

# DuetSim: Building User Simulator with Dual Large Language Models for Task-Oriented Dialogues

Xiang Luo, Zhiwen Tang\*, Jin Wang\* and Xuejie Zhang

School of Information Science and Engineering

Yunnan University

Kunming, China

luoxiang@mail.ynu.edu.cn, {zhiwen.tang, wangjin, xjzhang}@ynu.edu.cn

## Abstract

User Simulators play a pivotal role in training and evaluating task-oriented dialogue systems. Traditional user simulators typically rely on human-engineered agendas, resulting in generated responses that often lack diversity and spontaneity. Although large language models (LLMs) exhibit a remarkable capacity for generating coherent and contextually appropriate utterances, they may fall short when tasked with generating responses that effectively guide users towards their goals, particularly in dialogues with intricate constraints and requirements. This paper introduces DuetSim, a novel framework designed to address the intricate demands of task-oriented dialogues by leveraging LLMs. DuetSim stands apart from conventional approaches by employing two LLMs in tandem: one dedicated to response generation and the other focused on verification. This dual LLM approach empowers DuetSim to produce responses that not only exhibit diversity but also demonstrate accuracy and are preferred by human users. We validate the efficacy of our method through extensive experiments conducted on the MultiWOZ dataset, highlighting improvements in response quality and correctness, largely attributed to the incorporation of the second LLM. Our code is accessible at: <https://github.com/suntea233/DuetSim>.

**Keywords:** dialogue system, user simulation, prompt learning

## 1. Introduction

Task-oriented dialogue systems (TODS) are developed to engage with human users in natural language to accomplish tasks within specific domains, such as restaurant reservations and ticket bookings. A common approach to training TODS involves reinforcement learning, where the dialogue system engages with human users over tens of thousands of interactions to collect training dialogues, optimize dialogue policies, and enhance generated responses. However, operating within this training paradigm, which relies on interactions with actual human users, is labor-intensive and cost-prohibitive. To address this challenge, user simulators (US) have been introduced to emulate the behavior of real human users, thereby substituting them in the reinforcement learning training process for TODS.

User simulators play a crucial role in the training of TODS. They not only reduce training costs but also significantly influence the quality of the system's responses, thereby impacting the overall performance of TODS. Given this importance, researchers have long focused on developing user simulators capable of producing accurate, diverse, and human-like responses. Early efforts (Schatzmann et al., 2007; Schatzmann et al., 2005) centered on constructing user simulators using expert knowledge and handcrafted rules. However, rule-

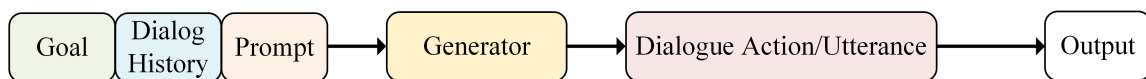
based user simulators face several challenges. One challenge is the complexity of human engineering required, which restricts their suitability to small, well-defined domains and makes them less adaptable for extension to new domains. Another challenge is the limited diversity in their responses, stemming from the determinism of the rules that guide them and the predefined templates used for generating natural language utterances. Although rule-based user simulators can produce correct responses most of the time, researchers have explored alternative approaches to address the limitations of rule-based simulators.

The advent of deep learning (DL) has also given rise to DL-based user simulators. Notable examples of such work include (Asri et al., 2016; Gür et al., 2018; Kreyssig et al., 2018; Lin et al., 2021, 2022). DL-based user simulators offer significant improvements in language variation, but they require extensive training on large volumes of human-annotated task-oriented dialogue data. The quality and diversity of this training data play a pivotal role in determining the accuracy and robustness of these user simulators, which also renders them less suitable for transfer to new domains, particularly when resources are limited in the new domain.

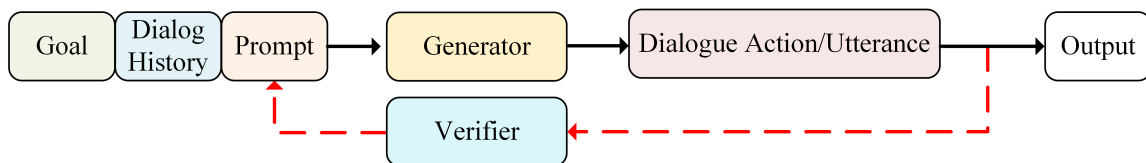
The advent of large language models (LLMs) and in-context learning has opened up new possibilities for constructing user simulators. LLMs are trained on vast amounts of text data in a

---

\*Corresponding Authors



(a) User Simulator with a single LLM for generator



(b) User Simulator with dual LLMs for generator and verifier

Figure 1: Different Architectures for LLM-based User Simulators

self-supervised manner and exhibit impressive zero-shot and few-shot capabilities in downstream tasks, often requiring only a small number of examples to perform well. An intriguing direction is to utilize LLMs as user simulators by providing them with appropriate prompts. This approach circumvents the need for training and fine-tuning, making it more adaptable and faster in accommodating new domains. However, such endeavors (Teragni et al., 2023) are not universally successful. Given the intricate requirements of task-oriented dialogues, LLM-based user simulators struggle to consistently generate responses that satisfy all the criteria specified in the prompts. This study reveals that the challenge arises from LLMs’ limitations in comprehending lengthy prompts. Our findings align with those of (Liu et al., 2023), which highlight that LLMs are more inclined to focus on information at the beginning or end of input, with model performance diminishing as input length increases.

