FACE: A Fine-grained Reference Free Evaluator for Conversational Recommender Systems

Anonymous ACL submission

Abstract

A systematic, reliable, and low-cost evaluation of Conversational Recommender Systems (CRSs) remains an open challenge. Existing automatic CRS evaluation methods are proven insufficient for evaluating the dynamic nature of recommendation conversations. This work proposes **FACE**: a **F**ine-grained, **A**spect-based **C**onversation **E**valuation method that provides evaluation scores for diverse turn and dialogue level qualities of recommendation conversations. FACE is reference-free and shows strong correlation with human judgments, achieving system correlation of 0.9 and turn/dialogue-level of 0.5, outperforming state-of-the-art CRS evaluation methods by a large margin. Additionally, unlike existing LLM-based methods that provide single uninterpretable scores, FACE provides insights into the system performance and enables identifying and locating problems within conversations.

1 Introduction

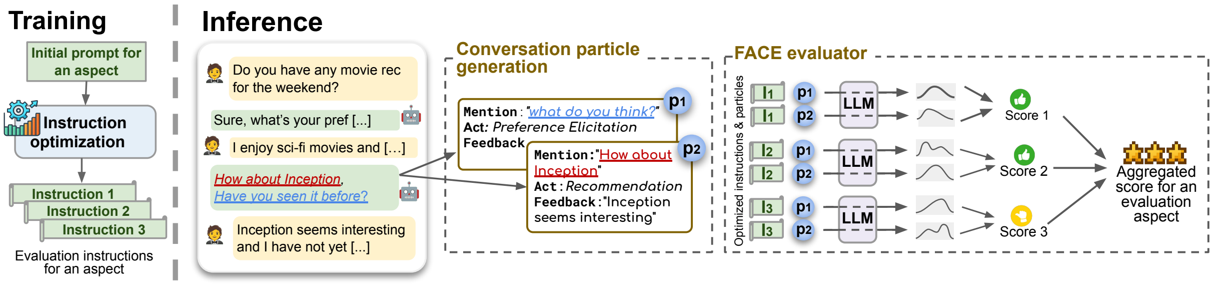
Evaluation is vital for developing powerful Conversational Recommender Systems (CRSs), where users are provided with relevant and personalized recommendations (Bernard et al., 2025; Wang et al., 2023a; Zhang et al., 2022). While human evaluation is considered the gold standard, it cannot be used intensively during the development of CRSs, due to its cost- and time-intensive nature (Zhang and Balog, 2020). Automatic evaluation methods fill this gap and serve as invaluable aids for early diagnosis of known problems and biases during development of CRSs (Dey and Desarkar, 2023; Dubois et al., 2024b)

There are a number of shortcomings in existing automatic evaluation methods that make them unreliable for CRS evaluation: (i) **Reference-based** metrics such as Recall, ROUGE-L, and BERTScore (Zhang et al., 2020) cannot capture the dynamic and evolving user-system interactions and limit the evaluation process to assessing single conversation turns given fixed conversation histories. (ii) Recently proposed reference free LLM-based evaluation methods (Liu et al., 2023; Zhong et al., 2022), while showing higher correlation with humans, provide a single **uninterpretable**[1] score for each evaluation aspect, which cannot be traced back to its contributing factors; (iii) Automatic evaluations of CRSs focus mainly on turn-level aspects (e.g., recommendation effectiveness), without providing insights into dialogue-level aspects (e.g., interest arousal and task completion), which are more indicative of why a user is (dis)satisfied (Siro et al., 2022) with a conversation.

In this paper, we propose FACE, a Fine-grained Aspect-based Conversation Evaluation method. FACE is **reference-free** and handles diverse conversation trajectories; see Figure 1. It first decomposes system responses into self-contained, contextualized information fragments, termed *conversation particles*. They are then evaluated by an LLM using a set of optimized instructions, through beam search and a bandit algorithm. These sub-scores are then aggregated into a single score per aspect. Therefore, FACE scores are **interpretable**, in the sense that they can be traced back to their contributing factors, providing insights into problems within the conversation. FACE evaluates CRSs on two **turn-level** aspects: *relevance* and *interestingness*, and five **dialogue-level** aspects: *understanding, task completion, conversation efficiency, interest arousal,* and *overall impression.*

To evaluate FACE, we collect 20,962 human annotations for 467 human-system conversations, covering the aforementioned evaluation aspects and nine diverse CRSs that are trained on the Re-

[1] In this paper, *interpretability* is less ambitious than what is defined in the AI field (Barredo Arrieta et al., 2020; Perrella et al., 2024) and concerns evaluation methods that enable humans to gain insights into a system's behavior and identify issues within conversations; i.e., interpretability of the evaluation process (Perrella et al., 2024).


Training
Initial prompt for an aspect
Instruction optimization
Instruction 1
Instruction 2
Instruction 3
Evaluation instructions for an aspect
Inference
Conversation particle generation
Mention: "what do you think?"
Act: Preference Elicitation
Feedback
p1
p2
Mention: "How about Inception"
Act: Recommendation
Feedback: "Inception seems interesting"
FACE evaluator
Optimized instructions & particles
I1 p1
I1 p2
I2 p1
I2 p2
I3 p1
I3 p2
LLM
LLM
LLM
Score 1
Score 2
Score 3
Aggregated score for an evaluation aspect

Do you have any movie rec for the weekend?

Sure, what's your pref [...]

I enjoy sci-fi movies and [...]

How about Inception, Have you seen it before?

Inception seems interesting and I have not yet [...]

Figure 1: Illustration of FACE for a turn-level aspect. Instruction optimization generates a set of diverse evaluation instructions for the given aspect (e.g., relevance) based on an initial prompt; See example instructions in Appendix E.2. For evaluation, each conversation is decomposed into particles containing a dialogue act, a mention (text span from a system turn), and user feedback from the following turn. A response distribution is created for each instruction-particle pair and the weighted summation of the scores is computed. The final score is obtained by aggregating the scores across all instructions and particles. For turn-level aspects, aggregation is performed per turn, while for dialogue-level aspects, scores of particles across the entire dialogue are aggregated.

Dial (Li et al., 2018) and OpenDialKG (Moon et al., 2019) datasets. Our experiments demonstrate that FACE outperforms state-of-the-art methods by a large margin, achieving system and turn/dialogue-level Spearman of 0.9 and 0.5, respectively, without observing any system-human conversations for instruction optimization. We further show that FACE generalizes to chatbots trained on Topical-Chat (Gopalakrishnan et al., 2019) and PersonaChat (Zhang et al., 2018) datasets, outperforming strong baselines. Importantly, we demonstrate how FACE scores can be interpreted by humans to diagnose issues of two competitive CRSs.

Key contributions of this paper include: (i) We propose FACE, a strong CRS evaluation method that evaluates dynamic user-system interactions for diverse turn- and dialogue-level evaluation aspects. (ii) We show FACE is generalizable to other LLMs and chitchat conversations, while offering interpretable scores that helps humans to identify and locate potential system issues. (iii) We develop and release a dataset for evaluating CRS evaluation methods, which covers high-quality human annotations of human-system conversations. This provides a meta-evaluation dateset that facilitate future benchmarking of CRS evaluation methods.[2]

2 Method

Our Fine-Grained Aspect-based Conversation Evaluation (FACE) approach handles the one-to-many nature of conversations and provides detailed scores for turn and dialogue-level evaluation aspect. Figure 1 illustrates the FACE method. Using beam search and bandit algorithms, FACE first optimize a set of instructions for a given evaluation aspect. During evaluation, a dialogue is decomposed into conversation particles, each containing a dialogue act (e.g., "Recommendation"), mention (e.g., "How about Inception"), and corresponding user feedback from the user response (e.g., "Inception seems interesting"). Each particle is independently evaluated with optimized instructions via an LLM, generating score distributions and resulting in turn/dialogue-level scores for a given aspect.

We note, without detailed elaboration, that FACE is applicable to a broad range of evaluation aspects (Sakai, 2023) and conversation types (e.g., task-oriented and chit-chat dialogues). The primary focus of this paper, however, is on CRSs and seven widely recognized turn- and dialogue-level evaluation aspects, following (Siro et al., 2022); see Sec. 3.1 for detailed description of these aspects.

This section describes the evaluation steps of FACE: particle generation (Sec. 2.1) and evaluation score computation (Sec. 2.2), followed by the instruction optimization process (Sec. 2.3).

