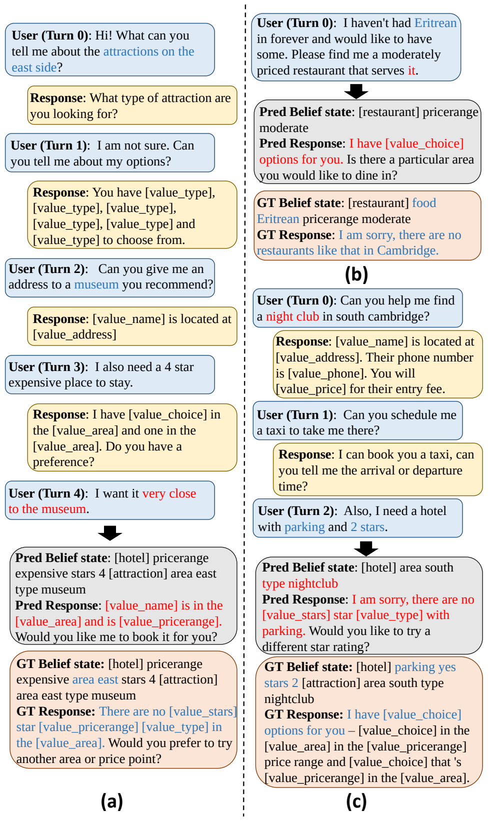
Comet: Dialog Context Fusion Mechanism for End-to-End Task-Oriented Dialog with Multi-task Learning

Haipeng Sun[1]*, **Junwei Bao**[1]†, **Youzheng Wu**[2], **Xiaodong He**[2]
[1]Zuoyebang, Beijing, China [2]JD AI Research, Beijing, China
{sunhaipeng.nlp, baojunwei001}@gmail.com, {wuyouzheng1, hexiaodong}@jd.com

Abstract

Existing end-to-end task-oriented dialog systems often encounter challenges arising from implicit information, coreference, and the presence of noisy and irrelevant data within the dialog context. These issues hinder the system's ability to fully comprehend critical information and lead to inaccurate responses. To address these concerns, we propose Comet, a dialog context fusion mechanism for end-to-end task-oriented dialog, augmented with three supplementary tasks: dialog summarization, domain prediction, and slot detection. Dialog summarization facilitates a more comprehensive understanding of important dialog context information by Comet. Domain prediction enables Comet to concentrate on domain-specific information, thus reducing interference from irrelevant information. Slot detection empowers Comet to accurately identify and comprehend essential dialog context information. Additionally, we introduce a data refinement strategy to enhance the comprehensiveness and recommendability of the generated responses. Experimental results demonstrate the superior performance of our proposed methods compared to existing end-to-end task-oriented dialog systems, achieving state-of-the-art results on the MultiWOZ and CrossWOZ datasets.



User (Turn 0): Hi! What can you tell me about the attractions on the east side?

Response: What type of attraction are you looking for?

User (Turn 1): I am not sure. Can you tell me about my options?

Response: You have [value_type], [value_type], [value_type], [value_type], [value_type] and [value_type] to choose from.

User (Turn 2): Can you give me an address to a museum you recommend?

Response: [value_name] is located at [value_address]

User (Turn 3): I also need a 4 star expensive place to stay.

Response: I have [value_choice] in the [value_area] and one in the [value_area]. Do you have a preference?

User (Turn 4): I want it very close to the museum.

Pred Belief state: [hotel] pricerange expensive stars 4 [attraction] area east type museum
Pred Response: [value_name] is in the [value_area] and is [value_pricerange]. Would you like me to book it for you?

GT Belief state: [hotel] pricerange expensive area east stars 4 [attraction] area east type museum
GT Response: There are no [value_stars] star [value_pricerange] [value_type] in the [value_area]. Would you prefer to try another area or price point?

(a)

User (Turn 0): I haven't had Eritrean in forever and would like to have some. Please find me a moderately priced restaurant that serves it.

Pred Belief state: [restaurant] pricerange moderate
Pred Response: I have [value_choice] options for you. Is there a particular area you would like to dine in?

GT Belief state: [restaurant] food Eritrean pricerange moderate
GT Response: I am sorry, there are no restaurants like that in Cambridge.

(b)

User (Turn 0): Can you help me find a night club in south cambridge?

Response: [value_name] is located at [value_address]. Their phone number is [value_phone]. You will [value_price] for their entry fee.

User (Turn 1): Can you schedule me a taxi to take me there?

Response: I can book you a taxi, can you tell me the arrival or departure time?

User (Turn 2): Also, I need a hotel with parking and 2 stars.

Pred Belief state: [hotel] area south type nightclub
Pred Response: I am sorry, there are no [value_stars] star [value_type] with parking. Would you like to try a different star rating?

GT Belief state: [hotel] parking yes stars 2 [attraction] area south type nightclub
GT Response: I have [value_choice] options for you – [value_choice] in the [value_area] in the [value_pricerange] price range and [value_choice] that 's [value_pricerange] in the [value_area].

(c)

Figure 1: Illustration of issues in the end-to-end task-oriented dialog systems: (a) implicit information, (b) coreference, and (c) noisy information that is irrelevant to the current utterance. The context that is not understood correctly is highlighted in red, whereas the context that should be understood is indicated in blue.

1 Introduction

Task-oriented dialog systems are designed to address fundamental tasks within dialog interactions, such as hotel booking and restaurant reservation. The current approach for these systems involves a pipeline structure with sequential sub-modules for dialog state tracking (Lee et al., 2019), dialog policy (Takanobu et al., 2019), and response generation (Wen et al., 2015). Recent advancements primarily rely on pre-trained models (Radford et al., 2019; Raffel et al., 2020) to establish an end-to-end framework to tackle all tasks, resulting in remarkable performance (Yang et al., 2021; Sun et al., 2023). However, these models occasionally struggle to fully comprehend crucial information within the dialog context, leading to incorrect responses. This is primarily due to issues related to implicit information, coreference, and noisy information, as depicted in Figure 1. For instance, the model fails to grasp the area information of the hotel conveyed implicitly as '*very close to the museum*', resulting

[*] Part of this work was done when Haipeng Sun and Junwei Bao were researchers at JD.