In order to address the aforementioned challenges, this paper introduces a novel framework, DuetSim, designed to tackle the complex requirements of task-oriented dialogues using LLMs. Instead of relying on a single LLM to construct the user simulator, we advocate the use of two LLMs, one for the generator and the other for the verifier, to build a more robust user simulator. The generator initially generates responses based on the dialogue context, while the verifier meticulously examines these responses, offering feedback to the generator if any generated responses are deemed unsuitable. Figure 1 highlights the distinctions between our approach and those employing a single LLM. With the inclusion of the verifier, DuetSim has the capacity to rectify erroneous responses identified by the verifier. Additionally, we incorporate chain-of-thought reasoning to narrow the search space, aiding the user simulator in comprehending the dialogue context and generating contextually

appropriate dialogue responses.

We conducted experiments with our approach using the MultiWOZ dataset. The experimental results clearly demonstrate that our proposed method generates responses that exhibit greater diversity, accuracy, and are preferred by human users. The incorporation of the second LLM significantly enhances the quality and correctness of the generated responses.

The remainder of this paper is structured as follows: Section 2 provides a brief review of related work. In Section 3, we offer a comprehensive description of our model. Section 4 provides a summary of the experimental results and implementation details. Finally, in Section 5, we draw our conclusions.

## 2. Related Work

### 2.1. User Simulator

Task-oriented dialogue system is designed to interact with users in a goal-directed manner to accomplish specific tasks or provide relevant information (Wen et al., 2017; Ham et al., 2020). They differ from chit-chat dialogue systems, which are more focused on free-form conversations and entertaining interactions, while task-oriented dialogue systems are geared towards addressing the specific needs of users.

User simulators also play a crucial role in task-oriented dialog systems. Different from dialogue systems, user simulators have a goal generator to guide them in simulating user behavior and evaluate the performance of the user simulator based on the extent to which the goals are achieved. Typically, constructing a user simulator can be achieved through agenda-based approach (Schatzmann et al., 2005; Schatzmann et al., 2006, 2007; Keizer et al., 2010), where rules are designed based on expert knowledge to sim-

ulate user behavior. Additionally, there are deep learning-based methods that involve training models on corpora of dialogues, resulting in models that don't require hand-crafted policy and have good scalability (Asri et al., 2016; Gür et al., 2018; Kreyszig et al., 2018; Lin et al., 2021, 2022). Different from the methods mentioned above, we directly use large language models to generate dialogue actions and utterances. Through this approach, we do not need to train a model from scratch, and it offers excellent transferability.

## 2.2. Large Language Models and Prompt Learning

In recent years, with the emergence of GPT-3 (Brown et al., 2020), large language models have achieved remarkable performance across various tasks (Zhao et al., 2023; Qiao et al., 2023; Yuan et al., 2022). Additionally, the few-shot and even zero-shot learning capabilities demonstrated by these models have been particularly impressive. As research on prompt learning has evolved, we've seen a progression from initially guiding large language models through simple prompts for downstream tasks to more advanced techniques, which includes step-by-step guided reasoning using the chain of thought approach (Wei et al., 2022), as well as recent developments involving strategies like tree-structured architectures (Yao et al., 2023). These approaches aim to effectively apply large language models to various tasks using different decoding methods (Zheng et al., 2023). Recently, there have also been many efforts to use prompt learning to guide large language models in exploring various downstream tasks such as dialogue system (Hudeček and Dusek, 2023), dialog state tracking (Wang et al., 2022), user simulator (Terragni et al., 2023). As done by Terragni et al. (Terragni et al., 2023), sampling a few examples from the dataset helps large language models better adapt to playing user simulator.

With the deepening research into hard prompts, there has been exploration of approaches like training additional verifier to verify the correctness of the final answer (Cobbe et al., 2021) or intermediate steps (Li et al., 2023; Lightman et al., 2023). Besides, there are methods that involve verifying generated code to improve result accuracy (Zhou et al., 2023). In contrast to these approaches, which either rely on a single large language model to perform all tasks or involve training an additional model to provide feedback, we utilize two large language models, solely harnessing their inference capabilities and facilitating interaction between the generator and the verifier to enhance the applicability of the entire model in user simulation.