2.1 Conversation Particle Generation

FACE sets two goals: (1) enable reference-free evaluations to address state explosion in natural conversation evaluation, and (2) provide fine-grained scores at both turn- and dialogue-level to locate undesired system behavior within the conversation (Sakai, 2023). To achieve these, we introduce *conversation particle*, a self-contained information unit decomposed from conversations. Each particle is composed of three parts: (i) Dialogue `act` is the system's action associated with the particle, such as "recommendation" or "preference elicitation;" (ii) `Mention` denotes the particle text within the

[2] The resources of FACE, including codes/prompts used, the interface developed, and collected annotations, will be available on the GitHub repository upon acceptance.

system's response, like "How about the movie A?"; and (iii) `Feedback` is the user's evaluative reply, for instance, "The movie A seems interesting." Following (Joko et al., 2024)), we use 5 dialogue acts: *greetings*, *preference elicitation*, *recommendation*, *goodbye*, and *others*.

We instruct an LLM to decompose system responses into particles, denoted as *decomposer* in the rest of the paper. Let r_t be the target system response at turn t, h the dialogue history preceding r_t, and r_{t+1} the user's turn following r_t. The decomposer $\mathcal{D}$ maps (h, r_t, r_{t+1}) to conversation particles $\mathbf{P}_r = \mathcal{D}(h, r_t, r_{t+1})$, where each particle $p \in \mathbf{P}_r$ is a triplet (`act`, `mention`, `feedback`). The full particle list for a dialogue, $\mathbf{P}_d$, is the union of particles from all responses with the dialogue, i.e., $\mathbf{P}_d = \bigcup_{r \in d} \mathbf{P}_r$.

Appendix E details prompts used for particle generation. It is shown that LLMs are more effective than traditional methods like dependency parsing and information extraction for decomposing texts into atomic units (Pradeep et al., 2024; Alaofi et al., 2024).

2.2 Evaluation Score Computation

FACE utilizes optimized instructions to generate scores for each conversation particle. Formally, given a particle p and the evaluation instruction I^a for the aspect a, an LLM generates a response r^a_p. To address known issues with LLM-generated scores, such as low variance and their noise (Liu et al., 2023), we obtain a response distribution $\{r^a_{p,i}\}_{i=1}^n$ and compute a weighted sum over the response set:

$$\mathcal{E}_{particle}(I^a, p) = \sum_{i=1}^{n} r^a_{p,i} P(r^a_{p,i} | I^a, p; \theta), \quad (1)$$

where $P(.)$ is a probability of $r^a_{p,i}$ from an LLM parameterized by θ.

These particles scores are then aggregated per turn or conversation by taking their mean, $\mathcal{E}(I^a, \mathbf{P}_x) = \frac{1}{|\mathbf{P}_x|} \sum_{p \in \mathbf{P}_x} \mathcal{E}_{particle}(I^a, p)$, where $\mathbf{P}_x$ is the set of particles for a given turn or dialogue, depending on evaluation aspect a; e.g., for *relevance*, aggregation is performed over particles of a turn, and for *task completion* aggregation is done for all particles of the dialogue.

A unique feature of FACE is utilizing diverse reasoning paths for each evaluation aspect, which is obtained by selecting optimized chain-of-thought (CoT) instructions. The intuition is that evaluation requires complex reasoning, and an optimal answer can be obtained by marginalizing various thought paths (Wang et al., 2023b). Here, a set of top-performing optimized instructions with various CoT instructions, $\mathbf{I}^a$, are applied to particles, and the resulting scores are aggregated to obtain the final score s^a:

$$s^a = FACE(\mathbf{I}^a, \mathbf{P}_x) = \frac{1}{|\mathbf{I}^a|} \sum_{I^a \in \mathbf{I}^a} \mathcal{E}(I^a, \mathbf{P}_x).$$

2.3 Instruction Optimization

Instruction optimizer generates diverse optimized CoT instructions for a given evaluation aspect. The goal is to obtain a representative set of thought processes (via CoT) for an evaluation aspect, and leverage them to evaluate unseen human-system conversations. The optimization process is performed on annotated human-human conversations to capture human thinking process and reasoning when assessing dialogue quality. We note, at the outset, that all the optimization and selection algorithms are performed independently for each aspect. For notational simplicity, we shall drop superscript a from aspect-related instruction and evaluation scores in this section.

To optimize instructions, we assume access to human evaluated dialogues, $\mathbf{H} = \{(x_i, l_i)\}_{i=1}^m$, where x_i is either a turn or an entire dialogue depending on the aspect, and l_i is its label. Similarly, we assume access to an LLM $\mathcal{L}_1$ that generates evaluation scores, $\mathbf{S}_\mathbf{I} = \{(x_i, \text{FACE}(\mathbf{I}, \mathbf{P}_{x_i}))\}_{i=1}^m$, where $\mathbf{P}_{\mathbf{x_i}}$ corresponds to particles of a specific turn or conversation and $\mathbf{P} = \bigcup_{x_i} \mathbf{P}_{\mathbf{x_i}}$ is all particles in the dialogue collection.

The optimization objective is to identify a set of optimal instructions $\mathbf{I}^*$ that maximizes the correlation between human labels and the scores generated by the automatic evaluator:

$$\arg\max_{\mathbf{I}^*} \mathcal{C}(\mathbf{H}, \mathbf{S}_{\mathbf{I}^*}),$$

where $\mathcal{C}(.)$ represents the correlation function.

The optimization process employs an LLM $\mathcal{L}_2$ to refine instructions based on the scores generated by the evaluator LLM $\mathcal{L}_1$. FACE employs a non-parametric optimization algorithm using textual gradients (Pryzant et al., 2023). Here, natural language "gradients" (as opposed to numerical gradients) are generated to describe the shortcomings of instructions. The gradients are used to rewrite original instructions in the opposite semantic direction. The best instructions are iteratively selected

Algorithm 1 Instruction optimization of FACE

```
Require: Human evaluations $\mathbf{H}$, initial prompt $I$, iterations $K$, beam width $b$, candidate size $b'$, gradient samples $\alpha$
 1: Initialize $\mathbf{I}^{\text{pool}} \leftarrow \emptyset$ , $\mathbf{I}_1 \leftarrow \{I\}$
 2: for $k = 1, ..., K$ do
 3:     for each instruction $I_{k_j} \in \mathbf{I}_k$ do
 4:         for each particle $p \in \mathbf{P}$ do
 5:             // Get score
 6:             $s_{p,k_j} \leftarrow \mathcal{E}_{particle}(I_{k_j}, p)$                ▷ Eq. 1
 7:             // Textual Gradient Generation
 8:             $\mathbf{G}_{p,k_j} \leftarrow \mathcal{G}_\nabla(I_{k_j}, s_{p,k_j}, l_p; \alpha)$    ▷ Eq. 2
 9:             // Instruction Rewriting
10:             $\mathbf{I}'_{p,k_j} \leftarrow \mathcal{R}_\delta(I_{k_j}, \mathbf{G}_{p,k_j})$          ▷ Eq. 3
11:         end for
12:     end for
13:     $\mathbf{I}'_k = \bigcup_{p\in\mathbf{P}} \bigcup_j \mathbf{I}'_{p,k_j}$                ▷ Collect all rewrites
14:     // Instruction Selection
15:     $\mathbf{I}^{\text{cand}}_k \leftarrow Select^{UCB}_{b'}(\mathbf{I}'_k)$              ▷ Appendix B.1
16:     $\mathbf{I}^{\text{pool}} \leftarrow \mathbf{I}^{\text{cand}}_k \cup \mathbf{I}^{\text{pool}}$                    ▷ Update pool
17:     // Select top-b instructions
18:     $\mathbf{I}_{k+1} \leftarrow \arg\max_{\mathbf{I}_k \subseteq \mathbf{I}^{\text{pool}}, |\mathbf{I}_k|=b} \mathcal{C}(\mathbf{H}, \mathbf{S}_{\mathbf{I}_k})$
19: end for
20: $\mathbf{I}^* \leftarrow \arg\max_{\mathbf{I}^* \subseteq \mathbf{I}^{\text{pool}}} \mathcal{C}(\mathbf{H}', \mathbf{S}_{\mathbf{I}^*})$
21: return $\mathbf{I}^*$
```

using beam search and Upper Confidence Bound (UCB) bandits, based on correlations with human judgments. The process consists of three stages.

