Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation

Ryuichi Takanobu[1], **Qi Zhu**[1], **Jinchao Li**[2], **Baolin Peng**[2], **Jianfeng Gao**[2], **Minlie Huang**[1*]
[1]DCST, Institute for AI, BNRist, Tsinghua University, Beijing, China
[2]Microsoft Research, Redmond, USA
[1]{gxly19,zhu-q18}@mails.tsinghua.edu.cn aihuang@tsinghua.edu.cn
[2]{jincli,bapeng,jfgao}@microsoft.com

Abstract

There is a growing interest in developing goal-oriented dialog systems which serve users in accomplishing complex tasks through multi-turn conversations. Although many methods are devised to evaluate and improve the performance of individual dialog components, there is a lack of comprehensive empirical study on how different components contribute to the overall performance of a dialog system. In this paper, we perform a system-wise evaluation and present an empirical analysis on different types of dialog systems which are composed of different modules in different settings. Our results show that (1) a pipeline dialog system trained using fine-grained supervision signals at different component levels often obtains better performance than the systems that use joint or end-to-end models trained on coarse-grained labels, (2) component-wise, single-turn evaluation results are not always consistent with the overall performance of a dialog system, and (3) despite the discrepancy between simulators and human users, simulated evaluation is still a valid alternative to the costly human evaluation especially in the early stage of development.

1 Introduction

Many approaches and architectures have been proposed to develop goal-oriented dialog systems to help users accomplish various tasks (Gao et al., 2019a; Zhang et al., 2020b). Unlike open-domain dialog systems, which are designed to mimic human conversations rather than complete specific tasks and are often implemented as end-to-end systems, a goal-oriented dialog system has access to an external database on which to inquire about information to accomplish tasks for users. Goal-oriented dialog systems can be grouped into three classes based on their architectures, as illustrated in Fig. 1.

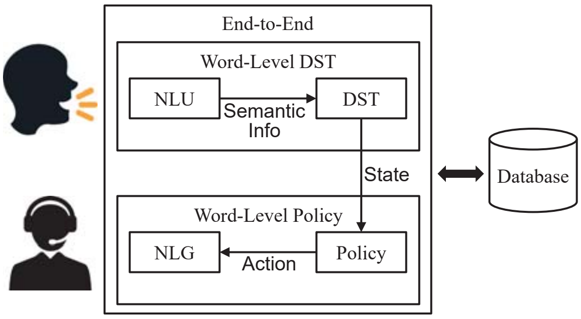

Figure 1: Different architectures of goal-oriented dialog systems. It can be constructed as a pipeline or end-to-end system with different granularity.

The first class is the pipeline (or modular) systems which typically consist of the four components: *Natural Language Understanding* (NLU) (Goo et al., 2018; Pentyala et al., 2019), *Dialog State Tracker* (DST) (Xie et al., 2015; Lee and Stent, 2016), *Dialog Policy* (Peng et al., 2017; Takanobu et al., 2019), and *Natural Language Generation* (NLG) (Wen et al., 2015; Balakrishnan et al., 2019). The second class is the end-to-end (or unitary) systems (Williams et al., 2017; Dhingra et al., 2017; Liu et al., 2018; Lei et al., 2018; Qin et al., 2019; Mehri et al., 2019), which use a machine-learned neural model to generate a system response directly from a dialog history. The third one lies in between the above two types, where some systems use joint models that combine some (but not all) of the four dialog components. For example, a joint word-level DST model combines NLU and DST (Zhong et al., 2018; Wu et al., 2019; Gao et al., 2019b), and a joint word-level policy model combines dialog policy and NLG (Chen et al., 2019; Zhao et al., 2019; Budzianowski and Vulić, 2019).

It is particularly challenging to properly evaluate and compare the overall performance of goal-oriented dialog systems due to the wide variety of system configurations and evaluation settings. Nu-

---

[*] Corresponding author

, pages 23 - 36
July 1-3, 2020 ©2020 Association for Computational Linguistics

merous approaches have been proposed to tackle different components in pipeline systems, whereas these modules are merely evaluated separately. Most studies only compare the proposed models with baselines of the same module, assuming that a set of good modules can always be assembled to build a good dialog system, but rarely evaluate the overall performance of a dialog system from the system perspective. A dialog system can be constructed via different combinations of these modules, but few studies investigated the overall performance of different combinations (Kim et al., 2019; Li et al., 2020). Although end-to-end systems are evaluated in a system-wise manner, none of such systems is compared with its pipeline counterpart. Furthermore, unlike the component-wise assessment, system-wise evaluation requires simulated users or human users to interact with the system to be evaluated via multi-turn conversations to complete tasks.

To this end, we conduct both simulated and human evaluations on dialog systems with a wide variety of configurations and settings using a standardized dialog system platform, Convlab (Lee et al., 2019b), on the MultiWOZ corpus (Budzianowski et al., 2018). Our work attempts to shed light on evaluating and comparing goal-oriented dialog systems by conducting a system-wise evaluation and a detailed empirical analysis. Specifically, we strive to answer the following research questions: (RQ1) Which configurations lead to better goal-oriented dialog systems? (§3.1); (RQ2) Whether the component-wise, single-turn metrics are consistent with system-wise, multi-turn metrics for evaluation? (§3.2); (RQ3) How does the performance vary when a system is evaluated using tasks of different complexities, e.g., from single-domain to multi-domain tasks? (§3.3); (RQ4) Does simulated evaluation correlate well with human evaluation? (§3.4).