## 3. DuetSim

In this paper, we propose DuetSim, a user simulator based on two LLMs for task-oriented dialogue systems. DuetSim consists of a dialogue generator and a response verifier. The dialogue generator drafts the response while the response verifier examines the generated response and offers feedback if needed. Both the dialogue generator and the response verifier are LLMs. By splitting the task between the dialogue generator and the response verifier, both LLMs can share the burden and better handle the assigned tasks. Further, we also propose a chain-of-thought approach to guide the user simulator to generate responses that better fit the context. Rather than directly generating responses in natural language, the proposed approach first generates dialogue acts step-by-step, which are then used to guide the utterance generation.

### 3.1. Prompt Learning for DuetSim

DuetSim leverages the capabilities of large language models (LLMs) with prompt learning. We create prompts for both the dialogue generator and response verifier to elicit responses from LLMs. By supplying LLMs with background information from the ongoing dialogue and previous conversation history, both models can effectively perform their designated tasks. This includes generating appropriate responses in dialogue acts or natural language and verifying whether the generated responses meet the specified requirements.

We employ a zero-shot learning approach to harness the inference capabilities of LLMs. In this approach, we abstain from providing any demonstrations but instead list the requirements and prompt the LLM to generate a response. The prompt for the dialogue generator includes essential information such as the target user goal, denoted as  $G$ , the dialogue context, represented as  $C$ , and the requirements, denoted as  $R_c$ . The user goal  $G$  is randomly generated, and the entire dialogue is designed to fulfill this user goal. The dialogue context  $C$  encompasses utterances from previous dialogue turns, while the requirements  $R^G = [R_1^G, R_2^G, \dots, R_n^G]$  specify the desired properties of the generated response. The task of the dialogue generator is to produce appropriate dialogue acts. Each dialogue act comprises four components: intent, domain, slot, and value, serving as semantic abstractions for the dialogue utterances. This process can be formulated as follows:

$$U^G = \text{Generator}(G, C, R^G) \quad (1)$$

where the  $U^G$  represent the output for dialogue generator. The left side of Figure 2 shows the

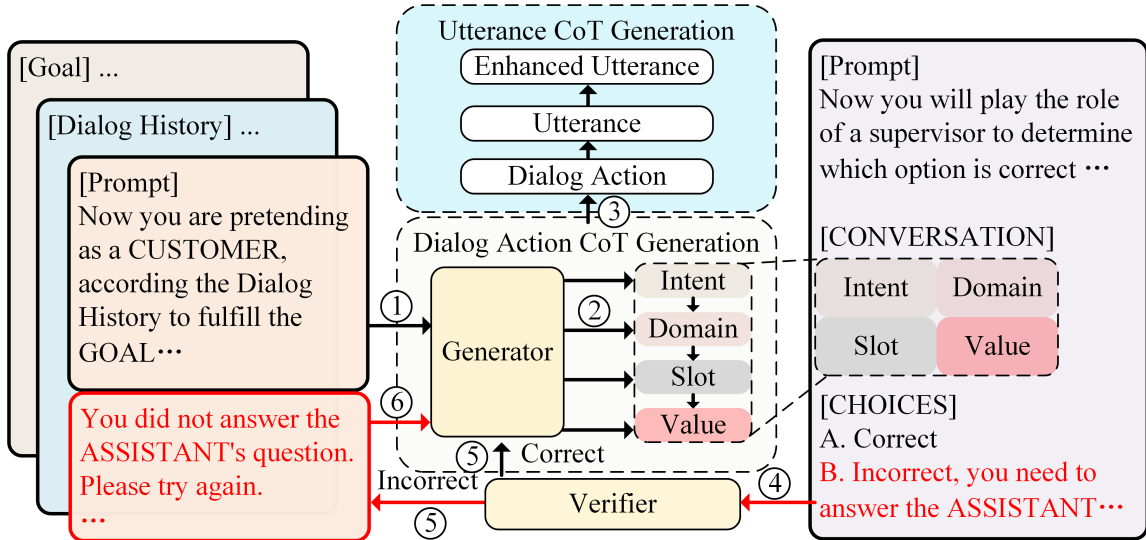


Figure 2: Details of DuetSim.

prompt learning process for the dialogue generator.