(1) Textual Gradient Generation. This is an iterative process, where a static prompt ∇ is used for generating textual gradients (lines 7-8 of Algorithm 1). At each iteration k, the prompt ∇ takes an evaluation instruction I_{k_j} from the current set of instructions $\mathbf{I}_k$, its prediction score $s_{p,k_j} = \mathcal{E}_{particle}(I_{k_j}, p)$, and the corresponding human label l_p for a given particle p. A set of textual gradients $\mathbf{G}_{p,k_j}$ is then generated by:

$$\mathbf{G}_{p,k_j} = \mathcal{G}_\nabla(I_{k_j}, s_{p,k_j}, l_p; \alpha), \qquad (2)$$

where $\mathcal{G}_\nabla(.)$ is the gradient generation function, with parameter α denoting the number of gradients generated per instruction-score pair. Since human annotations are provided at the turn- or dialogue-level, l_p represents human annotation for the turn or dialogue that contains the particle p.

(2) Instruction Rewriting. This step updates each instruction using textual gradients (lines 9-10 of Algorithm 1). For each particle p at iteration k, we use the rewriting function $\mathcal{R}_\delta$ with the prompt δ, which takes the current instruction I_{k_j} and gradients $\mathbf{G}_{p,k_j}$ to obtain updated instructions $\mathbf{I}'_{p,k_j}$:

$$\mathbf{I}'_{p,k_j} = \mathcal{R}_\delta(I_{k_j}, \mathbf{G}_{p,k_j}). \qquad (3)$$

An LLM $\mathcal{L}_3$ is used for the rewriting function $\mathcal{R}_\delta$ with the prompt δ guiding it to revise the current instruction, considering the provided feedback.

(3) Instruction Selection. The step identifies the most promising instructions for the next iteration in two stages of selecting candidate instructions using UCB bandit, and identifying the top beam based on evaluation scores on the training data. This step corresponds to lines 14-18 of Algorithm 1.

Let $\mathbf{I}'_k = \bigcup_{p\in\mathbf{P}} \bigcup_j \mathbf{I}'_{p,k_j}$ be the set of all rewritten instructions at the iteration k. The first stage identifies $b' \geq b$ promising candidates instructions $\mathbf{I}^{\text{cand}}_k$ from the rewritten instructions $\mathbf{I}'_k$ and stores them in an *instruction pool* $\mathbf{I}^{\text{pool}}$ (lines 15-16 of Algorithm 1).

The second stage then evaluates these candidate instructions by computing the correlation of the generated scores with human labels using the training set. Therefore, the instructions for the next iteration are generated (line 18 of Algorithm 1. This process follows a beam search approach, where at each iteration we create a pool of candidates, assess their performance, and select the top ones to form the new beam for continued exploration. Once all iterations are complete, the final optimal instructions $\mathbf{I}^*$ are selected by their correlation scores on a validation set $\mathbf{H}'$ (line 20 of Algorithm 1).

The UCB selection algorithm, denoted as $Select^{UCB}_{b'}$ in Algorithm 1, returns the top b' candidates using UCB bandits, following Pryzant et al. (2023). It iteratively selects instructions based on the estimated correlation with human annotations computed over sampled particles; see Appendix B.1 for algorithmic details.

3 Human Annotation Collection

To assess the correlation of automatic CRS evaluation methods with human judgments, we develop a dataset and crowdsource human annotations on a set of human-system conversations.

3.1 Dialogue Annotation

To create the meta-evaluation dataset, we need human annotations on multi-turn interactions between users and various CRSs. CRSArena-Dial (Bernard et al., 2025), consists of dialogues with multi-turn interactions between users and nine state-of-the-art CRSs trained on OpenDialKG (Moon et al., 2019) and ReDial (Li et al., 2018) datasets . To ensure annotation quality, we implement strict quality control procedures and develop a specialized annotation interface through multiple pilot studies. See Appendix A.1.3 for details on CRSs, quality control, and the developed interface.

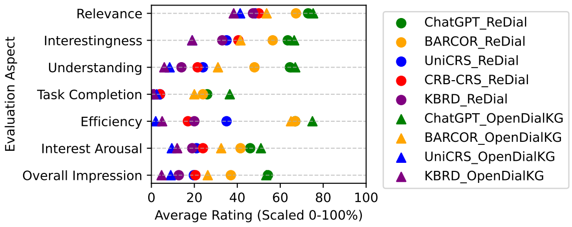


Figure 2: Distribution of human annotation scores for seven aspects across nine systems in CRSArena-Eval.

Evaluation Aspects. We follow Siro et al. (2022) and collect annotations for seven evaluation aspects that are central to CRSs, covering both system and user centric features of a conversation (see Appendix A.1.1 for more details). These aspects and their descriptions (used as instructions to annotators) are as follows:

Turn-level Aspects:
Relevance (0–3): Does the assistant's response make sense and meet the user's interests?
Interestingness (0–2): Does the response make the user want to continue the conversation?
Dialogue-level Aspects:
Understanding (0–2): Does the assistant understand the user's request and try to fulfill it?
Task Completion (0–2): Does the assistant make recommendations that the user finally accepts?
Efficiency (0–1): Does the assistant suggest items matching the user's interests within the first three interactions?
Interest Arousal (0–2): Does the assistant try to spark the user's interest in something new?
Overall Impression (0–4): What is the overall impression of the assistant's performance?

These instructions and scales are based on (Siro et al., 2022), with minor adjustments from Sakai (2023) for clarity. Turn-level aspects are evaluated for each system turn, while dialogue-level aspects are evaluated for the entire dialogue.

3.2 Analysis

We now analyze our collected annotations, referred to as *CRSArena-Eval*.

Statistics. A total of 20,962 annotations were collected, spanning 467 dialogues and 2,235 system turns. Each task was annotated by three workers, with additional annotations collected to resolve ties, yielding 6,805 final labels after majority voting. Annotation details are provided in Appendix A.2.1.

Inter-annotator Agreement. Given the ordinal nature of judgments, we report inter-annotator agreement using Pearson's r and Spearman's ρ, following (Mehri and Eskenazi, 2020), along with Krippendorff's α. Average scores across all aspects are $r = 0.443$, $\rho = 0.425$, and $\alpha = 0.436$, indicating moderate agreement. We note that these agreements exceed the existing high-quality annotations of AB-ReDial dataset (Siro et al., 2022, 2023), showing both the difficult nature of the task and high quality annotations of our dataset. Appendix A.2.2 details this comparison and the agreement calculation procedure.

System Score Distribution. Figure 2 shows the distribution of collected scores for the nine CRSs in CRSArena-Eval. It shows no system reaches the high end of the scale, indicating that existing CRSs do not fully satisfy users. This aside, the scores cover a broad range, reflecting differing system quality, which is crucial to assess the ability of automatic evaluators to distinguish system performance (Mehri and Eskenazi, 2020).

4 Experimental Setup

Datasets. For instruction optimization, we use the *AB-ReDial* (Siro et al., 2022, 2023) dataset, which contains annotations of human-human conversations from the ReDial dataset(cf. Sec. 3.1). The annotations are obtained for the seven evaluation aspects and are per turn/dialogue. This ensures no human-system conversations are involved in the optimization process. We use 60% of AB-ReDial for training and the rest for validation. For evaluation, we use the *CRSArena-Eval* dataset (cf. Sec. 3). CRSArena-Eval (RD)/(KG) denote the subset of the dataset for systems developed using ReDial (Li et al., 2018) and OpenDialKG (Moon et al., 2019), respectively.

Settings. Unless indicated otherwise Llama-3.1-8B-Instruct (Dubey et al., 2024) is used for FACE. We run the instruction optimization for $K = 6$ iterations and stored $b' = 16$ instructions in the instruction pool, resulting in 96 instructions. The final selection of the optimal instruction set $\mathbf{I}^*$ (line 20 of Algorithm 1) is done using the validation set and results in a set of 16 instructions. Prompts, hyperparameters, and other settings are detailed in Appendices C and E.

Baselines. Multiple automatic evaluators are used as our baselines: LLM^{Direct} directly prompts the LLM to annotate the given turn/dialogue using the same instructions as human annotations for CRSArena-Eval; $LLM^{CoT+ICL}$ adds CoT (Kojima et al., 2024; Wei et al., 2022) and in-context learning (ICL) (Brown et al., 2020) with two examples to the prompt of $\text{LLM}^{\text{Direct}}$; *UniEval* (Zhong et al., 2022) and *G-Eval* (Liu et al., 2023) are state-of-the-art reference-free conversation evaluation methods.

