Therefore, we believe that our research work meets the ethics of ACL.

References

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35:31668–31683.

Heng Er Metilda Chee, Jiayin Wang, Zhiqiang Guo, Weizhi Ma, and Min Zhang. 2025. Persrv: Personalized sticker retrieval with vision-language model. In *Proceedings of the ACM on Web Conference 2025*, pages 293–303.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2022. Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 191–200.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Artie Konrad, Susan C Herring, and David Choi. 2020. Sticker and emoji use in facebook messenger: Implications for graphicon change. *Journal of Computer-Mediated Communication*, 25(3):217–235.

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2024. From matching to generation: A survey on generative information retrieval. *ACM Transactions on Information Systems*.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview identifiers enhanced generative retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6636–6648.

Bin Liang, Bingbing Wang, Zhixin Bai, Qiwei Lang, Mingwei Sun, Kaiheng Hou, Lanjun Zhou, Ruifeng Xu, and Kam-Fai Wong. 2024. Reply with sticker: New dataset and model for sticker retrieval. *arXiv preprint arXiv:2403.05427*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

Xinwei Long, Jiali Zeng, Fandong Meng, Zhiyuan Ma, Kaiyan Zhang, Bowen Zhou, and Jie Zhou. 2024. Generative multi-modal knowledge retrieval with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18733–18741.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Anne Schuth, Katja Hofmann, and Filip Radlinski. 2015. Predicting search satisfaction metrics with interleaved comparisons. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 463–472.

Harald Steck. 2011. Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 125–132.

Yubao Tang, Ruqing Zhang, Jiafeng Guo, Jiangui Chen, Zuowei Zhu, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng. 2023. Semantic-enhanced differentiable search index inspired by learning strategies. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4904–4913.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Yanjing Wu, Yinfu Feng, Jian Wang, Wenji Zhou, Yunan Ye, Rong Xiao, and Jun Xiao. 2024. Hi-gen: Generative retrieval for large-scale personalized e-commerce search. *arXiv preprint arXiv:2404.15675*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

An Yang, Junshu Pan, Junyang Lin, Rui Men, Yichang Zhang, Jingren Zhou, and Chang Zhou. 2022. Chinese clip: Contrastive vision-language pretraining in chinese. *arXiv preprint arXiv:2211.01335*.

Zihan Yu, Liang He, Zhen Wu, Xinyu Dai, and Jiajun Chen. 2023. Towards better chain-of-thought prompting strategies: A survey. *arXiv preprint arXiv:2310.04959*.

Kun Zhang, Guangyi Lv, Le Wu, Enhong Chen, Qi Liu, Han Wu, and Fangzhao Wu. 2018. Image-enhanced multi-level sentence representation net for natural language inference. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 747–756. IEEE.

Yidan Zhang, Ting Zhang, Dong Chen, Yujing Wang, Qi Chen, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, Fan Yang, et al. 2024. Irgen: Generative modeling for image retrieval. In *European Conference on Computer Vision*, pages 21–41. Springer.

Sijie Zhao, Yixiao Ge, Zhongang Qi, Lin Song, Xiaohan Ding, Zehua Xie, and Ying Shan. 2023. Sticker820k: Empowering interactive retrieval with stickers. *arXiv preprint arXiv:2306.06870*.

Rui Zhou, Jasmine Hentschel, and Neha Kumar. 2017. Goodbye text, hello emoji: Mobile communication on wechat in china. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 748–759.

Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. 2022. Ultron: An ultimate retriever on corpus with a model-based indexer. *arXiv preprint arXiv:2208.09257*.

Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128*.