The output from the dialogue generator is then passed to the response verifier to assess its alignment with the context. The prompt for the response verifier includes the requirements denoted as  $R_v$ , the dialogue context represented as  $C$ , and the response generated by the dialogue generator, denoted as  $U^G$ . Within the requirement set  $R^V = [R_1^V, R_2^V, \dots, R_m^V]$ , we instruct the verifier LLM to scrutinize the response  $U^G$  for any potential errors. In these requirements, we enumerate common mistakes that the LLM might make, such as context inconsistency and the generation of meaningless words. This prompts the verifier to check for any such errors in the generated response. This process can be formulated as follows:

$$U^V = \text{Verifier}(G, C, R^V, U^G) \quad (2)$$

where the  $U^V$  represent the output for verifier. The right side of Figure 2 shows the prompt learning process for the response verifier.

### 3.2. Generator-Verifier Interaction

In DuetSim, instead of relying on a single LLM for generating dialogue responses, we introduce an additional response verifier to enhance the quality of the generated responses. The dialogue generator initially produces a draft response based on provided prompts. Subsequently, this draft response is forwarded to the response verifier for evaluation against specified requirements. If the verifier deems the draft generated response as suitable, it becomes the output of the user simulator. Conversely, if the verifier detects any errors in the draft response, it rejects the response and

provides feedback to the generator. The generator then proceeds to generate an alternative response.

Figure 2 illustrates the interactive process. In the case of the generator, this process entails inputting carefully crafted prompts into a large language model and leveraging the model’s reasoning capabilities to generate dialogue acts through a chain of thought. Following this, the generated dialogue acts, along with their context, are then transmitted to the verifier.

As for the verifier, it combines the received dialogue acts with pre-defined prompts and inputs them into a large language model. Likewise, the verifier utilizes the reasoning capabilities of the large language model to evaluate the accuracy of the dialogue acts. Subsequently, the verifier offers feedback to the dialogue generator regarding the correctness of the generated response.

If the draft response is found to be incorrect, the feedback will encompass the specific requirement  $R_i^V$  that the draft response has breached. The generator subsequently integrates this feedback information at the end of the prompt and proceeds to generate another response. This iterative process persists until the generator successfully generates the correct dialogue actions or reaches the maximum specified number of iterations. As a result, the final output is formulated as:

$$\widetilde{U}^G = \text{Generator}(G, C, R^G, U^V) \quad (3)$$

<sup>2</sup>PBUS enforces a maximum 15-turn limit for dialogues, unlike other models.



Procedure	Example
Dialog Action ->Utterance	[EXAMPLE] [["inform", "restaurant", "book day", "Tuesday"]] The restaurant is booked on Tuesday.
	[END EXAMPLE] {Dialog Action} Please translate the list into natural language.
Utterance-> Enhanced Utterance	{CONVERSATION} {CONVERSATION} {SENTENCE} {Utterance}
	Based on conversation, play the role of CUSTOMER and rewrite this sentence to make it smoother, more natural, and more conversational

Table 1: Response Generation with CoT

### 3.3. Chain-of-thought Response Generation

To enhance natural language generation, we use a "chain of thought" approach. Initially, we provide guidance through examples, enabling the large language model to effectively translate dialogue actions. As a foundation, we boost fluency, contextual relevance, and naturalness in generated utterances by integrating dialogue context and prompts. This approach yields higher-quality output compared to direct generation. We start by converting dialogue actions into simple utterances using examples and crafted prompts. Subsequently, we input this information, along with the dialogue context, into the large language model, leveraging its reasoning capabilities to produce coherent, context-aware utterances. See Table 1 for an illustrative example prompt.

To enhance dialogue action generation, we employ a step-by-step "chain of thought" approach. Directly generating dialogue actions, including *intent*, *domain*, *slot*, and *value*, can be challenging for large language models due to their limited understanding of abstract dialogue actions. This makes it difficult to produce well-structured actions aligned with intended goals. Thus, we adopt a method that breaks down the generation process into sequential steps for each component of dialogue actions. This approach reduces complexity, assisting the user simulator in producing precise dialogue actions. Specifically, we begin by generating the *intent* using a dedicated prompt. Once obtained, this *intent* serves as input for the *domain* prompt. We continue this sequential process for *slot* and *value*. This iterative approach allows us to construct complete dialogue actions by gradually generating each component.

## 4. Experiments

### 4.1. Dataset

We evaluate DuetSim using the MultiWOZ dataset (Budzianowski et al., 2018) available in ConvLab (Zhu et al., 2020). MultiWOZ is an extensively

annotated dataset comprising 10,000 human-to-human written conversations covering diverse domains and topics, making it a widely used benchmark dataset for evaluating task-oriented dialogue systems.

### 4.2. Evaluation Metrics

We evaluate the proposed method from two perspectives, goal fulfillment and utterance diversity. While goal fulfillment metrics evaluate whether the dialogue helps the user complete the task, utterance diversity metrics focus on the quality of generated natural language.