Our results show that (1) pipeline systems trained using fine-grained supervision signals at different component levels often achieve better overall performance than the joint models and end-to-end systems, (2) the results of component-wise, single-turn evaluation are not always consistent with that of system-wise, multi-turn evaluation, (3) as expected, the performance of dialog systems of all three types drops significantly with the increase of task complexity, and (4) despite the discrepancy between simulators and human users, simulated evaluation correlates moderately with human evaluation, indicating that simulated evaluation is still a valid alternative to the costly human evaluation, especially in the early stage of development.

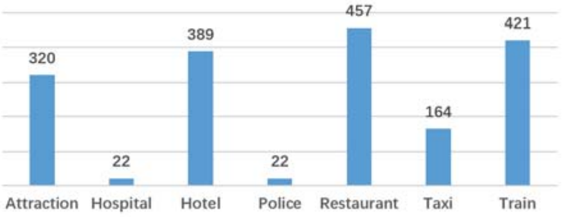

Figure 2: Domain distribution of the user goals used in the experiments. A goal with multiple domains is counted repeatedly for each domain.

2 Experimental Setting

2.1 Data

In order to conduct a system-wise evaluation and an in-depth empirical analysis of various dialog systems, we adopt the MultiWOZ (Budzianowski et al., 2018) corpus in this paper. It is a multi-domain, multi-intent task-oriented dialog corpus that contains 3,406 single-domain dialogs and 7,032 multi-domain dialogs, with 13.18 tokens per turn and 13.68 turns per dialog on average. The dialog states and system dialog acts are fully annotated. The corpus also provides the domain ontology that defines all the entities and attributes in the external databases. We also use the augmented annotation of user dialog acts from (Lee et al., 2019b).

2.2 User Goal

During evaluation, a dialog system interacts with a simulated or human user to accomplish a task according to a pre-defined user goal. A user goal is the description of the state that a user wants to reach in a conversation, containing *indicated constraints* (e.g., a restaurant serving Japanese food in the center of the city) and *requested information* (e.g., the address, phone number of a restaurant).

A user goal is initialized to launch the dialog session during evaluation. To ensure a fair comparison, we apply a fixed set of 1,000 user goals for both simulated and human evaluation. In the goal sampling process, we first obtain the frequency of each slot in the dataset and then sample a user goal from the slot distribution. We also apply additional rules to remove inappropriate combinations, e.g., a user cannot inform and inquire about the arrival time of a train in the same session. In the case

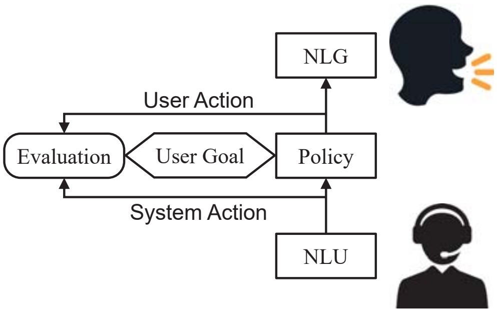

Figure 3: The framework of a user simulator and the mechanism for simulated evaluation.

where no matching database entry exists based on the sampled goal, we resample a new user goal until there is an entity in the database that satisfies the new constraints. In evaluation, the user first communicates with the system based on the initial constraints, and then can change the constraints if the system informs the user that the requested entity is not available. The detailed distribution of these goals is shown in Fig. 2. Among the 1,000 user goals, the numbers of goals involving 1/2/3 domains are 328/549/123, respectively.

2.3 Platform and Simulator

We use the open-source end-to-end dialog system platform, ConvLab (Lee et al., 2019b), as our experimental platform. ConvLab enables researchers to develop a dialog system using preferred architectures and supports system-wise simulated evaluation. It also provides an integration of crowd-sourcing platforms such as Amazon Mechanical Turk for human evaluation.

To automatically evaluate a multi-turn dialog system, Convlab implements an agenda-based user simulator (Schatzmann et al., 2007). Given a user goal, the simulator's policy uses a stack-like structure with complex hand-crafted heuristics to inform its goal and mimics complex user behaviors during a conversation. Since the system interacts with the simulator in natural language, the user simulator directly takes system utterances as input and outputs a user response. The overall architecture of user simulator is presented in Fig. 3. It consists of three modules: NLU, policy, and NLG. We use the default configuration of the simulator in Convlab: a RNN-based model MILU (**M**ulti-**I**ntent **L**anguage **U**nderstanding, extended (Hakkani-Tür et al., 2016)) for NLU, a hand-crafted policy, and a retrieval model for NLG.