Methods	Turn-level				Dialogue-level										All	
	Rel.		Int.		Und.		Task		Eff.		Int.		Overall			
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
CRSArena-Eval (RD)																
LLM$^{\text{Direct}}$	0.464	0.455	0.248	0.260	0.522	0.482	0.405	0.363	0.101	0.101	0.217	0.203	0.564	0.522	0.360	0.341
LLM$^{\text{CoT+ICL}}$	0.453	0.446	0.175	0.177	0.481	0.457	0.425	0.400	0.174	0.174	0.188	0.174	0.498	0.472	0.342	0.329
UniEval	0.311	0.288	0.182	0.242	0.246	0.225	–	–	–	–	–	–	0.395	0.387	–	–
G-Eval	0.490	0.471	0.302	0.289	0.490	0.444	0.351	0.364	**0.488**	0.482	0.332	0.325	0.577	0.577	0.433	0.422
FACE w/o train	0.468	0.462	0.279	0.290	0.605	0.574	0.482	0.392	0.339	0.423	0.235	0.255	0.617	0.555	0.432	0.422
FACE	**0.549**	**0.550**	**0.443**	**0.437**	**0.650**	**0.635**	**0.570**	**0.453**	0.484	**0.534**	**0.447**	**0.430**	**0.712**	**0.668**	**0.551**	**0.530**
CRSArena-Eval (KG)																
LLM$^{\text{Direct}}$	0.452	0.452	0.238	0.231	0.599	0.546	0.538	0.481	0.137	0.137	0.425	0.378	0.655	0.557	0.435	0.397
LLM$^{\text{CoT+ICL}}$	0.419	0.408	0.203	0.190	0.562	0.520	0.475	0.434	0.114	0.114	0.309	0.279	0.599	0.521	0.383	0.352
UniEval	0.416	0.428	0.262	0.401	0.563	0.541	–	–	–	–	–	–	0.618	0.659	–	–
G-Eval	0.533	0.505	0.334	0.316	0.535	0.475	0.430	0.422	0.485	0.463	0.424	0.403	0.656	0.642	0.485	0.461
FACE w/o train	0.492	0.486	0.308	0.324	0.664	0.611	0.426	0.411	0.240	0.419	0.297	0.322	0.672	0.557	0.443	0.447
FACE	**0.543**	**0.527**	**0.471**	**0.453**	**0.719**	**0.677**	**0.593**	**0.484**	**0.518**	**0.543**	**0.449**	**0.404**	**0.766**	**0.679**	**0.580**	**0.538**

Table 1: Annotation correlations for reference-free evaluation methods. All columns show the correlations averaged over all aspects. All FACE correlations are statistically significant with $p < 0.01$.

Methods	Turn-level		Dial-level		All	
	r	ρ	r	ρ	r	ρ
R@1	-0.197	0.060	-0.120	0.081	-0.142	0.075
R@10	-0.192	0.048	-0.111	0.071	-0.134	0.064
Distinct-3	0.716	0.841	0.665	0.780	0.680	0.798
Distinct-4	0.654	0.800	0.609	0.760	0.622	0.771
LLM$^{\text{Direct}}$	0.860	0.822	0.872	0.799	0.868	0.806
G-Eval	0.740	0.840	0.893	0.830	0.850	0.833
FACE	**0.930**	**0.842**	**0.913**	**0.837**	**0.918**	**0.838**

Table 2: System ranking correlations on CRSArena-Eval, averaged over corresponding aspects. All FACE correlations are statistically significant with $p < 0.05$.

Since UniEval covers limited aspects, we only report those overlapping with ours. For fair comparison, we use Llama-3.1-8B-Instruct as the backbone for LLM-based methods LLM$^{\text{Direct}}$, LLM$^{\text{CoT+ICL}}$, and G-Eval in Section 5.1. To show generalizability to different LLMs in Section 5.2, other LLMs are used for our experiments.

Metrics. For correlation metrics, we use Pearson's and Spearman's to appropriately handle the annotation scales. Following (Mehri and Eskenazi, 2020), correlation significance is computed by p-value derived from t-distribution using Python's SciPy library (Virtanen et al., 2020).

5 Results

We begin by outlining our key research questions and then present a series of experiments conducted to address them: **RQ1:** How does FACE correlate with human judgments? **RQ2:** How generalizable are FACE-optimized instructions across different LLMs and domains? **RQ3:** Can fine-grained evaluation scores of FACE provide insights about system's issues?

5.1 Annotation Correlation

Table 1 shows the annotation correlation results, demonstrating that FACE, on average, outperforms all baselines by a large margin. We note that FACE is not optimized on any subset of CRSArena-Eval (cf. Sec. 4), highlighting its strong generalization to unseen systems. Although one can argue that FACE might have captured some information from ReDial dataset during the optimization process, results on CRSArena-Eval (KG) shows generalization and robustness of FACE to unseen recommendation datasets.

The results also demonstrate that even without instruction optimization (FACE w/o train), FACE remains competitive with the state-of-the-art method, G-Eval, suggesting the effectiveness of our particle-based approach. The benefit of training is more pronounced for challenging aspects such as *Interestingness*, *Efficiency*, and *Interest Arousal*, which indicates that FACE effectively optimizes instructions for aspects that LLMs struggle to capture using a single thought process.

Table 2 shows the correlation of system rankings created by different automatic evaluation methods. System ranking correlations are calculated by averaging each system's score, ranking systems, and measuring correlation with system rankings based on human judgments. We obtain correlation for reference-based metrics by computing system rankings from reported scores (Wang et al., 2023a). As baselines, we report Recall, and Distinct-n (Li et al., 2016) as reference-based metrics, and LLM$^{\text{Direct}}$ and G-Eval as top-performing reference-free metrics from Table 1.

From the results, we notice that recall metrics are insufficient, which is in line with the litera-

Methods	LLM	Size	CRS-RD		CRS-KG		Avg.	
			r	ρ	r	ρ	r	ρ
LLM$^{\text{Direct}}$	Llama	8B	0.564	0.522	0.655	0.557	0.610	0.540
G-Eval	Llama	8B	0.577	0.577	0.656	0.642	0.617	0.610
FACE	Llama	8B	**0.712**	0.668	**0.766**	0.679	**0.739**	0.674
FACE*	Gemma	9B	0.689	**0.687**	0.718	**0.703**	0.704	**0.695**
FACE*	Gemma	2B	0.647	0.603	0.728	0.646	0.688	0.625
FACE*	Qwen	7B	0.698	0.664	0.764	0.693	0.731	0.679
FACE*	Qwen	3B	0.643	0.632	0.725	0.674	0.684	0.653
FACE*	Qwen	1.5B	0.557	0.606	0.605	0.635	0.581	0.621

Table 3: Results on generalizability of FACE to other LLMs. CRS-RD and -KG represent CRSArena-Eval (RD) and (KG), respectively. All FACE annotation correlations are statistically significant with $p < 0.01$.

Methods	USR-Persona		USR-Topical		Avg.	
	r	ρ	r	ρ	r	ρ
ROUGE-L	0.114	0.091	0.193	0.203	0.154	0.147
BLEU-4	0.147	0.151	0.131	0.235	0.139	0.193
METEOR	0.250	0.256	0.250	0.302	0.250	0.279
BERTScore	0.188	0.157	0.214	0.233	0.201	0.195
Dial-M	0.400	0.390	0.370	0.400	0.385	0.395
USR	0.607	0.528	0.416	0.377	0.512	0.453
UniEval	0.616	0.580	**0.595**	**0.613**	0.605	0.597
G-Eval$^{\text{GPT-3.5}}$	0.441	0.458	0.519	0.544	0.480	0.501
G-Eval$^{\text{GPT-4}}$	0.607	0.670	0.594	0.605	0.601	0.638
FACE	0.473	0.544	0.498	0.506	0.486	0.525
FACE*$_{72\text{B}}$	**0.681**	**0.697**	0.570	0.582	**0.625**	**0.639**

Table 4: Results on generalizability of FACE to chitchat conversations. All FACE correlations are statistically significant with $p < 0.01$.

ture (Bernard et al., 2025). While the results show that FACE outperforms all other methods, we note that system ranking correlations should be consumed with caution, as they are less informative than annotation correlation, especially for competitive systems (Faggioli et al., 2023).

Overall, we can answer **(RQ1)**: FACE achieves high annotation and system ranking correlations with human judgments, outperforming state-of-the-art methods by a large margin.

5.2 Generalizability of FACE

We hypothesize that the pool of optimized instructions by FACE can be reused for different LLMs and domains. To assess this hypothesis, we take the instruction pool and re-select the top instructions (line 20 of Algorithm 1) for different LLMs and datasets. We denote this adapted FACE as *FACE**.