Goal fulfillment metrics include success rate, completion rate, booking rate, precision, recall and  $F_1$ -score. Success rate refers to the fraction of dialogues that successfully accomplish the user's task. Completion rate evaluates whether the dialogue system make reservations disregarding whether the reserved entity match the user's requirement. Book rate assesses whether booking tasks present in the user's goal have been completed. Precision, recall and  $F_1$ -score are used to determine whether the dialogue system finds the required information.

The utterance diversity are measured with multiple different metrics, including number of unique n-grams (unigrams, bigrams, and trigrams), Shannon Entropy (SE) (Manning and Schutze, 1999), Conditional bigram Entropy (CE) (Manning and Schutze, 1999), Mean Segmental Type-Token Ratio (MSTTR) (Lu, 2012), Measure of Textual Lexical Diversity (MTLD) (McCarthy and Jarvis, 2010) and Hypergeometric Distribution Function (HDD) (Wu, 1993). Among them, SE measures the diversity of information contained in the text. CE assesses the fluency and coherence of text. MSTTR measures the diversity and lexical richness of text. MTLD additionally considers the overall structure of text when measuring the lexical diversity. HDD measures the diversity of vocabulary when randomly selecting a fixed number of words from the text.

In addition, we conducted human evaluation to assess whether responses generated by DuetSim aligns with human preferences.

### 4.3. Implementation Details

All the experiment results are the average of 100 task-oriented dialogues. All the user simulators interact with the dialogue system via dialogue acts. We implement the proposed method using different LLMs, including ChatGPT<sup>1</sup>, FLAN-T5 (Chung et al., 2022), LLAMA2 (Touvron et al., 2023) and ChatGLM2 (Du et al., 2022).

<sup>1</sup><https://openai.com/chatgpt>

User Simulator	Complete Rate	Success Rate	Precision	Recall	$F_1$ -score	Book Rate	Turn
ABUS	<b>0.97</b>	<b>0.97</b>	<b>0.902</b>	<b>0.983</b>	<b>0.924</b>	<b>0.970</b>	10.36
PBUS	0.41	0.30	0.580	0.670	0.710	0.659	<b>7.50<sup>2</sup></b>
DuetSim (ChatGLM2)	0.19	0.25	0.699	0.646	0.644	0.332	17.22
DuetSim (LLAMA2)	0.19	0.28	0.748	0.688	0.687	0.060	17.42
DuetSim (ChatGPT)	0.92	0.74	0.830	0.980	0.881	0.585	16.92
DuetSim (ChatGPT w/o verifier)	0.85	0.67	0.842	0.948	0.873	0.544	17.88
DuetSim (FLAN-T5)	0.92	0.71	0.820	0.979	0.872	0.648	16.16
DuetSim (FLAN-T5 w/o verifier)	0.90	0.63	0.834	0.961	0.875	0.551	16.18

Table 2: Goal Fulfillment Comparison on MultiWOZ dataset

User Simulator	Unigrams	Bigrams	Trigrams	Entropy	CE	MSTTR	HDD	MTLD
ABUS-T	446	1641	2717	6.87	2.39	0.71	0.75	45.97
ABUS-S	413	1514	2386	7.04	2.37	0.76	0.79	62.35
PBUS	949	4440	<b>7176</b>	7.40	<b>3.0</b>	0.70	0.78	45.50
DuetSim (ChatGPT)	1032	2781	3894	7.44	2.62	<b>0.77</b>	0.78	56.98
DuetSim (FLAN-T5)	916	2274	2818	<b>7.58</b>	2.73	0.74	0.79	50.63
DuetSim (ChatGLM2)	1288	3913	5793	7.51	2.31	0.74	<b>0.8</b>	55.28
DuetSim (LLAMA2)	<b>1421</b>	<b>4446</b>	6753	7.11	2.14	0.77	0.75	<b>63.75</b>

Table 3: Utterance Diversity Comparison on MultiWOZ dataset

#### 4.4. Baselines

We compare the proposed method with Agenda-Based User Simulator (ABUS) (Schatzmann et al., 2007) and Prompt-Based User Simulator (PBUS) (Terragni et al., 2023).

- **ABUS:** ABUS is an user simulator that follows hand-crafted rules to complete dialogues. We adopt two variants, namely ABUS-T that uses template NLG and ABUS-S that uses SC-GPT for NLG when evaluating utterance diversity.
- **PBUS:** PBUS is based on a single LLM and uses in-context learning to simulate users. Compared with our approach, PBUS does not involve chain-of-thought reasoning when simulating users.

The proposed method are respectively implemented using the following LLMs:

- **ChatGPT.** ChatGPT is a chatbot developed by OpenAI. It builds upon the foundation of GPT and is trained on large volumes of text data to comprehend and generate natural language text.
- **FLAN-T5.** FLAN-T5 is an instruction-tuned model, trained on extensive datasets, that excels in almost all downstream tasks due to its high versatility.
- **LLAMA2.** LLAMA2 is a model based on LLAMA, and it builds upon the foundation of LLAMA by improving the quality of training data, increasing context length, and optimizing memory related to caching.