[†] Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 10541–10553
January 19–24, 2025. ©2025 Association for Computational Linguistics

in an incorrect hotel recommendation (Figure 1 (a)). Additionally, the model struggles to resolve referential mentions '*it*' and '*Eritrean*' in user utterances, leading to an erroneous restaurant suggestion (Figure 1 (b)). Moreover, the model becomes confused by noisy information, such as '*night club*', which belongs to the attraction domain but is mistakenly incorporated into the response related to the hotel domain (Figure 1 (c)).

To address the aforementioned issues, we propose Comet, a dialog context fusion mechanism for end-to-end task-oriented dialog with a multi-task learning strategy. Our approach incorporates three tasks: dialog summarization, domain prediction, and slot detection, which aim to preserve crucial content from the dialog history. The fusion mechanism consists of both global and local context fusion mechanisms. Dialog summarization generates a concise summary of the original dialog context, allowing the model to grasp important information more comprehensively through the global context fusion mechanism. Domain prediction assists in identifying the current dialog domain, thereby reducing interference from irrelevant information through the local context fusion mechanism. Slot detection empowers the model to better identify and comprehend essential information within the dialog context. To enhance the system's capability in generating comprehensive and valuable responses, we also devise a data refinement strategy. Our proposed Comet achieves state-of-the-art performance, as demonstrated through experiments conducted on the MultiWOZ and CrossWOZ.

The contributions of this paper are as follows: (1) We conduct an in-depth analysis of the limitations present in current end-to-end task-oriented dialog systems with regards to dialog context understanding. (2) We introduce a novel approach employing a multi-task learning strategy and context fusion mechanism, enhancing the model's ability to comprehend the dialog context more effectively. (3) A data refinement strategy is designed to enable the model to generate responses that are more comprehensive and recommendable. (4) Empirical results demonstrate that our proposed Comet achieves state-of-the-art performance on the MultiWOZ and CrossWOZ datasets.

2 Related Work

The advent of large-scale pre-training models such as BERT (Devlin et al., 2019), GPT (Radford et al., 2019), T5 (Raffel et al., 2020), and GODEL (Peng et al., 2022) have led to the significant advancements in end-to-end task-oriented dialog systems (Hosseini-Asl et al., 2020; Yang et al., 2021; Sun et al., 2023). These systems have demonstrated impressive performance on various multi-domain task-oriented dialog datasets, including MultiWOZ (Budzianowski et al., 2018) and CrossWOZ (Zhu et al., 2020). In fact, the current end-to-end models need to first transfer dialog context into belief and action states before generating the delexicalized[1] system responses. Table 1 presents three strategies for dialog context composition, which are utilized in dialog context modeling.

Model	**Dialog Context Composition**
DAMD	$\{B_{t-1}, R_{t-1}, U_t\}$
LABES	$\{B_{t-1}, R_{t-1}, U_t\}$
BORT	$\{B_{t-1}, R_{t-1}, U_t\}$
MinTL	$\{B_{t-1}, U_{t-2}, R_{t-2}, U_{t-1}, R_{t-1}, U_t\}$
DoTS	$\{U_t, D_{t-1}, B_{t-1}\}$
AuGPT	$\{C_t^*, U_t\}$
SimpleTOD	$\{C_t^*, U_t\}$
SOLOIST	$\{C_t^*, U_t\}$
PPTOD	$\{C_t^*, U_t\}$
UBAR	$\{C_t, U_t\}$
MTTOD	$\{C_t, U_t\}$
GALAXY	$\{C_t, U_t\}$
Mars	$\{C_t, U_t\}$
KRLS	$\{C_t, U_t\}$

Table 1: Dialog context composition strategies of previous end-to-end models. Specifically, we denote t as the dialog turn, U as the user utterance, R as the delexicalized system response, B as the belief state, and D as the domain state. Dialog history C_t, contains the complete dialog information for all previous turns. It is formulated as $\{C_{t-1}, U_{t-1}, B_{t-1}, DB_{t-1}, A_{t-1}, R_{t-1}\}$, where DB represents the database state, and A represents the action state. Another dialog history C_t^*, which contains user and system utterances for all previous turns, excluding previous intermediate states. It is formulated as $\{C_{t-1}, U_{t-1}, R_{t-1}\}$.

DAMD (Zhang et al., 2020b), LABES (Zhang et al., 2020a), and BORT (Sun et al., 2022) utilize the previous belief state B_{t-1}, the previous system response R_{t-1}, and the current user utterance U_t as the dialog context. For MinTL (Lin et al., 2020), the dialog history consists of the user and system utterances from the last two turns. DoTS (Jeon and Lee, 2021) incorporates the previous domain state D_{t-1}. These approaches simplify the dialog

[1] Note that delexicalization, replacing specific slot values in dialog utterances by placeholders, is applied to improve the model's generalization ability (Zhang et al., 2020b).

history, reducing memory usage and computational costs. However, they often struggle to capture sufficient information from the dialog history, facing the problem of error accumulation arising from previously generated inaccurate dialog states.

Recently,there has been significant research interest in developing task-oriented dialog systems that leverage the entire dialog history. Several notable approaches, such as AuGPT (Kulhánek et al., 2021), SimpleTOD (Hosseini-Asl et al., 2020), SOLOIST (Peng et al., 2021), and PPTOD (Su et al., 2021), adopt a strategy where the dialog history C_t^* includes both user and system utterances from all previous turns. Additionally, Yang et al. (2021) argue that incorporating intermediate states such as belief state, action state, and database state from previous turns can further enhance the dialog system performance. To build upon these advancements, systems such as UBAR (Yang et al., 2021), MTTOD (Lee, 2021), GALAXY (He et al., 2022), Mars (Sun et al., 2023), and KRLS (Yu et al., 2023) achieve better performance by considering the entire dialog history C_t. However, these systems that rely on dialog history C_t or C_t^* are susceptible to the presence of noisy and irrelevant information, particularly in multi-domain scenarios. In this paper, we concentrate on leveraging the complete dialog history C_t and propose a multi-task learning and context fusion mechanism to enable the model to better comprehend the dialog context.

3 Methodology

In this section, we initially introduce the multi-task learning strategy, followed by the presentation of two context fusion mechanisms designed to enhance the model's comprehension of the dialog context. Additionally, we outline a data refinement strategy aimed at generating more comprehensive and recommendable responses.