Appendix

A WeChat offline dataset

We constructed a sticker corpus by sampling about 1.1 million stickers from the WeChat online system. Offensive, potentially harmful, and copyright-controversial stickers were filtered out. Specifically, the sticker corpus contains 1,092,122 stickers spanning 17,906 Character IPs, 38,895 entities, and 107 visual styles. Based on the actual usage of the sticker search function, we categorized users into four age groups (0-19,20-29,30-44, and 45-59) and two gender groups (male and female). We enlisted human annotators across all these user groups. We collect the user click logs with their permission and invite them to perform data annotation for both the training and test datasets. Specifically, the training dataset contains 1,891 unique queries, 2,308 user-query pairs, and 12,568 user-query-sticker triplets. The test dataset contains 258 unique queries, 347 user-query pairs, and 14,446 user-query-sticker triplets. The full text of the instructions for annotating the training and the test datasets given to participants is as follows: `Determine whether a given query and sticker match based on your personal preferences by selecting either "Match" or "No Match". The data collected will only be used to carry out research to improve the effectiveness of sticker retrieval. In this process, user privacy is protected, and no personal information is contained in the dataset.`

We invited human annotators from the crowdsourcing platform and paid the annotators a salary that is in line with the local pay scale. Due to the limited community of WeChat software users, we enlisted all data annotators from China. The data collection protocol was approved by an ethics review board. We manually filtered all collected data to remove any user privacy information. All data used contains neither information that uniquely identifies individual people nor offensive content.

B Prompt for intent permutation generation

The prompt applied in our implementation is as follows:

```
I am a user who is using the sticker search feature, and I have entered a query. Please help me analyze the intent behind my query.
There are five possible intents: OCR, IP, entity, style, and meaning. Here are the descriptions and examples for each intent.
OCR textual content refers to the text extracted from the sticker using Optical Character Recognition (OCR) technology.
Examples: {examples for the OCR intent}
Character IP refers to Intellectual Property (IP) related to the characters depicted on the sticker, which could be a well-known character from a movie, TV show, comic book, video game, or any other form of media.
Examples: {examples for the IP intent}
Entity refers to the specific object, symbol, or concept that is primarily depicted in the sticker.
Examples: {examples for the entity intent}
Visual style refers to the specific artistic style that the sticker's design follows.
Examples: {examples for the style intent}
Meaning refers to the intended message, sentiment, or symbolism that the sticker is designed to convey, which is typically provided by the source of the sticker.
Examples: {examples for the meaning intent}
Q: Based on the given explanation, arrange the order of intent for the query: Doraemon cute.
A: Let's think step by step. "Doraemon cute" is most likely to be an IP intent in OCR, IP, entity, style, meaning, because Doraemon is a well-known anime character. Excluding the IP intent, among the remaining OCR, entity, style, meaning, "Doraemon cute" is most likely to be a style intent, because the query includes the style description "cute". Excluding IP and style intents, among the remaining OCR, entity, meaning, "Doraemon cute" is most likely to be an entity intent, because Doraemon is a specific character. Excluding IP, style, and entity intents, among the remaining OCR and meaning, "Doraemon cute" is most likely to be a meaning intent, because "Doraemon cute" can be understood as a certain meaning. "Doraemon cute" is least likely to be an
```

Sticker	OCR textual content	Character IP	Entity	Visual style	Meaning
	Thank you boss	Doraemon	Cartoon characters	Cute	Thanks
	The only thing left in my world is loneliness	Hungry crazy bunny	Rabbit	Daily	Loneliness
	May you happy and prosperous	Liu Dehua	Male	Funny	Blessing

Figure 4: Examples for distinct properties of stickers in the corpus.

```
OCR intent, because it is not an image or
video with text content. Therefore, the
answer is: IP > style > entity > meaning
> OCR.
Q: Based on the given explanation,
arrange the order of intent for the query:
{query}
A: Let's think step by step.
```

C Product quantization

Product Quantization (PQ) is an efficient technique for approximate nearest neighbor (ANN) search in high-dimensional spaces, commonly used in large-scale retrieval tasks. It works by decomposing a D-dimensional vector space into m low-dimensional subspaces, i.e., each input vector $\mathbf{x} \in \mathbb{R}^D$ is split into m sub-vectors $\mathbf{x} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^m]$, where each $\mathbf{x}^i \in \mathbb{R}^{D/m}$. In each subspace, a separate codebook is learned via k-means clustering, and sub-vectors are quantized by mapping them to their nearest centroids. The full vector is then represented as a concatenation of centroid indices, significantly reducing storage requirements. During search, the distance between a query vector and database vectors is approximated efficiently using precomputed lookup tables, enabling fast and memory-efficient similarity computation without reconstructing full vectors.