Generalization to other LLMs. To assess generalizability of FACE to other LLMs, we use the AB-ReDial validation set and re-select instructions for five LLMs: Gemma 2 (9B and 2B) and Qewn 2.5 (7B, 3B, 1.B). Table 3 shows annotation correlation results for top-performing baselines of Table 1. The results indicate that adapting to Gemma 9B and Qwen 7B achieves performance comparable to FACE. Interestingly, our method is highly effective for small models: FACE adapted to Gemma 2B outperforms G-Eval, which uses an LLM with 4x more parameters. A similar observation can be made for Qwen 3B and 1.5B.

Generalization to Chitchat Conversations. To examine generalizability of FACE to another type of conversations, we evaluate FACE on the exiting chitchat datasets: (1) *USR-Persona* (Mehri and Eskenazi, 2020) (based on PersonaChat (Zhang et al., 2018)), containing personalized chit-chats, and (2) *USR-Topical* (Mehri and Eskenazi, 2020) (based on Topical-Chat (Gopalakrishnan et al., 2019)), containing knowledge-grounded conversations. These datasets provide annotations for six evaluation aspects, of which "maintains context" is the only aspect that is similar to ours. We use the instruction pool for the relevance aspect and re-select optimal instructions using the validation set of USR-Persona for USR-Topical evaluation and vice versa, to ensure that the test set is completely unseen.

Table 4 presents the results of FACE generalizability to chitchat conversations, with G-Eval results obtained using GPT-3.5 and 4. On average, FACE outperforms all baselines except G-Eval with GPT-4, while FACE*$_{72\text{B}}$ outperforms all baselines. This is especially striking, considering that FACE is completely blind to category of conversations and uses an LLM with a lower number of parameters than GPT. Additionally, using an open model for evaluation has the added value of reproducibility.

Based on the results of Tables 3 and 4, we answer our second research question **(RQ2)**: FACE-optimized instructions are highly generalizable to different LLMs and domains, by performing a simple adaptation of FACE to a new LLM/domain. The adaptation to larger models can even surpass the state-of-the-art method with GPT-4 as a backbone on chit-chat conversations.

5.3 FACE Interpretability

To demonstrate how FACE fine-grained scores can help humans to identify issues of CRSs, we compare two competitive CRSs: BARCOR (Wang et al., 2022a) and UniCRS (Wang et al., 2022b). Humans and FACE prefer BARCOR (cf. Fig. 2 and (Bernard et al., 2025)), while recall-based metrics favor UniCRS (Wang et al., 2023a). The left radar chart in Figure 3 shows FACE analysis for BARCOR and UniCRS. While the overall impression indicates similar performance, UniCRS excels in relevance and efficiency, whereas BARCOR is better in user understanding and keeping users interested.

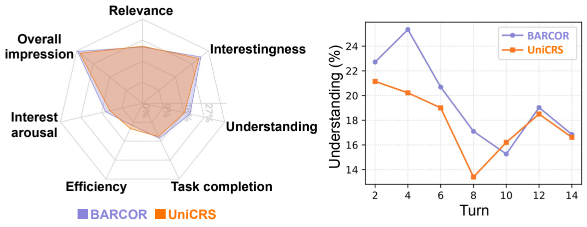


Figure 3: Breakdown analysis comparing BARCOR and UNICRS. Left: FACE evaluation results for each aspect. Right: Scores of Understanding aspect per system turn.

The right graph shows the user understanding aspect for each system turn. BARCOR shows higher scores in earlier turns, indicating that it understands user preferences early on, which may explain its higher human preference. These insights suggest that, while UniCRS excels in recommendations, overall performance can be improved by focusing on user understanding and preference elicitation.

Overall, we answer **(RQ3)** positively: FACE providing valuable insights into system's behavior, which are useful for system improvement.

6 Analysis

Sample Efficiency. To determine the systems' score and create a system ranking, the evaluation method needs to be fed with a sample of user-system conversations. To measure how many samples are needed to find a system ranking with a high correlation with human judgments, we plot system ranking correlations for various conversation counts per system in Figure 4. The results indicate that FACE has strong sample efficiency; it achieves a Spearman correlation of 0.8 with gold rankings using only 3 dialogues per system, making it twice as efficient as the best-performing existing method, G-Eval. Given that collecting human-system conversations require cost and effort, FACE's sample efficiency significantly enhances actual usability.

Bias Analysis. We analyze whether FACE shows known LLM biases: length bias and self-bias (cf. Sect. 7). Notably, we find no evidence of either bias in FACE. Indeed, FACE correlates less with response length than human evaluators and, surprisingly, tends to prefer human responses to system responses compared to human evaluators; see Appendix D for details.

7 Related Work

Recent advancements in LLMs have led to various LLM-based, reference-free automatic evaluation methods (Upadhyay et al., 2024; Dubois et al., 2024b; Liu et al., 2023; Lin and Chen, 2023; Zheng et al., 2024a; Dey and Desarkar, 2023). However, improved performance comes with challenges. *Evaluation bias* is one of them; LLMs tend to favor longer responses (length bias) (Wang et al., 2024; Dubois et al., 2024b) and are biased toward texts from similar models (self-bias) (Xu et al., 2024; Liu et al., 2023). *Generalizability* is another challenge; LLMs are sensitive to handcrafted, arbitrary prompts, which are not reusable for different models (Sclar et al., 2024; Razavi et al., 2025). FACE mitigates these biases with conversation particles and instruction optimization (see Section 6).

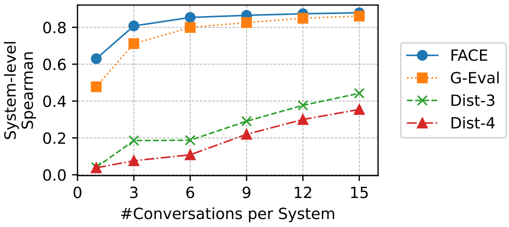


Figure 4: Sample efficiency of different evaluation methods for the overall aspect on CRSArena-Eval.

To assign granular evaluation scores, multiple nugget-based evaluation was proposed (Voorhees, 2003; Mayfield et al., 2024; Pradeep et al., 2024; Lin and Demner-Fushman, 2005; Ekstrand-Abueg et al., 2013; Takehi et al., 2023; Rajput et al., 2011; Dietz, 2024). However, these works are reference-based and/or focus on individual responses, and more importantly, they do not target information-seeking conversations. To this aim, SWAN (Sakai, 2023) proposes a conceptual framework for fine-grained evaluations. While promising in concept, its execution remains an open question; inspired by SWAN, we tackle these challenges.

The detailed related work on automatic evaluations, nugget-based approaches, and instruction optimizations, is in Appendix F.

8 Conclusion

We present FACE, a fine-grained, aspect-based evaluation method for CRSs. It addresses the shortcomings of existing metrics, such as focusing on fixed dialogue history with reference-based metrics, overlooking diverse conversation trajectories, and relying on non-granular scores with limited insights. FACE is shown to strongly correlate with human judgments, generalize across LLMs and domains, and provide insights for system improvement. Future work needs to address current limitations by further examining evaluation biases, assessing effectiveness across broad domains, and exploring how FACE can help expert evaluators.

9 Limitations

The limitations of this paper are as follows: (1) While FACE is a general method applicable to various conversations, this paper evaluated it only on CRSs and one aspect for chit-chat; further exploration in other domains/aspects is needed. (2) Although FACE did not exhibit bias in our analysis (Sec. 6), evaluating unknown biases remains underexplored. (3) Knowing the limitations of LLM-based evaluation (Soboroff, 2024; Clarke and Dietz, 2024; Faggioli et al., 2023), we emphasize that FACE may not replace expert human evaluations. Instead, it facilitates research and development on conversational systems by offering a scalable and efficient evaluation method (Dubois et al., 2024a).

References

Marwah Alaofi, Negar Arabzadeh, Charles LA Clarke, and Mark Sanderson. 2024. Generative information retrieval evaluation. *arXiv preprint arXiv:2404.08137*.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*.

Nolwenn Bernard and Krisztian Balog. 2023. Mgshopdial: A multi-goal conversational dataset for e-commerce. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 23–27, 2023, Taipei, Taiwan*.