3.1 Multi-task Learning Strategy

In Figure 2 (a), we present a shared encoder-decoder framework designed for an end-to-end task-oriented dialog system, encompassing five tasks: dialog state tracking, response generation, dialog summarization, domain prediction, and slot detection. To accomplish the first three tasks, we employ identical dialog context alongside distinct learnable soft prompt tokens (Lester et al., 2021), as depicted in Figure 2 (a).

3.1.1 Dialog State Tracking

Dialog state tracking is responsible for predicting the belief state, encompassing the dialog domain, slot name, and slot value. The accurate extraction of slot values from the dialog context is vital for successful task completion. During the training process of dialog state tracking, the hidden representation H_{dst} is generated by encoding the prompt token P_{dst}, dialog history C_t, and the current user utterance U_t using a shared encoder. Subsequently, the shared decoder generates the corresponding belief state B_t:

$$\begin{aligned} H_{dst} &= encoder(P_{dst}, C_t, U_t), \\ B_t &= decoder(\widetilde{H_{dst}}), \end{aligned} \tag{1}$$

where $\widetilde{H_{dst}}$ is equal to H_{dst} in the baseline system and $\widetilde{H_{dst}}$ is defined as H_{dst}^{fus} when the context fusion mechanism is applied. A comprehensive explanation of the context fusion mechanism is provided in Section 3.2. The dialog state tracking process is optimized by minimizing the following objective function:

$$\mathcal{L}_{dst} = -logP(B_t|P_{dst}, C_t, U_t). \tag{2}$$

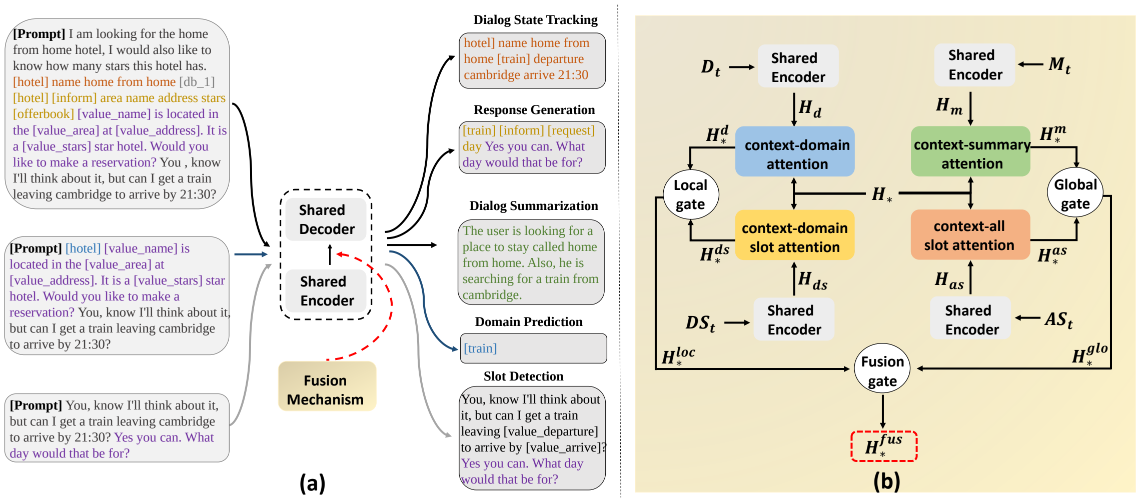
3.1.2 Response Generation

For the response generation task, our approach involves simultaneous generation of the action state and system response. Prior to generating them, we utilize the generated belief state B_t to query the domain-specific database to obtain the database state DB_t, which represents the number of entities that match the user's requirements. The database state DB_t serves as the start token embedding of the shared decoder in response generation task. Given the combination of the prompt token P_{rg}, the dialog history C_t, and the current user utterance U_t, the shared encoder outputs hidden representation H_{rg}. Subsequently, the action state A_t and the system response R_t^{de} are generated sequentially using the shared decoder:

$$\begin{aligned} H_{rg} &= encoder(P_{rg}, C_t, U_t), \\ A_t, R_t &= decoder(DB_t, \widetilde{H_{rg}}), \end{aligned} \tag{3}$$

where $\widetilde{H_{rg}}$ is equivalent to H_{rg} when the model serves as the baseline system and $\widetilde{H_{rg}}$ is defined as H_{rg}^{fus} when the context fusion mechanism is employed. Consequently, the objective function for optimizing the response generation process can be formulated as follows:

$$\mathcal{L}_{rg} = -logP(A_t, R_t|P_{rg}, C_t, U_t, DB_t). \tag{4}$$



[Prompt] I am looking for the home from home hotel, I would also like to know how many stars this hotel has. [hotel] name home from home [db_1] [hotel] [inform] area name address stars [offerbook] [value_name] is located in the [value_area] at [value_address]. It is a [value_stars] star hotel. Would you like to make a reservation? You , know I'll think about it, but can I get a train leaving cambridge to arrive by 21:30?

[Prompt] [hotel] [value_name] is located in the [value_area] at [value_address]. It is a [value_stars] star hotel. Would you like to make a reservation? You, know I'll think about it, but can I get a train leaving cambridge to arrive by 21:30?

[Prompt] You, know I'll think about it, but can I get a train leaving cambridge to arrive by 21:30? Yes you can. What day would that be for?

Dialog State Tracking

hotel] name home from home [train] departure cambridge arrive 21:30

Response Generation

[train] [inform] [request] day Yes you can. What day would that be for?

Dialog Summarization

The user is looking for a place to stay called home from home. Also, he is searching for a train from cambridge.

Domain Prediction

[train]

Slot Detection

You, know I'll think about it, but can I get a train leaving [value_departure] to arrive by [value_arrive]? Yes you can. What day would that be for?

Figure 2: Illustration of our proposed Comet framework. (a) Multi-task learning strategy. User utterances are represented by black tokens, belief states by orange tokens, database states by gray tokens, action states by yellow tokens, system responses by purple tokens, dialog summaries by green tokens, and domain states by blue tokens. (b) Context fusion mechanism. Layer normalization and residual connection are omitted for clarity. The symbol $*$ denotes dst and rg, indicating that the context fusion mechanism is specifically utilized for dialog state tracking and response generation tasks.