2.4 Evaluation Metrics

We use the number of *dialog turns*, averaging over all dialog sessions, to measure the efficiency of accomplishing a task. A user utterance and a subsequent system utterance are regarded as one dialog turn. The system should help each user accomplish his/her goal within 20 turns, otherwise the dialog is regarded as failure. We utilize two other metrics: *inform F1* and *match rate* to estimate the task success. Both metrics are calculated based on the *dialog act* (Stolcke et al., 2000), an abstract representation that extracts the semantic information of an utterance. The dialog act from the input and output of the user simulator's policy will be used to calculate two scores, as shown in Fig. 3. Inform F1 evaluates whether all the information *requests* are fulfilled, and match rate assesses whether the offered entity meets all the *constraints* specified in a user goal. The dialog is marked as successful if and only if both inform recall and match rate are 1.

2.5 System Configurations

To investigate how much system-wise and component-wise evaluations differ, we compare a set of dialog systems that are assembled using different state-of-the-art modules and settings in our experiments. The full list of these systems are shown in Table 1, which includes 4 pipeline systems (*SYSTEM-1∼4*), 10 joint-model systems (*SYSTEM-5∼13*) and 2 end-to-end systems (*SYSTEM-15∼16*). Note that some systems (e.g. SYSTEM-4, SYSTEM-10) generate *delexicalized* responses where the slot values are replaced with their slot names. We convert these responses to natural language by filling the slot values based on dialog acts and/or database query results.

In what follows, we briefly introduce these modules and the corresponding models[1] used in our experiments. The component-wise evaluation results of these modules are shown in Table 2. For published works, we train all the models using the open-source code with the training, validation and test split offered in MultiWOZ, and replicate the performance reported in the original papers or on the leaderboard.

NLU A *natural language understanding* module identifies user intents and extracts associated information from users' raw utterances. We consider

---

[1] All state-of-the-art models mentioned in this paper are based on the open-source code that is available and executable as of February 29, 2020.

two approaches that can handle multi-intents as reference: a RNN-based model MILU which extends (Hakkani-Tür et al., 2016) and is fine-tuned on multiple domains, intents and slots; and a fine-tuned BERT model (Devlin et al., 2019). Following the joint tagging scheme (Zheng et al., 2017), the labels of intent detection and slot filling are annotated for domain classification during training. Both models use dialog history up to the last dialog turn as context. Note that there can be multiple intents or slots in one sentence, we calculate two F1 scores for intents and slots, respectively.

DST A *dialog state tracker* encodes the extracted information as a compact set of dialog state that contains a set of informable slots and their corresponding values (user constraints), and a set of requested slots[2]. We have implemented a rule-based DST to update the slot values in the dialog state based on the output of NLU. We then compare four word-level DST: a multi-domain classifier MDBT (Ramadan et al., 2018) which enumerates all possible candidate slots and values, SUMBT (Lee et al., 2019a) that uses a BERT encoder and a slot-utterance matching architecture for classification, TRADE (Wu et al., 2019) that shares knowledge among domains to directly generate slot values, and COMER (Ren et al., 2019) which applies a hierarchical encoder-decoder model for state generation. We use two metrics for evaluation. The joint goal accuracy compares the predicted dialog states to the ground truth at each dialog turn, and the output is considered correct if and only if all the predicted values exactly match the ground truth. The slot accuracy individually compares each (domain, slot, value) triplet to its ground truth label.

Policy A *dialog policy* relies on the dialog state provided by DST to select a system action. We compare two dialog policies: a hand-crafted policy, and a reinforcement learning policy GDPL (Takanobu et al., 2019) that jointly learns a reward function. We also include in our comparison three joint models, known as word-level policies, which combine the policy and the NLG module to produce natural language responses from dialog states. They are MDRG (Wen et al., 2017) where an attention mechanism is conditioned on the dialog states, HDSA (Chen et al., 2019) that decodes response from predicted hierarchical dialog acts, and LaRL (Zhao et al., 2019) which uses a latent action framework. We use BLEU score (Papineni et al., 2002), inform rate and task success rate as metrics for evaluation. Note that the inform rate and task success for evaluating policies are computed at the turn level, while the ones used in system-wise evaluation are computed at the dialog level.

NLG A *natural language generation* module generates a natural language response from a dialog act representation. We experiment with two models: a retrieval-based model that samples a sentence randomly from the corpus using dialog acts, and a generation-based model SCLSTM (Wen et al., 2015) which appends a sentence planning cell in RNN. To evaluate the performance of NLG, we adopt BLEU score to evaluate the quality of the generated text, and slot error rate (SER) to measure whether the generated response contains missing or redundant slot values.

E2E An *end-to-end* model takes user utterances as input and directly output system responses in natural language. We experiment with two models: TSCP (Lei et al., 2018) that uses belief spans to represent dialog states, and DAMD (Zhang et al., 2020a) that further uses action spans to represent dialog acts as additional information. For single-turn evaluation, BLEU, inform rate and success rate are provided.

3 Empirical Analysis

3.1 Performance under Different Settings (RQ1)

We compare the performance of three types of systems, pipeline, joint-model and end-to-end. Results in Table 1 show that pipeline systems often achieve better overall performance than the joint models and end-to-end systems because using fine-grained labels at the component level can help pipeline systems improve the task success rate.