- **ChatGLM2.** ChatGLM2 is a model based on ChatGLM. Building on ChatGLM, ChatGLM2 upgrades the base model to enhance performance, increases context length, and improves inference capabilities.

#### 4.5. Main Results

**Goal Fulfillment:** As shown in Table 2, DuetSim performs best in all the user simulators that are based on LLMs and the performance of DuetSim(ChatGPT) is the closest to ABUS. Among all the variants of DuetSim, DuetSim(ChatGPT) and DuetSim(FLAN-T5) perform much better than DuetSim(ChatGLM2) and DuetSim(LLAMA2), which can be attributed to the strong inference capabilities of ChatGPT and FLAN-T5. ChatGPT’s strong inference capabilities are largely due to its large parameter count, whereas for FLAN-T5, the benefits arise from its architecture, data, and instruction fine-tuning, which enable it to also possess good inference capabilities.

We further compare with variants of DuetSim that do not include response verifier. We include all the requirements in the prompt for the single LLM. Results show that performance deteriorate after removing the response verifier from the user simulator. The utterance generation with CoT and the interaction between response verifier and dialogue generator greatly improve the performance of DuetSim.

**Utterance Diversity:** Table 3 presents the performance of different models in terms of utterance diversity using various automated metrics, which demonstrates that our utterance generation mechanism based on CoT significantly improves the language diversity in all metrics. Due to differences

US Model	Complete Rate	Success Rate	Precision	Recall	$F_1$ -score	Book Rate	Turn
DuetSim	0.92	0.74	0.830	0.980	0.881	0.585	16.92
w/o Dialogue history	0.85	0.68	0.793	0.904	0.824	0.626	17.86
w/o Goal	0.91	0.70	0.815	0.980	0.871	0.455	16.52
w/o Both	0.89	0.66	0.821	0.939	0.859	0.584	17.46

Table 4: Ablation results.

User Simulator	Complete Rate	Success Rate	Precision	Recall	$F_1$ -score	Book Rate	Turn
DuetSim w. dialogue context in utterance	<b>0.92</b>	<b>0.74</b>	0.830	<b>0.980</b>	0.881	<b>0.585</b>	16.92
DuetSim w. dialogue context in dialogue act	0.89	0.66	<b>0.853</b>	0.968	<b>0.890</b>	0.523	<b>16.60</b>

Table 5: DuetSim with different forms of dialogue context.

in the training data, architecture, and parameter counts of various models, there are variations in performance among these models as well.

#### 4.6. Ablation Study

To investigate the effectiveness of different components in our prompts, we conducted an ablation experiment on both the goal and dialogue history parts. The results of ablation experiments in Table 4 demonstrate the effectiveness of the two components in our prompts. Removing dialogue history or user goal will result in the worsening of performance since both contains important information for understanding the current status of the dialogue.

We further investigate the impact of different forms of dialogue context, namely dialogue acts or natural language utterances. Results in Table 5 show that DuetSim is better at comprehending dialogue context in the form of natural language, which supports our initial design that uses natural language to express dialogue history.

#### 4.7. Cross-model Evaluation

An ideal user simulator should help the dialogue system obtain better generalization ability. With such observation, we conduct a cross-model evaluation where we train the dialogue system on a user simulator first, then test the dialogue system on another simulator. The training of dialogue system is driven by reinforcement learning algorithm, i.e. proximal policy optimization (PPO) (Schulman et al., 2017). The experiment results are reported in 6.

We observe that the dialogue system training on DuetSim and testing on ABUS performs much better than the one training on ABUS and testing on DuetSim. This indicates that training on DuetSim greatly improves the generalization ability of the dialogue system.

In the meantime, we also observe that, when

	Testing	DuetSim	ABUS
Training			
DuetSim		Complete 0.55 Success 0.46	Complete 0.87 Success 0.83
ABUS		Complete 0.46 Success 0.41	Complete 0.91 Success 0.88

Table 6: Cross-model evaluation results.

training and testing the dialogue system on the same user simulator, dialogue system that interacts with DuetSim perform much worse than the that interacts with ABUS. This indicates that responding to dialogues generated by DuetSim is more challenging than interacting with ABUS.

The above differences come from the fact ABUS is driven by human-engineered agenda, which means the generated response may lack the diversity and stochasticity that are better handled by LLMs. Training the dialogue system using DuetSim is more challenging and but it also improves the generalizability for the dialogue system.

#### 4.8. Human Evaluation

Apart from evaluating the user simulator with automatic metrics, we further conduct human evaluation to study human user’s preference towards different user simulators. The human evaluation involves four user simulators, ABUS-T, ABUS-S, DuetSim (ChatGPT), DuetSim (FLAN-T5).