3.1.3 Dialog Summarization

Dialog summarization aims to produce a concise and informative abstractive overview that captures the essential information from the original dialog context. This task is crucial for enabling models to comprehend the complete semantic structure of the dialog. However, a significant challenge arises due to the absence of dialog summary labels in existing task-oriented dialog datasets. To address this issue, we employ a dialog template written by Shin et al. (2022) to automatically synthesize dialog summaries from dialog states. We encode the soft prompt token P_{sum} along with the corresponding dialog context $\{C_t, U_t\}$ to generate the dialog summary M_t. Consequently, the dialog summarization process is optimized through the minimization of the following objective function:

$$\mathcal{L}_{sum} = -logP(M_t|P_{sum}, C_t, U_t). \quad (5)$$

3.1.4 Domain Prediction

A comprehensive dialog session typically encompasses multiple domains. Identifying the precise domain enables the model to grasp the pertinent information within the dialog history more effectively. To forecast the domain state D_t for the current turn, we incorporate several source inputs, namely the soft prompt token P_{dom}, the previous domain state D_{t-1}, the previous system response R_{t-1}, and the current user utterance U_t. The optimization of the domain prediction process involves minimizing the following objective function:

$$\mathcal{L}_{dom} = -logP(D_t|P_{dom}, D_{t-1}, R_{t-1}, U_t). \quad (6)$$

3.1.5 Slot Detection

In task-oriented dialog, slot values hold greater significance within the dialog context compared to other elements. The model acquires the capability to identify and comprehend this crucial information from the slot detection task. The model would predict the delexicalized version $\{U^{de}_t, R_t\}$ conditioned on the lexicalized user and system utterance $\{U_t, R^{le}_t\}$ as the source input. The slot detection process is optimized by minimizing the objective function expressed as follows:

$$\mathcal{L}_{sd} = -logP(U^{de}_t, R_t|P_{sd}, U_t, R^{le}_t), \quad (7)$$

where U^{de}_t denotes the delexicalized user utterance, and R^{le}_t denotes the lexicalized system response.

Given that the first three tasks are considerably more intricate compared to the domain prediction and slot detection tasks, we introduce the hyperparameter λ to adjust the weights of these objective functions. To summarize, the optimization of the complete end-to-end task-oriented dialog system can be achieved by minimizing:

$$\mathcal{L}_{all} = \mathcal{L}_{dst} + \mathcal{L}_{rg} + \mathcal{L}_{sum} + \lambda(\mathcal{L}_{dom} + \mathcal{L}_{sd}). \quad (8)$$

(1). User:	Great, thanks! **I am also looking for a 4 star hotel in the north.** Can you please help with that?
System:	There are [value_choice] that meet your criteria. Is there a price range you are interested in?
User:	I would like to keep it in the moderate range, please.
System:	I see [value_choice] options. They are [value_choice] [value_type]. Do you have preferences for internet and parking? If not, I would recommend [value_name].
(2). User:	I think Chinese sounds good, are there any of those in the centre?
System:	Yes. I have [value_choice] places that are [value_food] food. **Would you like to book one?** How about [value_name]?
(3). User:	I am trying to find **information about adc theatre**, can you help me?
System:	[value_name] is in the [value_area] of town on [value_address], the zip code is [value_postcode], would you like their phone number?

Table 2: Examples of refined training data from three aspects. Our newly added tokens are highlighted in red.

3.2 Context Fusion Mechanism

To further enhance the dialog context modeling for the task-oriented dialog system, we propose two context fusion methods: the global context fusion mechanism and the local context fusion mechanism. These methods are designed to improve dialog state tracking and response generation tasks, as depicted in Figure 2 (b).

3.2.1 Global Context Fusion

To enhance dialog state tracking in our approach, we introduce two crucial components: dialog summary and all slot names extracted from the dialog ontology. These additions serve as global knowledge, aiding the model in comprehending the dialog context more effectively. By leveraging dialog summary, the system can acquire crucial details from the dialog context in a more comprehensive manner, thus mitigating the impact of noise, coreference, and other inherent issues in the dialog history. Meanwhile, the slot names serve to direct the model's attention towards the slot values within the dialog history, which plays a vital role in comprehending the user's requirements and accomplishing the corresponding task.

To encode the dialog summary M_t and the sequence of all slot names AS_t, we employ the shared encoder that transforms them into hidden representations H_m and H_{as}, respectively:

$$H_m = encoder(M_t), \tag{9}$$

$$H_{as} = encoder(AS_t). \tag{10}$$

The representations H_m and H_{as} are then combined with the normalized hidden representation of the dialog context, H_{dst}^{LN}, through multi-head attention. This process results in a globally enriched representation, which is obtained by applying a residual connection (He et al., 2016):

$$H_{dst}^{LN} = LayerNorm(H_{dst}), \tag{11}$$

$$H_{dst}^{m} = MultiHead(H_{dst}^{LN}, H_m, H_m) + H_{dst}, \tag{12}$$

$$H_{dst}^{as} = MultiHead(H_{dst}^{LN}, H_{as}, H_{as}) + H_{dst}. \tag{13}$$

To integrate the globally enriched representations H_{dst}^{m} and H_{dst}^{as}, a gate α_g is employed. The estimation of the gate α_g is achieved through the following expression:

$$\alpha_g = \sigma(MLP(H_{dst}^{m}) + MLP(H_{dst}^{as})), \tag{14}$$

where $\sigma(\cdot)$ represents the sigmoid function, MLP refers to the multi-layer perceptron module. The globally fused context representation can be described as follows:

$$H_{dst}^{glo} = \alpha_g H_{dst}^{m} + (1 - \alpha_g) H_{dst}^{as}. \tag{15}$$

3.2.2 Local Context Fusion

To make the model focus on domain-related information within the dialog context and minimize the impact of irrelevant data, we incorporate domain state and domain-specific slot names as supplementary local knowledge. The domain state, denoted as D_t, and the sequence of domain-specific slot names, represented as DS_t, are encoded into hidden representations H_d and H_{ds} using the shared encoder:

$$H_d = encoder(D_t), \tag{16}$$

$$H_{ds} = encoder(DS_t). \tag{17}$$

Similarly, we integrate these two representations H_d and H_{ds} with the normalized context representation H_{dst}^{LN} by employing multi-head attention. This process results in the locally enhanced representation by utilizing a residual connection (He et al., 2016):

$$H_{dst}^{d} = MultiHead(H_{dst}^{LN}, H_d, H_d) + H_{dst}, \tag{18}$$

$$H_{dst}^{ds} = MultiHead(H_{dst}^{LN}, H_{ds}, H_{ds}) + H_{dst}. \tag{19}$$

To integrate the locally enhanced representations H_{dst}^{d} and H_{dst}^{ds}, we employ the gate α_l. The calculation of the gate α_l is as follows:

$$\alpha_l = \sigma(MLP(H_{dst}^{d}) + MLP(H_{dst}^{ds})). \quad (20)$$

Hence, the locally fused context representation can be expressed as:

$$H_{dst}^{loc} = \alpha_l H_{dst}^{d} + (1 - \alpha_l) H_{dst}^{ds}. \quad (21)$$

3.2.3 Both Context Fusion

We combine the globally enriched representation H_{dst}^{glo} with the locally enhanced representation H_{dst}^{loc} to combine the final fused context representation H_{dst}^{fus} using the fusion gate α_f :

$$H_{dst}^{fus} = \alpha_f H_{dst}^{glo} + (1 - \alpha_f) H_{dst}^{loc}, \quad (22)$$

$$\alpha_f = \sigma(MLP(H_{dst}^{glo}) + MLP(H_{dst}^{loc})). \quad (23)$$

The final fused context representation H_{dst}^{fus} is employed instead of the original representation H_{dst} for generating the dialog state in the dialog state tracking task. Similarly, we apply the same context fusion strategy to obtain the fused context representation H_{rg}^{fus} for the response generation task.

During the inference process, the dialog summary and domain state are first generated. Subsequently, they are integrated as additional knowledge using the context fusion mechanism to create the dialog state and system response.

3.3 Data Refinement

We propose a data refinement strategy aimed at enhancing the comprehensiveness and recommendability of the generated responses. Our approach involves modifying the training data from three distinct aspects, as outlined in Table 2. First and foremost, we observe that the system often makes multiple requests when a user is searching for a particular entity. For instance, when a user is looking for a hotel, the system may ask for additional details such as price, internet availability, and parking, without providing a recommendation (Table 2 (1)). To improve user satisfaction and the system's recommendation capability, we propose incorporating a recommendation sentence when the system seeks further details for the second time. Secondly, when the system asks users whether they would like to make a booking, it should also present a recommended entity for their consideration. Lastly, when a user seeks information about a specific entity, the system should deliver a more comprehensive and complete response to fulfill their needs. By implementing these refinements to the training data, we aim to enhance the overall quality of the generated responses, making them more informative and satisfying for users.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate our proposed methods on two task-oriented dialog datasets: MultiWOZ 2.0 (Budzianowski et al., 2018) and CrossWOZ (Zhu et al., 2020). MultiWOZ 2.0 is a large-scale dataset that encompasses seven domains, namely attraction, hospital, police, hotel, restaurant, taxi, and train. The dataset is divided into three subsets, consisting of 8438, 1000, and 1000 dialog sessions for the training, validation, and testing sets, respectively. On the other hand, CrossWOZ (Zhu et al., 2020) is a Chinese dataset that covers five domains, including attraction, restaurant, hotel, taxi, and metro. The dataset is partitioned into 5012, 500, and 500 dialog sessions for training, validation, and testing, respectively.

We evaluate our proposed Comet framework on two benchmark task-oriented dialog tasks: end-to-end dialog modeling for response generation and dialog state tracking. To assess the quality of the generated responses in the task-oriented dialog system on MultiWOZ2.0, we employ the automatic evaluation metrics proposed by Nekvinda and Dušek (2021). **Inform rate** measures whether the dialog system has accurately provided the required entities. **Success rate** evaluates whether the dialog system has successfully answered all the requested information. The BLEU score (Papineni et al., 2002) assesses the fluency of the generated response. To provide an overall evaluation of the dialog system's quality, we compute a **combined score**, which is calculated as $(Inform + Success) \times 0.5 + BLEU$. In addition, we utilize the **joint goal accuracy** to evaluate the performance of dialog state tracking on both MultiWOZ 2.0 and CrossWOZ datasets.

4.2 Settings

Our proposed Comet framework is built upon the MTTOD toolkit (Lee, 2021) and the HuggingFace Transformers library (Wolf et al., 2020). For the MultiWOZ 2.0, we utilize T5-small (Raffel et al., 2020), while for the CrossWOZ, we employ T5-base-Chinese (Zhao et al., 2019). To train the model, we set the batch size to 4 and perform

Model	# Params	Pre-trained	Inform	Success	BLEU	Combined
DAMD (Zhang et al., 2020b)	2M	-	57.9	47.6	16.4	69.2
LABES (Zhang et al., 2020a)	6M	-	68.5	58.1	18.9	82.2
AuGPT (Kulhánek et al., 2021)	117M	GPT-2	76.6	60.5	16.8	85.4
MinTL (Lin et al., 2020)	102M	T5-small	73.7	65.4	19.4	89.0
SOLOIST (Peng et al., 2021)	117M	GPT-2	82.3	72.4	13.6	91.0
DoTS (Jeon and Lee, 2021)	110M	BERT-base	80.4	68.7	16.8	91.4
UBAR (Yang et al., 2021)	81M	DistilGPT2	83.4	70.3	17.6	94.5
PPTOD (Su et al., 2021)	223M	T5-base	83.1	72.7	18.2	96.1
BORT (Sun et al., 2022)	144M	T5-small	85.5	77.4	17.9	99.4
MTTOD (Lee, 2021)	361M	T5-base	85.9	76.5	19.0	100.2
GALAXY (He et al., 2022)	109M	UniLM-base	85.4	75.7	19.6	100.2
Mars (Sun et al., 2023)	102M	T5-small	88.9	78.0	**19.9**	103.4
KRLS (Yu et al., 2023)	361M	GODEL-base	89.2	80.3	19.0	103.8
DiactTOD (Wu et al., 2023)	584M	T5-base	89.5	**84.2**	17.5	104.4
Baseline ($\{B_{t-1}, R_{t-1}, U_t\}$)	61M	T5-small	83.5	73.6	19.3	97.9
Baseline ($\{C_t^*, U_t\}$)	61M	T5-small	85.6	71.8	19.4	98.1
Baseline ($\{C_t, U_t\}$)	61M	T5-small	85.6	76.4	19.5	100.5
Comet* ($\{C_t^*, U_t\}$)	68M	T5-small	86.6	77.9	19.2	101.5
Comet ($\{C_t, U_t\}$)	68M	T5-small	**89.9**	81.3	19.7	**105.3**