NLU with DST or joint DST It is essential to predict dialog states to determine what a user has expressed and wants to inquire. The dialog state is used to query the database, predict the system dialog act, and generate a dialog response. Although many studies have focused on the word-level DST that directly predicts the state using the

---

[2] Dialog state can include everything a system must know in order to make a decision about what to do next, e.g., DSTC2 corpus (Henderson et al., 2014) contains *search method* representing user intents in the dialog state, but only aforementioned items are taken into account as our experiments are conducted on MultiWOZ in this paper.

ID	Configuration				Turn	Inform			Match	Succ.
	NLU	DST	Policy	NLG		Prec.	Rec.	F1		
1	BERT	rule	rule	retrieval	6.79	0.79	0.91	0.83	90.54	80.9
2	MILU	rule	rule	retrieval	7.24	0.76	0.88	0.80	87.93	77.6
3	BERT	rule	GDPL	retrieval	10.86	0.72	0.69	0.69	68.34	54.1
4	BERT	rule	rule	SCLSTM	13.38	0.64	0.58	0.58	51.41	43.0
5	MDBT		rule	retrieval	16.55	0.47	0.35	0.37	39.76	18.8
6	SUMBT		rule	retrieval	13.71	0.51	0.44	0.44	46.44	27.8
7	TRADE		rule	retrieval	9.56	0.39	0.41	0.37	38.37	22.4
8	COMER		rule	retrieval	16.79	0.30	0.28	0.28	29.06	17.3
9	BERT	rule	MDRG		17.90	0.35	0.34	0.32	29.07	19.2
10	BERT	rule	HDSA		15.91	0.47	0.62	0.50	39.21	34.3
11	BERT	rule	LaRL		13.08	0.40	0.68	0.48	68.95	47.7
12	SUMBT		HDSA		18.67	0.27	0.32	0.26	14.78	13.7
13	SUMBT		LaRL		13.92	0.36	0.64	0.44	57.63	40.4
14	TRADE		LaRL		14.44	0.35	0.57	0.40	36.07	30.8
15	TSCP				18.20	0.37	0.32	0.31	13.68	11.8
16	DAMD				11.27	0.64	0.69	0.64	59.67	48.5

Table 1: System-wise simulated evaluation with different configurations and models. We use *SYSTEM-<ID>* to represent the configuration's abbreviation throughout the paper.

user query, we also investigate the cascaded configuration where an NLU model is followed by a rule-based DST. As shown in Table 1, the success rate has a sharp decline when using word-level DST, compared to using an NLU model followed by a rule-based DST (17.3%∼27.8% in *SYSTEM-(5∼8)* vs. 80.9% in *SYSTEM-1*). The main reason is that the dialog act predicted by NLU contains both slot-value pairs and **user intents**, whereas the dialog state predicted by the word-level DST only records the user constraints in the current turn, causing information loss for action selection (via dialog policy) as shown in Fig. 4. For example, a user may want to confirm the booking time of the restaurant, but such an intent cannot be represented in the slot values. However, we can observe that word-level DST achieves better overall performance by combining with word-level policy, e.g., 40.4% success rate in *SYSTEM-13* vs. 27.8% in *SYSTEM-6*. This is because word-level policy implicitly detects user intents by encoding the **user utterance** as additional input, as presented in Fig. 5. Neverthsless, all those joint approaches still under-perform traditional pipeline systems.

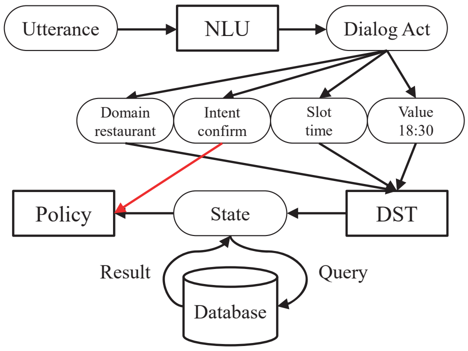

Figure 4: Illustration of NLU and DST in the dialog system. The intent information (red arrow) is missing in the dialog state on MultiWOZ if the system merges a word-level DST with a dialog policy.

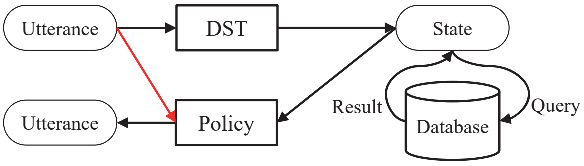

Figure 5: The common architecture of a system using word-level or end-to-end models. User utterances are encoded again (red arrow) for response generation.

NLG from dialog act or state We compare two strategies for generating responses. One is based on an ordinary NLG module that generates a response according to dialog act predicted by dialog policy. The other uses the word-level policy to directly generates a natural language response based on dialog state and user query. As we can see in Table 1 that the performance drops substantially when we replace the retrieval NLG module with a joint model such as *MDRG* or *HDSA*. This indicates that the dialog act has encoded sufficient semantic information so that a simple retrieval NLG module can give high-quality replies. However, the fact, that *SYSTEM-11* which uses word-level policy *LaRL* even outperforms *SYSTEM-4* which uses the NLG model *SCLSTM* in task success (47.7% vs. 43.0%), indicates that response generation can be improved by jointly training policy and NLG modules.