We recruited 20 human annotators to take part in the experiments. For each user simulator, we first generate 50 task-oriented dialogues by letting the user simulator interact with a rule-based dialogue system. In total, we generated 200 dialogues for human evaluation. And we evaluated dialogues from three dimensions, namely naturalness, informativeness, and coherence. We ask annotators to rate each dialogue among 0 (poor), 1 (average), and 2 (good) for each dimension.

Table 7 reports the average score for the ex-

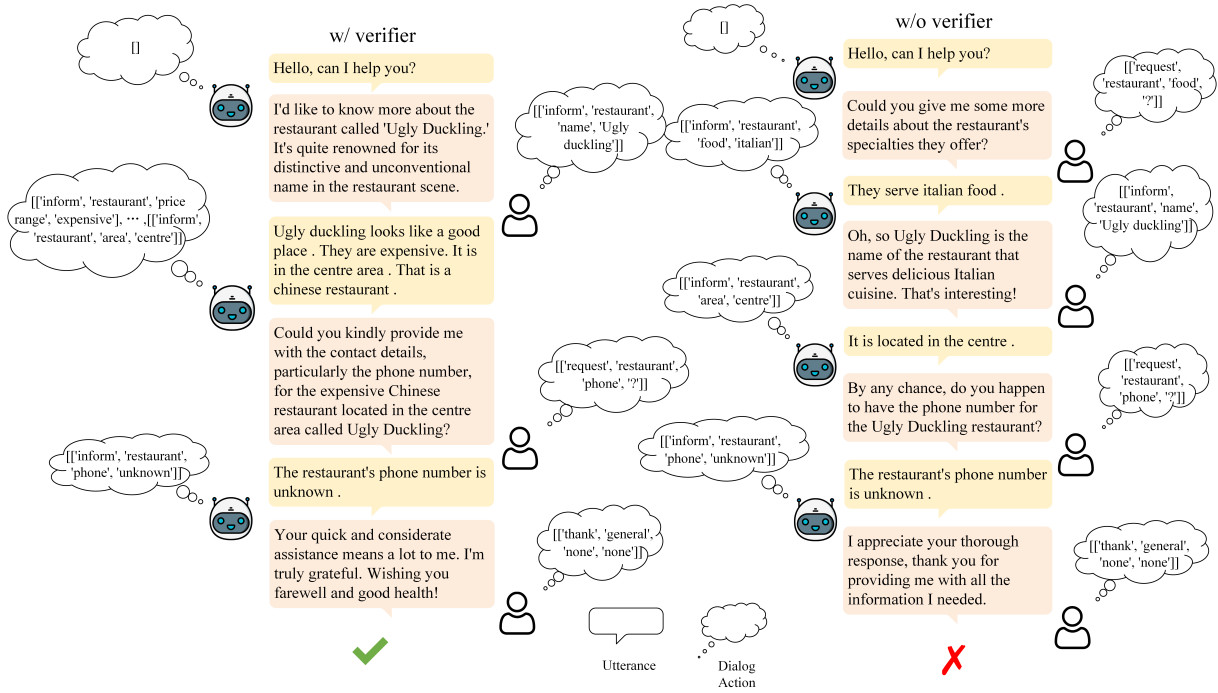


Figure 3: Dialogues generated by the model w/ and w/o verifier.

User Simulator	Naturalness	Informativeness	Coherence
ABUS-T	1.40	1.08	<b>1.38</b>
ABUS-S	1.44	1.32	1.22
DuetSim (ChatGPT)	<b>1.60</b>	<b>1.42</b>	1.36
DuetSim (FLAN-T5)	0.66	0.30	0.52

Table 7: Human preference evaluation on MultiWOZ.

periment. We found that DuetSim (ChatGPT) performs better on naturalness and informativeness compared with all other user simulators. DuetSim (ChatGPT) also performs on par with ABUS-T in terms of coherence. Such results show that DuetSim is able to generate response that better fit human user’s preference.

In the meantime, we observe that DuetSim (FLAN-T5) does not perform well in human evaluation, though it is one of the top performer when evaluating with automatic metrics. When inspecting the output of FLAN-T5, we found that FLAN-T5 often fail to generate proper utterance from dialogue acts. Such results show that the discrepancies between automatic metrics and human evaluation for task-oriented dialogues cannot be neglected.