Table 3: Comparison of end-to-end models evaluated on MultiWOZ. The results of previous work are reported on the official leaderboard of MultiWOZ (https://github.com/budzianowski/multiwoz).

gradient accumulation every 2 steps. The model parameters are optimized using the AdamW optimizer (Loshchilov and Hutter, 2019) with linear learning rate decay. We set the initial learning rate to 0.0005 and the warm-up ratio to 0.2. To balance the importance of different components, we set the hyper-parameter λ to 0.1 for the end-to-end model and 1 for the dialog state tracking model on the MultiWOZ 2.0. On the CrossWOZ, we set λ to 0.1 for the dialog state tracking model. The sequence length of soft prompt tokens for every task is set to 5. We train all dialog systems on a single NVIDIA A100 GPU for 10 epochs and select the checkpoint model with the best performance on the validation dataset. Our baseline model employs the base architecture of a task-oriented dialog system, trained with dialog state tracking and response generation tasks. We investigate three distinct strategies for composing dialog context, namely $\{B_{t-1}, R_{t-1}, U_t\}$, $\{C_t^*, U_t\}$, and $\{C_t, U_t\}$. These three strategies serve as our respective baselines. In this work, we implement Comet based on the entire dialog context $\{C_t, U_t\}$, and a variant called Comet* based on the dialog context $\{C_t^*, U_t\}$.

4.3 Main Results

Table 3 presents the detailed parameters, pre-trained models, inform rates, success rates, BLEU scores, and combined scores of end-to-end dialog models on the MultiWOZ dataset. Our re-implemented baseline system, which is based on the entire dialog context $\{C_t, U_t\}$, outperforms the other two baselines based on $\{B_{t-1}, R_{t-1}, U_t\}$ and $\{C_t^*, U_t\}$, consistent with our analysis in Section 2. Additionally, our re-implemented baseline performs similarly with GALAXY and MTTOD, indicating that the baseline is a strong system. Our proposed Comet and Comet* consistently outperform the corresponding baseline by 3.4 and 4.8 combined scores, respectively. Furthermore, Comet, using fewer parameters, substantially outperforms the previous state-of-the-art DiactTOD by 0.9 combined scores, achieving state-of-the-art performance in terms of inform rate and combined score. This demonstrates our proposed multi-task learning and context fusion mechanism enable Comet to better comprehend essential information in the dialog context and generate more suitable responses. Further analysis across various dialog turns and domain is provided in Appendix A.

4.4 Dialog State Tracking

Tables 4 and 5 present the dialog state tracking results on MultiWOZ 2.0 and CrossWOZ, respectively. It is worth noting that the previous intermediate states in the dialog history C_t solely consist of belief states for the dialog state tracking model. As shown in Table 4, the baseline based on $\{C_t^*, U_t\}$ performs better than that based on $\{C_t, U_t\}$, which is opposite to the performance of the end-to-end

Model	Joint goal accuracy
Baseline ($\{C_t, U_t\}$)	54.0
Baseline ($\{C_t^*, U_t\}$)	54.5
Comet ($\{C_t, U_t\}$)	54.9
Comet* ($\{C_t^*, U_t\}$)	56.4

Table 4: Dialog state tracking results on MultiWOZ 2.0.

Model	Joint goal accuracy
TRADE (Wu et al., 2019)	36.1
BART-CSP (Moradshahi et al., 2021)	53.6
GEEX (Li et al., 2021)	54.7
Mars (Sun et al., 2023)	59.8
Comet ($\{C_t, U_t\}$)	61.1
Comet* ($\{C_t^*, U_t\}$)	**61.9**

Table 5: Dialog state tracking results on CrossWOZ.

model. This indicates that the inclusion of previous intermediate states does not benefit the dialog state tracking model. Our proposed Comet and Comet* consistently outperform the corresponding baseline by 0.9 and 1.9 points on MultiWOZ 2.0. As shown in Table 5, both Comet and Comet* substantially outperform the previous state-of-the-art Mars by 1.3 and 2.1 points, respectively, resulting in joint goal accuracies of 61.1 and 61.9 on CrossWOZ. Moreover, Comet* achieves the highest joint goal accuracy. These results further demonstrate the effectiveness of our proposed strategies.

4.5 Ablation Study

We empirically investigate the performance of the different components of Comet, as illustrated in Table 6. In comparison to the baseline system and Comet without data refinement, our proposed data refinement strategy yields improvements of 0.4 and 1.7 combined scores, respectively. These results indicate that the utilization of more comprehensive and recommendable responses positively contributes to task completion. Furthermore, through the incorporation of multi-task learning along with data refinement, an additional improvement of 0.4 combined scores is achieved. In terms of the two proposed context fusion mechanisms, global context fusion outperforms local context fusion by a margin of 2.1 in combined scores. This outcome can be attributed to the fact that global context fusion enables Comet to more comprehensively capture essential information from the dialog context. Conversely, local context fusion may have a tendency to overlook certain domain-irrelevant yet potentially valuable details. For instance, certain hotel and restaurant names may be easily disregarded in the context of the taxi domain, but they can serve as essential departure or destination values in the taxi domain. Moreover, the integration of all these components can complement each other to further improve the dialog system performance.