Database query As part of dialog management, it is crucial to identify the correct entity that satisfies the user goal. MultiWOZ contains a large number of entities across multiple domains, making it impossible to explicitly learn the representations of all the entities in the database as previous work did (Dhingra et al., 2017; Madotto et al., 2018). This requires the designed system to deal with a large-scale external database, which is closer to reality. It can be seen in Table 1 that most joint models have a lower match rate than the pipeline systems. In particular, *SYSTEM-15* rarely selects an appropriate entity during the dialog (13.68% match rate) since the proposed *belief spans* only copy the values from utterances without knowing which domain or slot type the values belong to. Due to the poor performance in dialog state prediction, it cannot consider the external database selectively, thereby failing to satisfy the user's constraints. In comparison, *SYSTEM-16* has achieved the highest success rate (48.5%) and the second-highest match rate (59.67%) among all the systems using joint models (*SYSTEM-5*∼*14*). This is because *DAMD* utilizes *action spans* to predict both user and system dialog acts in addition to belief spans, which behaves like a pipeline system. This indicates that an explicit dialog act supervision can improve dialog state tracking.

3.2 Component-wise vs. System-wise Evaluation (RQ2)

It is important to verify whether the component-wise evaluation is consistent with system-wise evaluation. By comparing the results in Table 1 and Table 2, we can observe that sometimes they are consistent (e.g., *BERT* > *MILU* in Table 2a, and *SYSTEM-1* > *SYSTEM-2*), but not always (e.g., *TRADE* > *SUMBT* in Table 2b, but *SYSTEM-6* > *SYSTEM-7*).

In general, a better NLU model leads to a better multi-turn conversation, and *SYSTEM-1* outperforms all other configurations in completing user goals. With respect to DST, though word-level DST models directly predict dialog states without explicitly detecting user intents, most of them perform poorly in terms of joint accuracy as shown in Table 2b. This severely harms the overall performance because the downstream tasks strongly rely on the predicted dialog states. Interestingly, *TRADE* has higher accuracy than *SUMBT* on DST. But *TRADE* performs worse than *SUMBT* in system-wise evaluation (22.4% in *SYSTEM-7* vs. 27.8% in *SYSTEM-6*). The observation is similar to *COMER* vs. *TRADE*. This indicates that the results of component-wise evaluation in DST are not consistent with those of system-wise evaluation, which may be attributed to the noisy dialog state annotations (Eric et al., 2019).

Model	Slot	Intent	Overall
MILU	81.90	85.82	83.27
BERT	84.25	89.84	86.21

(a) NLU

Model	Slot Acc.	Joint Acc.
MDBT†	89.53	15.57
SUMBT†	96.44	46.65
TRADE†	96.92	48.62
COMER	95.52	48.79

(b) Word-level DST

Model	BLEU	Inform	Succ.
MDRG†	18.8	71.3	61.0
HDSA†	23.6	82.9	68.9
LaRL†	12.8	82.8	79.2

(c) Word-level Policy

Model	BLEU	SER
Retrieval	33.1	–
SCLSTM	51.6	3.10

(d) NLG

Model	BLEU	Inform	Succ.
TSCP	15.5	66.4	45.3
DAMD	16.6	76.3	60.4

(e) E2E

Table 2: Component-wise performance of each module. †: results from the MultiWOZ leaderboard.

As for word-level policy, *HDSA* that uses explicit dialog acts in supervision has higher BLEU than *LaRL* that uses latent dialog acts, but *LaRL* that is finetuned with reinforcement learning has much higher match rate than *HDSA* in system-wise evaluation (68.95% vs. 39.21%). Although there is small difference between *MDRG* and *HDSA* in component-wise evaluation (61.0% vs. 68.9% in

ID	Restaurant				Train				Attraction		
	Turn	Info.	Match	Succ.	Turn	Info.	Match	Succ.	Turn	Info.	Succ.
1	2.82	0.94	96.9	98	3.06	1.0	100	100	3.12	0.69	63
2	2.84	0.92	100	98	2.99	1.0	94.2	97	3.70	0.73	65
3	8.68	0.70	69.4	70	6.07	0.80	67.3	75	5.61	0.67	62
4	6.00	0.77	68.8	78	11.53	0.71	67.3	55	12.57	0.57	46
6	9.41	0.64	72.7	60	5.13	0.97	90.4	93	14.79	0.23	9
11	9.91	0.39	66.7	61	4.02	0.86	88.5	97	4.73	0.68	80
13	8.35	0.40	65.6	60	4.19	0.85	94.2	96	6.06	0.60	73
15	14.72	0.37	11.5	27	16.02	0.46	11.5	25	16.12	0.51	24
16	6.36	0.80	92.2	90	10.21	0.61	55.8	58	8.32	0.69	67

Table 3: Performance with different single domain. Most systems achieve better performance in *Restaurant* and *Train* than *Attraction*.