D Data examples

Detailed examples of the properties in the sticker corpus are provided in Figure 4.

E Online evaluation metircs

For the sticker-level assessment, we report the relative advantage of PEARL over the baseline with two per-query paired-difference metrics: $\Delta\mathrm{CTR}$ and $\Delta\mathrm{ACP}$.

Click-through-rate difference. For each query q, let $\mathrm{CTR}_P(q)$ and $\mathrm{CTR}_B(q)$ denote the fractions of exposed stickers that were clicked for PEARL and the baseline, respectively. The evaluation metric $\Delta\mathrm{CTR}$ is defined as

$$\Delta\mathrm{CTR} = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}(\mathrm{CTR}_P(q) - \mathrm{CTR}_B(q)), \tag{17}$$

where $\mathcal{Q}$ denotes the collections of all queries.

Average-click-position difference. Let $\mathrm{ACP}_P(q)$ and $\mathrm{ACP}_B(q)$ be the mean rank positions of the clicks attributed to each system. The evaluation metric $\Delta\mathrm{ACP}$ is defined as

$$\Delta\mathrm{ACP} = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}(\mathrm{ACP}_P(q) - \mathrm{ACP}_B(q)), \tag{18}$$

where $\mathcal{Q}$ denotes the collections of all queries. A negative value indicates that PEARL receives clicks closer to the top of the interleaved list.

Figure 5: Case study for the user query "Angry" by male users aged 0-19.

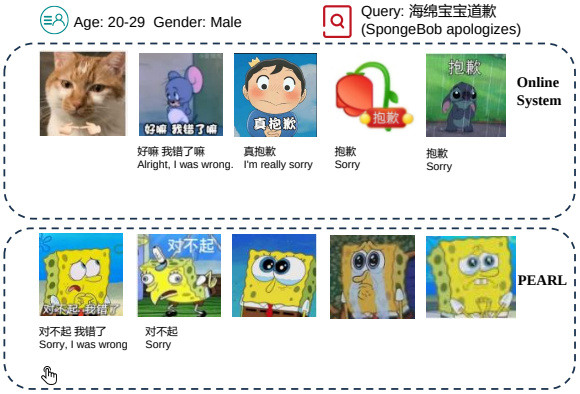

Figure 6: Case study for the user query "SpongeBob apologizes" by male users aged 20-29.

For the session-level assessment, we report the relative gain of PEARL over the baseline with the metric ΔGSB, which can be defined as follows:

$$\Delta\text{GSB} = \frac{\#Good - \#Bad}{\#Good + \#Same + \#Bad}, \quad (19)$$

where $\#Good$ (or $\#Bad$) indicates the number of queries that the PEARL provides better (or worse) final results against the baseline.

F Case study

Figure 5 and Figure 6 provide two additional cases to further illustrate the advantage of PEARL.

G Experimental details

The offline experiments are conducted on 4 × NVIDIA Tesla A100 80G GPUs. The training process of PEARL takes approximately 8 hours. We leverage the pyserini library (Lin et al., 2021) for the implementation of BM25, DPR, and ANCE. We leverage the fairseq library (Ott et al., 2019) for the training of MINDER and PEARL. We use the transformers library (Wolf et al., 2020) for the training of the remaining baselines, following the setup of the original literature. All models are trained with the AdamW (Loshchilov and Hutter, 2017) optimizer. We train PEARL with a batch size of 8192 tokens and a learning rate of 1e-5. We repeat our experiment 3 times to get the average results. To improve efficiency, we collected the top 10,000 most frequent queries from the online system for intent analysis and precomputed their corresponding intent lists offline. During the inference time of PEARL, if a user's query matches an entry in the offline table, the system retrieves the intent list directly without utilizing LLMs.

As for the evaluation of online tests, the full text of the instructions for the sticker-level assessment is as follows: `Enter a query and click your favorite sticker based on your preference.` The full text of the instructions for the session-level assessment is as follows: `Enter a query and determine which exposure session you prefer, with the response limited to "preferring the left exposure session", "preferring the right exposure session", and "preferring both equally".`