Model	Inform	Success	BLEU	Combined
Baseline ($\{C_t, U_t\}$)	85.6	76.4	19.5	100.5
+DR	86.4	76.5	19.4	100.9
+DR + ML	87.0	76.6	19.5	101.3
+DR + ML + LF	87.1	77.4	19.7	102.0
+DR + ML + GF	88.8	80.2	19.6	104.1
+ML + LF + GF (Comet w/o DR)	89.1	78.8	19.6	103.6
+DR + ML + LF + GF (Comet)	89.9	81.3	19.7	105.3

Table 6: The performance of the different components on MultiWOZ. DR denotes data refinement, ML denotes multi-task learning, LF denotes local context fusion, and GF denotes global context fusion.

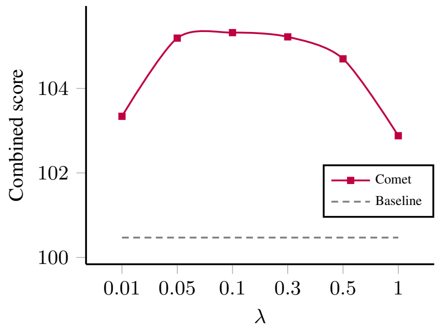


Figure 3: Effect of λ for Comet on MultiWOZ test set.

4.6 Hyper-parameter Analysis

We empirically investigate the influence of the hyper-parameter λ in Equation (8) on the performance of response generation, as illustrated in Figure 3. The selection of λ during multi-task training significantly impacts the roles of domain prediction and slot detection tasks. These tasks assume a more crucial role compared to other loss terms when λ is set to a large value. Conversely, reducing the value of λ diminishes their importance. As illustrated in Figure 3, the performance of Comet exhibits improvements across a wide range of λ values from 0.01 to 1, ultimately achieving optimal results with a well-balanced λ of 0.1.

4.7 Case Study

In order to assess the effectiveness of Comet, we conduct an analysis of response examples. Table 7 presents the predicted example mentioned in Figure 1 (a), while the remaining two examples can be found in Appendix B. The dialog summary produced by Comet contains information about the hotel's location, enabling it to comprehend implicit

User (Turn 0):	Hi! What can you tell me about the attractions on the east side?
Response:	What type of attraction are you looking for?
User (Turn 1):	I am not sure. Can you tell me about my options?
Response:	You have [value_type], [value_type], [value_type], [value_type], [value_type] and [value_type] to choose from.
User (Turn 2):	Can you give me an address to a museum you recommend?
Response:	[value_name] is located at [value_address]
User (Turn 3):	I also need a 4 star expensive place to stay.
Response:	I have [value_choice] in the [value_area] and one in the [value_area]. Do you have a preference?
User (Turn 4):	I want it very close to the museum.
Golden BS:	[hotel] pricerange expensive area east stars 4 [attraction] area east type museum
Golden R:	There are no [value_stars] star [value_pricerange] [value_type] in the [value_area]. Would you prefer to try another area or price point?
Base BS:	[hotel] pricerange expensive stars 4 [attraction] area east type museum
Base R:	[value_name] is in the [value_area] and is [value_pricerange]. Would you like me to book it for you?
Comet SUM:	The user is looking for an attraction which is a museum located in the east. Also, he is searching for a place to stay ranked 4 stars with an expensive price located in the east.
Comet DOM:	[hotel]
Comet BS:	[hotel] pricerange expensive area east stars 4 [attraction] area east type museum
Comet R:	I am sorry, there are no [value_type] in the [value_area]. Would you like to try a different area?

Table 7: Comparison of generated responses between baseline and Comet.

details such as '*very close to the museum*' through the fusion mechanism of dialog summary and context. This illustrates the effectiveness of Comet in comprehending dialog context and generating appropriate system responses in comparison to the baseline model.

5 Conclusion

We first analyze certain shortcomings observed in current end-to-end task-oriented dialog systems regarding dialog context comprehension. To overcome these concerns, we present a multi-task learning strategy and context fusion mechanism. Furthermore, we introduce a data refinement strategy to enhance the model's capability to generate more comprehensive and recommendable responses. Experimental results on the MultiWOZ and CrossWOZ datasets demonstrate that our proposed strategies substantially outperform the original baseline, ultimately achieving state-of-the-art performance.

Limitations

In contrast to the original baseline system, our proposed Comet exhibits certain limitations in terms of training efficiency. The introduction of a multi-task learning strategy and context fusion mechanism leads to an increase in training time and computational costs. However, it is worth noting that we employ a smaller pre-trained model, resulting in our proposed Comet having the fewest number of parameters among the currently available task-oriented dialog systems, as presented in Table 3. Consequently, despite these trade-offs, our proposed Comet still offers a lower computational cost compared to existing task-oriented dialog systems such as MTTOD and GALAXY.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33*, pages 20179–20191. Curran Associates, Inc.

Hyunmin Jeon and Gary Geunbae Lee. 2021. Domain state tracking for a simplified dialogue system. *CoRR*, abs/2103.06648.

Jonás Kulhánek, Vojtech Hudecek, Tomás Nekvinda, and Ondrej Dusek. 2021. Augpt: Dialogue with pre-trained language models and data augmentation. *CoRR*, abs/2102.05126.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483. Association for Computational Linguistics.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinmeng Li, Qian Li, Wansen Wu, and Quanjun Yin. 2021. Generation and extraction combined dialogue state tracking with hierarchical ontology integration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2249. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3391–3405. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*. OpenReview.net.

Mehrad Moradshahi, Victoria Tsai, Giovanni Campagna, and Monica S. Lam. 2021. Contextual semantic parsing for multilingual task-oriented dialogues. *CoRR*, abs/2111.02574.

Tomáš Nekvinda and Ondřej Dušek. 2021. Shades of BLEU, flavours of success: The case of MultiWOZ. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics*, pages 34–46. 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*, pages 311–318. Association for Computational Linguistics.

Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. 2022. Godel: Large-scale pre-training for goal-directed dialog. arXiv.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Jamin Shin, Hangyeol Yu, Hyeongdon Moon, Andrea Madotto, and Juneyoung Park. 2022. Dialogue summaries as dialogue states (DS2), template-guided summarization for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3824–3846, Dublin, Ireland. Association for Computational Linguistics.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *CoRR*, abs/2109.14739.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. BORT: Back and denoising reconstruction for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: NAACL2022*, pages 2156–2170. Association for Computational Linguistics.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2023. Mars: Modeling context & state representations with contrastive learning for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11139–11160. Association for Computational Linguistics.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 100–110. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819. Association for Computational Linguistics.