ID	Single				Two				Three			
	Turn	Info.	Match	Succ.	Turn	Info.	Match	Succ.	Turn	Info.	Match	Succ.
1	3.22	0.84	84.7	87	6.96	0.81	94.9	78	8.15	0.82	88.4	69
2	3.90	0.78	79.7	82	6.74	0.76	95.3	72	10.54	0.79	85.0	66
3	9.18	0.67	66.7	60	12.38	0.60	42.9	42	13.55	0.50	44.6	21
4	8.65	0.66	58.3	62	17.24	0.38	28.0	14	18.03	0.46	24.4	13
6	10.35	0.44	60.4	41	14.74	0.44	50.9	17	15.97	0.25	20.9	0
11	8.79	0.45	72.2	55	13.37	0.52	74.0	59	19.30	0.39	50.4	0
13	8.48	0.45	62.5	61	14.08	0.45	61.0	47	18.95	0.36	40.7	0
15	15.09	0.33	10.0	26	19.10	0.25	17.8	8	20.00	0.19	0.0	1
16	8.89	0.66	68.1	65	13.48	0.52	57.1	34	18.59	0.58	45.5	12

Table 4: Performance with different number of domains. All systems have performance drop as the number of domains increases.

Table 2c), the gap is increased (19.2% in *SYSTEM-9* vs. 34.3% in *SYSTEM-10*) in system-wise evaluation. In addition, even *SCLSTM* achieves a higher BLEU score than the retrieval-based model (51.6% vs. 33.1% in Table 2d), it only obtains a lower success rate (43.0% in *SYSTEM-4* vs. 80.9% in *SYSTEM-1*) when assembled with other modules. These results show again the discrepancy between component-wise and system-wise evaluation. The superiority of the systems using retrieval models may imply that lower SER in NLG is more critical than higher BLEU in goal-oriented dialog systems.

Error in multi-turn interactions Most existing work only evaluates the model with single-turn interactions. For instance, *inform rate* and *task success* at each dialog turn are computed given the current user utterance, dialog state and database query results for context-to-context generation (Wen et al., 2017; Budzianowski and Vulić, 2019). A strong assumption is that the model would be fed with the ground truth from the upstream modules or the last dialog turn. However, this assumption does not hold since a goal-oriented dialog consists of a sequence of associated inquiries and responses between the system and its user, and the system may produce erroneous output at any time. The errors may propagate to the downstream modules and affect the following turns. For instance, end-to-end models get worse success rate in multi-turn interactions than in single-turn evaluation in Table 2e. A sample dialog from *SYSTEM-1* and *SYSTEM-6* is provided in Table 6. *SYSTEM-6* does not extract the *pricerange* slot (highlighted in red color) correctly. The incorrect dialog state further harms the performance of dialog policy, and the conversation gets stuck where the user (simulator) is always asking for the postcode, thereby failing to complete the task.

To summarize, the component-wise, single-turn evaluation results do not reflect the real performance of the system well, and it is essential to evaluate a dialog system in an end-to-end, interactive setting.

3.3 Performance of Task with Different Complexities (RQ3)

With the increasing demands to address various situations in multi-domain dialog, we choose 9 representative systems across different configurations and approaches to further investigate how their performance varies with the complexities of the tasks. 100 user goals are randomly sampled under each domain setting. Results in Table 3 and 4 show that the overall performance of all systems varies with different task domains and drops significantly with the increase of task complexity, while pipeline systems are relatively robust to task complexity.

Performance with different single domains Table 3 shows the performance with respect to different single domains. *Restaurant* is a common domain where users inquiry some information about a restaurant and make reservations. *Train* has more entities and its domain constraints can be more complex, e.g., the preferred train should *arrive before* 5 p.m. *Attraction* is an easier one where users do not make reservations. There are 7/6/3 informable slots that need to be tracked in *Restaurant*/*Train*/*Attraction* respectively. Surprisingly, most systems perform better in *Restaurant* or *Train* than *Attraction*. This may result from the noise database in *Attraction* where *pricerange* information is missing sometimes, and from the uneven data distribution where *Restaurant* and *Train* appear more frequently in the training set. In general, pipeline systems perform more stably across multiple domains than joint models and end-to-end systems.

Performance with different number of domains Table 4 demonstrates how the performance varies with the number of domains in a task. We can observe that most systems fall short to deal with multi-domain tasks. Though some systems such as *SYSTEM-13* and *SYSTEM-16* can achieve a relatively high inform F1 or match rate for a single domain, the overall success rate drops substantially on two-domain tasks, and most systems fail to complete three-domain tasks. The number of dialog turns also increases remarkably when the number of domains increases. Among all these configurations, only the pipeline systems *SYSTEM-2* and *SYSTEM-1* can keep a high success rate when there are three domains in a task. These results show that current dialog systems are still insufficient to deal with complex tasks, and that pipeline systems outperform joint models and end-to-end systems.

3.4 Simulated vs. Human Evaluation (RQ4)

Since the ultimate goal of a task-oriented dialog system is to help users accomplish real-world tasks, it is essential to justify the correlation between simulated and human evaluation. For human evaluation, 100 Amazon Mechanical Turk workers are hired to interact with each system and then give their judgement on task success. The ability of Language Understanding (LU) and Response Appropriateness (RA) of the systems are assessed at the same time, and each worker gives a score on these two metrics with a five-point scale. We compare 5 systems that achieve the best performance in the simulated evaluation under different settings.