Nolwenn Bernard, Hideaki Joko, Faegheh Hasibi, and Krisztian Balog. 2025. Crs arena: Crowdsourced benchmarking of conversational recommender systems. In *Proceedings of the 18th ACM International Conference on Web Search and Data Mining*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2024a. Instructzero: Efficient instruction optimization for black-box large language models. In *Proceedings of International Conference on Learning Representations*.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Weizhe Chen, Sven Koenig, and Bistra Dilkina. 2024b. Reprompt: Planning by automatic prompt engineering for large language models agents. *arXiv preprint arXiv:2406.11132*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2025. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of International Conference on Learning Representations*.

Charles L. A. Clarke and Laura Dietz. 2024. Llm-based relevance assessment still can't replace human relevance assessment. *arXiv preprint arXiv:2412.17156*.

Suvodip Dey and Maunendra Sankar Desarkar. 2023. Dial-m: A masking-based framework for dialogue evaluation. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Laura Dietz. 2024. A workbench for autograding retrieve/generate systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2024a. Alpacafarm: a simulation framework for methods that learn from human feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Yann Dubois, Percy Liang, and Tatsunori B. Hashimoto. 2024b. Length-controlled alpacaeval: A simple way to debias automatic evaluators. In *Conference on Language Modeling (COLM)*.

Carsten Eickhoff and Arjen P. de Vries. 2011. How crowdsourcable is your task? In *Proc. of CSDM '11*, pages 11–14.

Matthew Ekstrand-Abueg, Virgil Pavlu, Makoto Kato, Tetsuya Sakai, Takehiro Yamamoto, and Mayu Iwata. 2013. Exploring semi-automatic nugget extraction for japanese one click access evaluation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2025. Promptbreeder: self-referential self-improvement via prompt evolution. In *Proceedings of International Conference on Learning Representations*.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, Dilek Hakkani-Tur, and Amazon Alexa AI. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. In *Proceedings of Interspeech 2019*.

Hideaki Joko, Shubham Chatterjee, Andrew Ramsay, Arjen P. de Vries, Jeff Dalton, and Faegheh Hasibi. 2024. Doing personal laps: Llm-augmented dialogue construction for personalized multi-session conversational search. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2024. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Weize Kong, Spurthi Hombaiah, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Prewrite: Prompt rewriting with reinforcement learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Jimmy Lin and Dina Demner-Fushman. 2005. Automatically evaluating answers to definition questions. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In *Proceedings of the 5th Workshop on NLP for Conversational AI*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*.

Ahtsham Manzoor and Dietmar Jannach. 2022. Towards retrieval-based conversational recommendation. *Information Systems*.

James Mayfield, Eugene Yang, Dawn Lawrie, Sean MacAvaney, Paul McNamee, Douglas W. Oard, Luca Soldaini, Ian Soboroff, Orion Weller, Efsun Kayi, Kate Sanders, Marc Mason, and Noah Hibbler. 2024. On the evaluation of machine-generated reports. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Stefano Perrella, Lorenzo Proietti, Pere-Lluís Huguet Cabot, Edoardo Barba, and Roberto Navigli. 2024. Beyond correlation: Interpretable evaluation of machine translation metrics. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2024. Initial nugget evaluation results for the trec 2024 rag track with the autonuggetizer framework. *arXiv preprint arXiv:2411.09607*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with “gradient descent” and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached conversational preference elicitation: A case study in understanding movie preferences. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*.

Shahzad Rajput, Virgil Pavlu, Peter B. Golbus, and Javed A. Aslam. 2011. A nugget-based test collection construction paradigm. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*.

Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part III*.

Tetsuya Sakai. 2023. Swan: A generic framework for auditing textual conversational systems. *arXiv preprint arXiv:2305.08290*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *Proceedings of International Conference on Learning Representations*.

Clemencia Siro, Mohammad Aliannejadi, and Maarten de Rijke. 2022. Understanding user satisfaction with task-oriented dialogue systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Clemencia Siro, Mohammad Aliannejadi, and Maarten De Rijke. 2023. Understanding and predicting user satisfaction with conversational recommender systems. *ACM Trans. Inf. Syst.*

Ian Soboroff. 2024. Don’t use llms to make relevance judgments. *arXiv preprint arXiv:2409.15133*.

Rikiya Takehi, Akihisa Watanabe, and Tetsuya Sakai. 2023. Open-domain dialogue quality evaluation: Deriving nugget-level scores from turn-level scores. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*.

Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. 2024. A large-scale study of relevance assessments with large language models: An initial look. *arXiv preprint arXiv:2411.08275*.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, and 1 others. 2020. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*.

Ellen M. Voorhees. 2003. Overview of the trec 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference*.

Ting-Chun Wang, Shang-Yu Su, and Yun-Nung Chen. 2022a. Barcor: Towards a unified framework for conversational recommendation systems. *arXiv preprint arXiv:2203.14257*.

Xiaolei Wang, Xinyu Tang, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023a. Rethinking the evaluation for conversational recommendation in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022b. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of International Conference on Learning Representations*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2024. How far can camels go? exploring the state of instruction tuning on open resources. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: LLM amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *Proceedings of International Conference on Learning Representations*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2025. Large language models as optimizers. In *Proceedings of International Conference on Learning Representations*.

Qinyuan Ye, Mohamed Ahmed, Reid Pryzant, and Fereshte Khani. 2024. Prompt engineering a prompt engineer. In *Findings of the Association for Computational Linguistics ACL 2024*.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. 2024. Textgrad: Automatic "differentiation" via text. *arXiv preprint arXiv:2406.07496*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Shuo Zhang and Krisztian Balog. 2020. Evaluating conversational recommender systems via user simulation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Shuo Zhang, Mu-Chun Wang, and Krisztian Balog. 2022. Analyzing and simulating user utterance reformulation in conversational recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *Proceedings of International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024a. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue (Livia) Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph Gonzalez, Clark Barrett, and Ying Sheng. 2024b. Sglang: Efficient execution of structured language model programs. In *The Thirty-Eighth Annual Conference on Neural Information Processing Systems*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *Proceedings of International Conference on Learning Representations*.

A Human Annotation Collection Details

A.1 Annotation Setup

In this section, we provide details on the human annotation collection process, including the annotation interface and aspect selection, participant selection and quality control, and the CRSArena-Dial dataset used for annotation.

A.1.1 Interface and Aspect Selection

This section describes the annotation interface and the selection of evaluation aspects for our annotation collection.

Annotation Interface. To crowdsource high-quality annotations, we built an interface, through multiple pilot experiments, to overcome the widely reported challenges in the literature (Joko et al., 2024; Bernard and Balog, 2023; Radlinski et al., 2019; Eickhoff and de Vries, 2011). This includes workers (1) skipping reading context, (2) misunderstanding aspect definitions, (3) geting distracted by overwhelming information on the annotation page, and (4) annotating randomly without focus. Our interface requires users to pass a quiz on aspect definitions and enforces the annotation of each turn before evaluating the entire dialogue. It further includes hidden tests with expert-verified answers, and workers who fail to meet the required agreement are dismissed.

Evaluation Aspect Selection. To determine the evaluation aspects for annotation collection, we performed a comprehensive review of literature on conversational system evaluations, identifying over 50 aspects. The full list of these aspects, compiled from 21 studies, will be available in the GitHub repository upon acceptance.

A.1.2 Participants and Quality Control

We recruited Prolific[3] workers from English-speaking countries with a 100% approval rate and ≥ 1000 previous submissions. Considering some

[3] https://www.prolific.co/

workers exhibit behavior aimed at just maximizing financial gain (Eickhoff and de Vries, 2011), we filtered out those with a history of subpar submissions to ensure quality. Furthermore, workers with <30% agreement with experts on hidden tests were excluded. Each batch of work contained annotations for 20 dialogues, taking around 40 minutes to complete, at the cost of £6. Three annotations per annotation task were collected. In case of disagreement, additional annotations were collected until ties were resolved.

A.1.3 CRSArena-Dial Dataset

Here, we provide the CRSs contained in the CRSArena-Dial (Bernard et al., 2025) dataset and the preprocessing steps we applied to the dataset.

CRSs. The CRSArena-Dial dataset consists of human conversation with nine state-of-the-art CRSs, including KBRD (Chen et al., 2019), BARCOR (Wang et al., 2022a), UniCRS (Wang et al., 2022b), ChatGPT (Wang et al., 2023a), and CRB-CRS (Manzoor and Jannach, 2022), each developed based on OpenDialKG (Moon et al., 2019) and ReDial (Li et al., 2018) datasets, except CRB-CRS, which is solely on the ReDial dataset.

Preprocessing. To ensure the quality of dialogues, we excluded seven dialogues that were unsuitable for our annotation, such as those with only a single user utterance, resulting in 467 dialogues with a total of 2,235 system responses for our annotations.