#### 4.9. Case Study

Fig 3 presents the dialogue of different models when goal is set as follows: “You are looking forward to trying local restaurants. You are looking

for a particular restaurant. Its name is called ugly duckling. Once you find a restaurant, make sure you get phone number. Make sure to ask about what food it serves.” User simulators must meet goal requirements. However, models without a response verifier may make direct food inquiries from a restaurant without specifying the restaurant’s name for their intended dining location. This lack of restaurant information prevents the system from offering the correct details, leading to inaccurate responses, such as identifying the restaurant as Italian when the user wants to reserve a table at a Chinese restaurant. Consequently, a dialogue system lacking restaurant information fails to align with goal requirements, resulting in an unsuccessful dialogue. In contrast, DuetSim, equipped with a response verifier, identifies errors and provides feedback, enabling the generator to focus on specific prompts. As a result, our model can accurately deliver semantic actions, ensuring the system provides the correct information.

## 5. Conclusion

In conclusion, this paper introduces a zero-shot user simulator that is based on dual large language models, which consists of a generator and a verifier. The generator first generates draft response while the verifier examines the draft response and provides feedback. We achieve this by in-context learning and employing chain of thoughts to produce natural and high-quality natural language responses. Empirical experiments



show that our model get competitive results on MultiWOZ.

Future work will focus on extending methods to the multi-modal task-oriented dialogue domain or attempting to address issues with large language models not paying attention to intermediate portions in long-context tasks.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051, the Yunnan University Talent Research Startup Support Project under grant No. CY22623101 and No. CZ22623101, the Postgraduate Research and Innovation Foundation of Yunnan University under Grant No. KC-23234274 and the Exam-Exempted Postgraduate Research and Innovation Foundation of Yunnan University under Grant No. TM-23236972. The authors would like to thank the anonymous reviewers for their constructive comments.

## Bibliographical References

- Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. *arXiv preprint arXiv:1607.00070*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- 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, Brussels, Belgium. Association for Computational Linguistics.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. **GLM: General language model pretraining with autoregressive blank infilling**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.
- Izzeddin Gür, Dilek Hakkani-Tür, Gokhan Tür, and Pararth Shah. 2018. **User modeling for task oriented dialogues**. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 900–906.
- Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. **End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.
- Vojtěch Hudeček and Ondrej Dusek. 2023. **Are large language models all you need for task-oriented dialogue?** In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.
- Simon Keizer, Milica Gašić, Filip Jurčićek, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. **Parameter estimation for agenda-based user simulation**. In *Proceedings of the SIGDIAL 2010 Conference*, pages 116–123, Tokyo, Japan. Association for Computational Linguistics.
- Florian Kreyszig, Iñigo Casanueva, Paweł Budzianowski, and Milica Gašić. 2018. **Neural user simulation for corpus-based policy optimisation of spoken dialogue systems**. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 60–69, Melbourne, Australia. Association for Computational Linguistics.
- Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. **Making language models better reasoners with step-aware verifier**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.
- Hsien-chin Lin, Christian Geishauer, Shutong Feng, Nurul Lubis, Carel van Niekerk, Michael Heck, and Milica Gasic. 2022. [GenTUS: Simulating user behaviour and language in task-oriented dialogues with generative transformers](#). In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 270–282, Edinburgh, UK. Association for Computational Linguistics.
- Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geishauer, Michael Heck, Shutong Feng, and Milica Gasic. 2021. [Domain-independent user simulation with transformers for task-oriented dialogue systems](#). In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 445–456, Singapore and Online. Association for Computational Linguistics.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.
- Xiaofei Lu. 2012. The relationship of lexical richness to the quality of esl learners' oral narratives. *The Modern Language Journal*, 96(2):190–208.
- Christopher Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.
- Philip M McCarthy and Scott Jarvis. 2010. Mtd, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.
- Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. [Reasoning with language model prompting: A survey](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. [Agenda-based user simulation for bootstrapping a POMDP dialogue system](#). In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.
- Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(2):97–126.
- Jost Schatzmann, Matthew N Stuttle, Karl Weilhammer, and Steve Young. 2005. Effects of the user model on simulation-based learning of dialogue strategies. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 220–225. IEEE.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Silvia Terragni, Modestas Filipavicius, Nghia Khau, Bruna Guedes, André Manso, and Roland Mathis. 2023. In-context learning user simulators for task-oriented dialog systems. *arXiv preprint arXiv:2306.00774*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shrutit Bhoale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Qingyue Wang, Yanan Cao, Piji Li, Yanhe Fu, Zheng Lin, and Li Guo. 2022. [Slot dependency modeling for zero-shot cross-domain dialogue state tracking](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 510–520, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- 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*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. [A network-based end-to-end trainable task-oriented dialogue system](#). In *Proceedings*

of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Trong Wu. 1993. An accurate computation of the hypergeometric distribution function. *ACM Transactions on Mathematical Software (TOMS)*, 19(1):33–43.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2022. Hierarchical template transformer for fine-grained sentiment controllable generation. *Information Processing & Management*, 59(5):103048.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921*.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. [ConvLab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149, Online. Association for Computational Linguistics.