ID	Turn	LU	RA	Succ.	Corr.
1	18.58	3.62	3.69	62	0.57
6	20.63	2.85	2.91	27	0.72
11	19.98	2.36	2.41	23	0.53
13	19.26	2.17	2.49	14	0.46
16	16.33	2.61	2.65	23	0.55

Table 5: System-wise evaluation with human users. Correlation coefficient between simulated and human evaluation is presented in the last column.

Table 5 shows the human evaluation results of 5 dialog systems. Comparing with the simulated evaluation in Table 1, we can see that Pearson's correlation coefficient lies around 0.5 to 0.6 for most systems, indicating that simulated evaluation correlates moderately well with human evaluation. Similar to simulated evaluation, the pipeline system *SYSTEM-1* obtains the highest task success rate in human evaluation. A sample human-machine dialog from *SYSTEM-1* and *SYSTEM-6* is provided in Table 7. The result is similar to the simulated session in Table 6 but *SYSTEM-6* fails to respond with the *phone* number in Table 7 instead (highlighted in red color). All these imply the reliability of the simulated evaluation in goal-oriented dialog systems, showing that simulated evaluation can be a valid alternative to the costly human evaluation for system developers.

However, compared to simulated evaluation, we can observe that humans converse more naturally than the simulator, e.g., the user confirms with *SYSTEM-1* whether it has booked 7 seats in Table 7, and most systems have worse performance in human evaluation. This indicates that there is still a gap between simulated and human evaluation. This is due to the discrepancy between the corpus and human conversations. The dataset only contains limited human dialog data, on which the user simulator is built. Both the system and the simulator are hence limited by the training corpus. As a result, the task success rate of most systems decreases significantly in human evaluation, e.g., from 40.4% to 14% in *SYSTEM-13*. This indicates that existing dialog systems are vulnerable to the variation of human language (e.g., the sentence highlighted in brown in Table 7), which demonstrates a lack of robustness in dealing with real human conversations.

4 Related Work

Developers have been facing many problems when evaluating a goal-oriented dialog system. A range of well-defined automatic metrics have been designed for different components in the system, e.g., joint goal accuracy in DST and task success rate in policy optimization introduced in Table 2b and 2c. A broadly accepted evaluation scheme for the goal-oriented dialog was first proposed by PARADISE (Walker et al., 1997). It estimates the user satisfaction by measuring two types of aspects, namely *dialog cost* and *task success*. Paek (2001) suggests that a useful dialog metric should provide an estimate of how well the goal is met and allow for a comparative judgement of different systems. Though a model can be optimized against these metrics via supervised learning, each component is trained or evaluated separately, thus difficult to reflect real user satisfaction.

As human evaluation by asking crowd-sourcing workers to interact with a dialog system is much expensive (Ultes et al., 2013; Su et al., 2016) and prone to be affected by subjective factors (Higashinaka et al., 2010; Schmitt and Ultes, 2015), researchers have tried to realize automatic evaluation of dialog systems. Simulated evaluation (Araki and Doshita, 1996; Eckert et al., 1997) is widely used in recent works (Williams et al., 2017; Peng et al., 2017; Takanobu et al., 2019, 2020) and platforms (Ultes et al., 2017; Lee et al., 2019b; Papangelis et al., 2020; Zhu et al., 2020), where the system interacts with a user simulator which mimics human behaviors. Such evaluation can be conducted at the dialog act or natural language level. The advantages of using simulated evaluation are that it can support multi-turn language interaction in a full end-to-end fashion and generate dialogs unseen in the original corpus.

5 Conclusion and Discussion

In this paper, we have presented the system-wise evaluation result and empirical analysis to estimate the practicality of goal-oriented dialog systems with a number of configurations and approaches. Though our experiments are only conducted on MultiWOZ, we believe that such results can be generalized to all goal-oriented scenarios in dialog systems. We have the following observations:

1) We find that rule-based pipeline systems generally outperform state-of-the-art joint systems and end-to-end systems, in terms of both overall performance and robustness to task complexity. The main reason is that fine-grained supervision on dialog acts would remarkably help the system plan and make decisions, because the system should predict the user intent and take proper actions during the conversation. This supports that good pragmatic parsing (e.g. dialog acts) is essential to build a dialog system.

2) Results show that component-wise, single-turn evaluation results are not always consistent with the overall performance of dialog systems. In order to accurately assess the effectiveness of each module, system-wise, multi-turn evaluation should be used from the practical perspective. We advocate assembling the proposed model of a specific module into a complete system, and evaluating the system with simulated or human users via a standardized dialog platform, such as Rasa (Bocklisch et al., 2017) or ConvLab. Undoubtedly, this will realize a full assessment of the module's contribution to the overall performance, and facilitate fair comparison with other approaches.