A.2 Annotation Results Details

A.2.1 Statistics

In total, 109 workers were recruited for our annotation collection. On average, each dialogue has 14.6 annotation tasks, each annotated by three workers. Noteworthy, our annotation interface made the process highly efficient, requiring only 8 seconds per annotation. For 92% of the tasks, an agreement was achieved by the first three annotators and for the remaining 8% of tasks additional annotations were collected to resolve ties.

A.2.2 Inter-annotator Agreement

This section describes the inter-annotator agreement calculation process, as well as the comparison of our results with the AB-ReDial dataset (Siro et al., 2022, 2023). We calculate the Person's r and Spearman's ρ by taking the average of correlation between each pair of annotations (Manning et al., 2008; Mehri and Eskenazi, 2020). For comparison, we calculate the agreement for the same aspects on the AB-ReDial dataset (Siro et al., 2022, 2023), yielding $r = 0.328$, $\rho = 0.325$, and $\alpha = 0.327$, showing higher quality of CRSArena-Eval. Table 5 shows the inter-annotator agreement for each aspect. CRSArena-Eval demonstrates higher agreement than AB-ReDial in all aspects except for interest arousal, highlighting the quality of our annotations.

Aspect	CRSArena-Eval			AB-ReDial		
	r	ρ	α	r	ρ	α
Turn-level						
Relevance	**0.613**	**0.611**	**0.612**	0.527	0.502	0.526
Interestingness	**0.386**	**0.386**	**0.375**	0.209	0.217	0.209
Dialogue-level						
Understanding	**0.505**	**0.477**	**0.496**	0.321	0.313	0.318
Task Completion	**0.481**	**0.440**	**0.482**	0.345	0.321	0.346
Efficiency	**0.297**	**0.297**	**0.289**	0.225	0.225	0.226
Interest Arousal	0.242	0.241	0.226	**0.254**	**0.291**	**0.247**
Overall Impression	**0.573**	**0.526**	**0.572**	0.321	0.300	0.321

Table 5: Inter-annotator agreement of CRSArena-Eval and AB-ReDial (Siro et al., 2023) based on Pearson's r, Spearman's ρ, and Krippendorff's α correlations.

B Method Algorithm Details

B.1 UCB Selection Algorithm

The UCB selection algorithm, denoted as $Select^{UCB}_{b'}(\cdot)$, is presented in Algorithm 2. For each iteration t, $N_t(I)$ denotes the number of evaluations of instruction I on sampled particles and $Q_t(I)$ denotes its estimated correlation. Following (Pryzant et al., 2023), it samples a subset of particles and their corresponding human annotations, then selects the instruction that maximizes the UCB criterion $Q_t(I) + c\sqrt{\log t / N_t(I)}$, where c is an exploration constant. The selected instruction is evaluated on the sampled particles, and its estimated effectiveness is updated based on the correlation with human annotations. After T iterations, the algorithm returns the candidate instructions $\mathbf{I}^{\text{cand}}_k$ containing the top b' instructions according to their final estimated effectiveness Q_T with the $SelectTopInstructions_{b'}(Q_T)$ function. This forms a set of promising candidates for the next stage of the selection process. We note that, for efficient execution of the UCB process, we approximate it by dividing T into multiple small batches and processing each batch in parallel.

B.2 Instruction Optimization Details

For the prompt ∇, we employ reasoning templates (Ye et al., 2024), which provide a set of items to be

Algorithm 2 $Select_{b'}^{UCB}(\cdot)$ - Candidate Selection with UCB Bandits

```
Require: All rewritten instructions $\mathbf{I}'_k$, particles $\mathbf{P}$, human annotations $\mathbf{H}$, number of UCB iterations $T$, and the number of selected instructions $b'$.
Initialize $N_t(I) \leftarrow 0, Q_t(I) \leftarrow 0, \forall I \in \mathbf{I}'_k$
for $t = 1, ..., T$ do
    // Sample particles and corresponding labels uniformly
    $\mathbf{P}_{smp} \subset \mathbf{P}, \mathbf{H}_{smp} \subset \mathbf{H}$
    // Select the instruction with the highest UCB criterion
    $I \leftarrow \arg\max_{I \in \mathbf{I}'_k} \{Q_t(I) + c\sqrt{\frac{\log t}{N_t(I)}}\}$
    // Evaluate the instruction $I$ on the sampled particles
    $\mathbf{S}_{smp} \leftarrow \{\mathcal{E}_{particle}(I, p)\}_{p \in \mathbf{P}_{smp}}$
    // Compute correlation and update UCB criterion
    Observe reward $r \leftarrow \mathcal{C}(\mathbf{S}_{smp}, \mathbf{H}_{smp})$
    $N_t(I) \leftarrow N_t(I) + |\mathbf{P}_{smp}|$
    $Q_t(I) \leftarrow Q_t(I) + \frac{r - Q_t(I)}{N_t(I)}$
end for
return $\mathbf{I}_k^{\text{cand}} \leftarrow SelectTopInstructions_{b'}(Q_T)$
```

considered by $\mathcal{G}_\nabla$. Our items include identifying inconsistencies between the predicted and human annotations, evaluating the correctness of the current task and CoT instructions, and suggesting edits to these instructions, if necessary.

C Experimental Setup Details

C.1 Software Libraries and Hyperparameters

All experiments were performed using the SGLang (Zheng et al., 2024b) library for its prediction efficiency. The temperature of 0.6 is set across all experiments unless otherwise stated. For instruction optimization, we set parameters $\alpha = 2$, $c = 1$, $b = 4$, and use the batch size of $B = |\mathbf{I}'_k|/2$ and $T = 5B$ iterations. A sampling size of $n = 5$ is used to create a score distribution (Sect. 2.2). All hyper parameters are obtained using the validation set or following (Pryzant et al., 2023).

C.2 Implementation Details

Here, we describe the implementation we used to improve the efficiency of instruction optimization process (Sect. 2.3). In this process, while theoretically two distinct LLMs $\mathcal{L}_2$ and $\mathcal{L}_3$ are used for gradient generation and instruction rewriting, the two steps can be merged into a single LLM call by concatenating ∇ and δ. This halves the number of LLM calls, resulting in a significant speedup of the optimization process.

D Bias Analysis

We conduct analyses to see whether FACE exhibits length bias and self-bias.

D.1 Length Bias

For length bias (Dubois et al., 2024b; Wang et al., 2024), using CRSArena-Eval, we examine the correlation between a system's average word count in conversations and the overall score. We find that Pearson's correlations are 0.824 and 0.868 for FACE and humans, respectively. This indicates no sign of length bias compared to humans, which is in line with existing work (Chiang et al., 2025; Dubois et al., 2024b) that report humans also favour longer responses, highlighting the nuanced nature of the LLM length bias.

D.2 Self-Bias

For self-bias, where LLMs prefer system responses over human ones, we use FACE to evaluate pairs of system- and human-generated responses and see if they show any preferences compared to gold human annotators. We examine two conversation types: USR-Persona for chit-chat, and a combination of CRSArena-Eval and AB-ReDial for a recommendation. We could not find evidence for self-bias in FACE; e.g., for USR-Persona, FACE aligns with human preferences 77.8% of the time when humans prefer human-generated responses and 71.4% of the time when they prefer system-generated responses.

E Prompts

In this section, we first provide an overview (Appendix E.1) of the specific prompts employed in our method and then illustrate concrete examples (Appendix E.2) of each prompt.

E.1 Overview

Inspired by TREC RAG 2024 (Pradeep et al., 2024), the decomposer prompt starts with: *"Your task is to extract conversation particles [...]"* followed by CoT prompts and the format of nuggets. The textual gradient prompt ∇ begins with *"Examine the original instructions, predicted nugget score, and gold score."* and then identifies inconsistency between predicted and gold scores, followed by suggestions to the instruction if necessary. The instruction rewriting prompt δ starts with *"Propose new instructions of 50 words based on [...]"* followed by the guidance on how to rewrite based on the textual gradient, inspired by (Ye et al., 2024). The seed evaluation instruction I is as follows *"Given the dialogue, evaluate the quality of the target nugget based on the given aspect. Step 1: [...]"*. Examples of prompts are provided below, and all used and optimized prompts can be found in our GitHub repository upon acceptance.