Qingyang Wu, James Gung, Raphael Shu, and Yi Zhang. 2023. DiactTOD: Learning generalizable latent dialogue acts for controllable task-oriented dialogue systems. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 255–267, Prague, Czechia. Association for Computational Linguistics.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. UBAR: towards fully end-to-end task-oriented dialog system with GPT-2. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 14230–14238. AAAI Press.

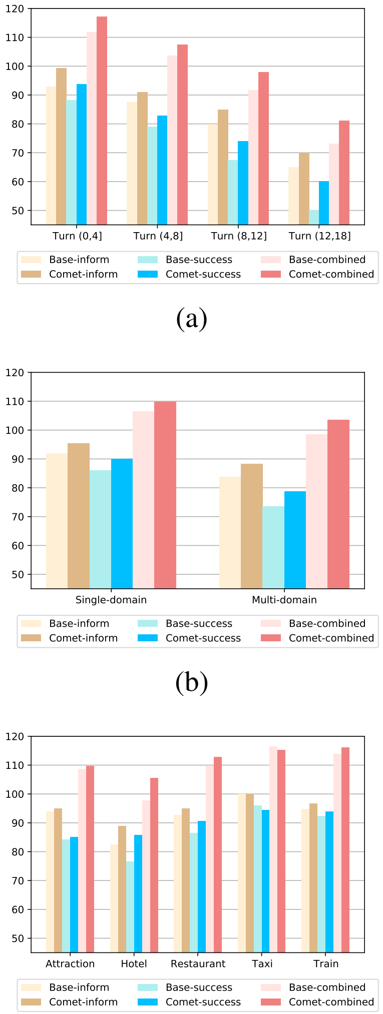
Xiao Yu, Qingyang Wu, Kun Qian, and Zhou Yu. 2023. KRLS: Improving end-to-end response generation in task oriented dialog with reinforced keywords learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12338–12358, Singapore. Association for Computational Linguistics.

Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020a. A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 9207–9219. Association for Computational Linguistics.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020b. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 9604–9611. AAAI Press.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. UER: An open-source toolkit for pre-training models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 241–246. Association for Computational Linguistics.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. CrossWOZ: A large-scale Chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.



(a)

(b)

(c)

Figure 4: Illustration of the dialog system performance with respect to (a) different dialog turns, (b) single-domain vs. multi-domain, and (c) different domains.

A Dialog Turn and Domain Analysis

To provide a comprehensive assessment of the effectiveness of our proposed multi-task learning and context fusion mechanism, we thoroughly examine the performance of Comet and the baseline system across various dialog turns and domains. As depicted in Figure 4(a), we categorize the test sets into four groups based on dialog turns. Notably, in each dialog turn group, Comet consistently outperforms the baseline system, underscoring the robustness of Comet. Particularly in multi-domain dialogs, Comet exhibits a more pronounced improvement compared to single-domain dialogs, as illustrated in Figure 4(b). This observation can be attributed to the fact that the baseline model is more susceptible to the challenges we address in Section 1 when dealing with multi-domain dialogs. However, our proposed Comet effectively mitigates these issues. A similar trend is observed in long-turn dialogs, as depicted in Figure 4(a). Additionally, Figure 4(c) presents the performance across different domains. We observe that Comet achieves a substantial improvement of 7.8 combined scores in the hotel domain. Overall, these findings establish the superior effectiveness of our proposed Comet approach, lending support to its suitability for various dialog turns and domains.

User (Turn 0):	I haven't had Eritrean in forever and would like to have some. Please find me a moderately priced restaurant that serves it.
Golden BS:	[restaurant] food Eritrean pricerange moderate
Golden R:	I am sorry, there are no restaurants like that in Cambridge.
Base BS:	[restaurant] pricerange moderate
Base R:	I have [value_choice] options for you. Is there a particular area you would like to dine in?
Comet SUM:	The user is looking for a restaurant serves eritrean with a moderate price.
Comet DOM:	[restaurant]
Comet BS:	[restaurant] food eritrean pricerange moderate
Comet R:	I am sorry, there are no restaurants that meet your criteria. Would you like to try a different type of food?

Table 8: Comparison of generated responses between baseline and Comet.

User (Turn 0):	Can you help me find a night club in south cambridge?
Response:	[value_name] is located at [value_address]. Their phone number is [value_phone]. You will [value_price] for their entry fee.
User (Turn 1):	Can you schedule me a taxi to take me there?
Response:	I can book you a taxi, can you tell me the arrival or departure time?
User (Turn 2):	Also, I need a hotel with parking and 2 stars.
Golden BS:	[hotel] parking yes stars 2 [attraction] area south type nightclub
Golden R:	I have [value_choice] options for you – [value_choice] in the [value_area] in the [value_pricerange] price range and [value_choice] that 's [value_pricerange] in the [value_area].
Base BS:	[hotel] area south type nightclub
Base R:	I am sorry, there are no [value_stars] star [value_type] with parking. Would you like to try a different star rating?
Comet SUM:	The user is looking for a place to stay ranked 2 stars has parking. Also, he is searching for an attraction which is a nightclub located in the south.
Comet DOM:	[hotel]
Comet BS:	[hotel] parking yes stars 2 [attraction] area south type nightclub
Comet R:	I have [value_choice] options for you. Do you have a price range in mind? If not, I would recommend [value_name].

Table 9: Comparison of generated responses between baseline and Comet.

B More examples

Table 8 illustrates an example of prediction as mentioned in Figure 1 (b). The dialog summary generated by Comet encompasses crucial restaurant-related information. Additionally, the slot detection task employed by Comet aids in the identification and comprehension of the term '*Eritrean*'. Consequently, our proposed Comet model adeptly resolves referential mentions such as '*it*' and '*Eritrean*' within user utterances, owing to the dialog summary and context fusion mechanism. Table 9 presents another prediction example as discussed in Figure 1 (c). The domain prediction task accurately identifies the relevant domain. Through the use of domain prediction and context fusion mechanism, the model effectively disregards extraneous information '*night club*', which pertains to the attraction domain. These findings further validate the effectiveness of Comet in comprehending the dialog context and generating appropriate system responses in comparison to the baseline model.