E.2 Examples

E.2.1 Conversation Particle Generation Prompt

> *Dialogue History:* `{dialogue_history}`
> *Target Assistant Turn:* `{target_turn}`
> *User's Response:* `{user_response}`
>
> ***Your task is to extract conversation particles, which are minimal, atomic units of information or facts from the target assistant turn.***
>
> *Each nugget consists of:*
>
> - `"dialogue_act"`*: one of the following labels:* `"greeting,"` `"preference elicitation,"` `"recommendation,"` `"goodbye,"` or `"others."`
> - `"nugget_mention"`*: the atomic unit of information from the target assistant turn. [...]*
> - `"user_feedback"`*: the excerpt of user feedback against the given nugget. [...]*
>
> *The output must be a JSON list of nuggets. [...]*
>
> *Must think step by step:*
>
> 1. *Explain the dialogue history, the target assistant turn, and the user feedback.*
> 2. *How many conversation particles are found in the target assistant turn?*
> 3. *For each nugget, discuss the meaning of the user feedback.*
> 4. *Output in JSON format.*

E.2.2 Textual Gradient Prompt ∇

> ***Examine the original instructions, predicted nugget score, and gold dialogue (or turn) score.***
>
> - *Based on the gold dialogue (or turn) score, is the predicted nugget score reasonable?*
> - *Does original instructions describe how to use the nugget's information correctly?*
> - *Necessary to edit the original instructions?*

E.2.3 Instruction Rewriting Prompt δ

> ***Propose new instructions of ~50 words based on the feedback.***
>
> - *Note that the full dialogue can be changed, thus your new instructions must be general enough to handle different contexts.*
> - *Note that the task is "nugget" evaluation, not "turn" or "dialogue" evaluation; thus, the new instructions should focus on how to use the nugget.*
> - *Must provide "task description" and explicit "step-by-step instructions" for the nugget evaluation; in step-by-step instructions labeling each step as "Step 1," "Step 2," and so on.*
> - *Break down the evaluation into smaller steps and provide a checklist ("Does the nugget...?" or "Is this nugget...?") for each step.*
>
> ...

E.2.4 Initial Prompt (Before Optimization)

> ***Task description:*** *Given the dialogue, evaluate the quality of the target nugget based on the* `{evaluation_aspect}`.
>
> ***Step-by-step instructions:***
>
> - ***Step 1:*** *Read the dialogue history, target nugget, and user's response.*
>   - *What does the target nugget convey?*
> - ***Step 2:*** *Carefully read the grading criteria.*
>   - *What are the grading criteria?*
> - ***Step 3:*** *Evaluate the target nugget.*
>   - *Which grade should be assigned to the target nugget?*

E.2.5 FACE-Optimized Instructions

Here, we provide the optimized instruction examples for the dialogue-level overall impression aspect and the turn-level relevance aspect. Please note that, in the actual process, FACE optimizes **multiple** instructions for each aspect, as shown in Fig. 1.

Optimized Instructions for Overall Impression Aspect (Dialogue-level)

> ***Task description:*** *Evaluate the nugget based on its relevance, accuracy, and usefulness.*
>
> ***Step-by-step instructions:***
>
> - ***Step 1:*** *Check if the nugget is relevant to the conversation.*
>   - *Does the nugget relate to the dialogue context?*
>   - *Is the nugget a direct response to the user's question or concern?*
>   - *Is the nugget related to the user's preferences or interests?*
> - ***Step 2:*** *Evaluate the nugget's accuracy.*
>   - *Is the information in the nugget accurate based on the dialogue?*
>   - *Does the nugget correctly represent the conversation?*
> - ***Step 3:*** *Assess the nugget's usefulness.*
>   - *Does the nugget provide a helpful or relevant suggestion?*
>   - *Does the nugget address the user's needs or concerns?*
>   - *Does the nugget facilitate a meaningful continuation of the conversation?*

Optimized Instructions for Relevance Aspect (Turn-level Aspect)

> ***Task description:*** *Evaluate the quality of the target nugget based on its relevance to the user's request.*
>
> ***Step-by-step instructions:***
>
> - ***Step 1:*** *Identify the user's request and the nugget's suggestion.*
>   - *Step 1.1: Does the nugget's suggestion directly address the user's request?*
>   - *Step 1.2: Is the nugget's genre or category aligned with the user's interest?*

- ***Step 2:*** *Assess the nugget's relevance.*
  - *Step 2.1: Does the nugget's information accurately address the user's need?*
  - *Step 2.2: Is the nugget's suggestion consistent with the user's preferences or interests?*

F Related Work Details

Automatic Conversation Evaluation. Although human annotations are the gold standard for evaluating CIS systems, they are expensive and time-consuming; thus, automatic evaluation methods have been proposed to scale up the evaluation process. There are two main types of automatic evaluation methods: *reference-based* and *reference-free*.

Reference-based methods use gold references to evaluate system responses, which include Recall@K, BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BERTScore (Zhang et al., 2020). While these methods are effective for machine translation and traditional IR tasks, they have limitations in conversation evaluation, as they overlook diverse response possibilities and various evaluation aspects. These limitations are supported by various studies showing a weak correlation between reference-based methods and human evaluations (Mehri and Eskenazi, 2020; Bernard et al., 2025; Liu et al., 2016).

Reference-free methods have been proposed to address these limitations (Mehri and Eskenazi, 2020; Zhong et al., 2022; Liu et al., 2023). These methods evaluate system responses without relying on gold references, consider multiple aspects of system quality, and allow for the assessment of various response possibilities. However, these methods primarily focus on turn-level evaluation with a fixed dialogue history, limiting their ability to assess the whole conversation and capture the diverse user-system interaction trajectories, which are crucial for evaluating system performance in real-world scenarios (Siro et al., 2022, 2023). FACE addresses these limitations by evaluating the system based on the whole conversation and capturing multiple conversation trajectories from diverse user-system interactions.

LLMs for Evaluation. For instance, G-Eval (Liu et al., 2023) uses an LLM to generate an evaluation prompt, which is then used to assess the system's response, demonstrating a strong correlation with human evaluations in chit-chat conversations.

Instruction optimization. To address the arbitrary nature of handcrafted prompts, various instruction optimization (or prompt optimization) methods have been proposed (Chen et al., 2024a; Kong et al., 2024; Pryzant et al., 2023; Ye et al., 2024; Chen et al., 2024b; Fernando et al., 2025; Zhou et al., 2023; Yang et al., 2024; Yuksekgonul et al., 2024; Yang et al., 2025). Zhou et al. (2023) introduced APE, an automatic prompt engineering method that uses LLMs to generate prompt candidates, and perform a Monte Carlo search to find the optimal prompt. Kong et al. (2024) proposed PRewrite, where prompts are optimized using proximal policy optimization (PPO) (Schulman et al., 2017). Pryzant et al. (2023) proposed an instruction optimization method using "textual gradient," which provides natural language feedback for an LLM to optimize prompts. These studies focus on common tasks like QA and classification, leaving conversation evaluation unexplored. More importantly, optimized prompts are not generalizable between LLMs (Zhou et al., 2023), which is crucial for evaluation tasks, where reusability is essential. In this work, we present a method for applying a textual gradient approach to enhance conversation evaluation alongside effective strategies for transferring optimized prompts across different settings.

Nugget-based Evaluation. Here, we describe the expanded version of related work on nugget-based evaluation (cf. Sect. 7). Nugget-based evaluation was proposed (Voorhees, 2003) to assign granular scores to system responses for non-binary queries, wherein a nugget, which is an atomic piece of information, serves as the unit of evaluation, enabling a more traceable assessment. Although the original nugget-based evaluation was intended as a manual method, many efforts have aimed to automate or semi-automate it (Mayfield et al., 2024; Pradeep et al., 2024; Lin and Demner-Fushman, 2005; Ekstrand-Abueg et al., 2013; Takehi et al., 2023; Rajput et al., 2011; Dietz, 2024). One of the earlier methods is POURPRE (Lin and Demner-Fushman, 2005), an automatic nugget-based evaluation method that uses n-gram co-occurrences to assess nugget presence in system responses. More recent research employs LLMs for nugget matching; Pradeep et al. (2024) introduced the AutoNuggetizer framework in TREC RAG 2024, where LLMs automatically create nuggets and assign them to system responses. However, these works are reference-based and/or focus on individual responses, and more importantly, they do not target CIS systems.

SWAN (Sakai, 2023) proposes a conceptual framework to evaluate CIS systems by decomposing user-system interactions into units and assessing them on various aspects. While SWAN's concept is promising, its execution remains an open question, specifically how to automatically create and assess nuggets while ensuring the method's generalizability. This work is inspired by SWAN and addresses these challenges.