3) Simulated evaluation can have a good assessment of goal-oriented dialog systems and show a moderate correlation with human evaluation, but it remarkably overestimates the system performance in human interactions. Thus, there is a need to devise better user simulators that resemble humans more closely. A simulator should be able to generate a natural and diverse response, and may change goals in complex dialog, etc. In addition, the simulator itself may make mistakes which derive the wrong estimation of the performance. However even with human evaluation a dialog system needs to deal with more complicated and uncertain situations. Therefore, it is vital to enhance the robustness of the dialog systems. Despite the discrepancy between simulators and human users, simulated evaluation is still a valid alternative to the costly human evaluation especially in the early stage of development.

Acknowledgements

This work was jointly supported by the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096), and the National Key R&D Program of China (Grant No. 2018YFC0830200). We would like to thank anonymous reviewers for their valuable suggestions, and Sungjin Lee for helpful discussions.

References

Masahiro Araki and Shuji Doshita. 1996. Automatic evaluation environment for spoken dialogue systems. In *Workshop on Dialogue Processing in Spoken Language Systems*, pages 183–194. Springer.

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural nlg from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, its gpt-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709.

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.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmad, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–495.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 80–87. IEEE.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019a. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019b. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of the 17th Annual Conference of the International Speech Communication Association*, pages 715–719.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.

Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka, and Toyomi Meguro. 2010. Issues in predicting user satisfaction transitions in dialogues: individual differences, evaluation criteria, and prediction models. In *Proceedings of the Second international conference on Spoken dialogue systems for ambient environments*, pages 48–60.

Seokhwan Kim, Michel Galley, R. Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, Minlie Huang, Luis Lastras, Jonathan K. Kummerfeld, Walter S. Lasecki, Chiori Hori, Anoop Cherian, Tim K. Marks, Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, and Raghav Gupta. 2019. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019a. 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.

Sungjin Lee and Amanda Stent. 2016. Task lineages: Dialog state tracking for flexible interaction. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 11–21.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019b. Convlab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 64–69.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. 2020. Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1468–1478.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177.

Tim Paek. 2001. Empirical methods for evaluating dialog systems. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Alexandros Papangelis, Mahdi Namazifar, Chandra Khatri, Yi-Chia Wang, Piero Molino, and Gokhan Tur. 2020. Plato dialogue system: A flexible conversational ai research platform. *arXiv preprint arXiv:2001.06463*.

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.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240.

Shiva Pentyala, Mengwen Liu, and Markus Dreyer. 2019. Multi-task networks with universe, group, and task feature learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 820–830.

Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019. Entity-consistent end-to-end task-oriented dialogue system with kb retriever. 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 133–142.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437.

Liliang Ren, Jianmo Ni, and Julian McAuley. 2019. Scalable and accurate dialogue state tracking via hierarchical sequence generation. 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 1876–1885.

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.

Alexander Schmitt and Stefan Ultes. 2015. Interaction quality: assessing the quality of ongoing spoken dialog interaction by expertsand how it relates to user satisfaction. *Speech Communication*, 74:12–36.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Pei-Hao Su, Milica Gasic, Nikola Mrkšić, Lina M Rojas Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2431–2441.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. In *Proceedings of the 58th Annual Meeting of the 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.

Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*, pages 73–78.

Stefan Ultes, Alexander Schmitt, and Wolfgang Minker. 2013. On quality ratings for spoken dialogue systems–experts vs. users. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 569–578.

Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. 1997. Paradise: a framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 271–280.

Tsung-Hsien Wen, Milica Gasic, 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.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, 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.

Jason D Williams, Kavosh Asadi Atui, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 665–677.

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.

Qizhe Xie, Kai Sun, Su Zhu, Lu Chen, and Kai Yu. 2015. Recurrent polynomial network for dialogue state tracking with mismatched semantic parsers. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 295–304.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020a. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

Zheng Zhang, Ryuichi Takanobu, Minlie Huang, and Xiaoyan Zhu. 2020b. Recent advances and challenges in task-oriented dialog system. *arXiv preprint arXiv:2003.07490*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. 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 1208–1218.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467.

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*.

Hotel			Restaurant	
Info	Reqt	Book	Info	Book

parking=yes
pricerange=moderate
	postcode
	address
		stay=2
		day=sunday
			food=british
			pricerange=moderate
			area=centre
				day=sunday
				time=16:30
				people=8

SYSTEM-1	SYSTEM-6
U: I would prefer it in the centre area . Actually I need a moderately priced restaurant . Are there any fitting that description ? This restaurant should serve british food too .	U: I would prefer it in the centre area . Actually I need a moderately priced restaurant . Are there any fitting that description ? This restaurant should serve british food too .
D: *restaurant-food:british, restaurant-pricerange:moderate, restaurant-area:centre*	D: *restaurant-food:british, restaurant-pricerange:moderate, restaurant-area:centre*
S: I have 4 options for you ! How about the copper kettle ? It is located in the centre .	S: I have 4 options for you ! It is in the centre area . That is a british restaurant . the copper kettle is just your thing . It is in the centre area .
U: I need a table for 8 people . Can you try to book it at 16:30 . We will be there sunday night .	U: It will be for 8 people . Can you try for 16:30 ? I am also looking for a restaurant to book for sunday.
D: *restaurant-people:8, restaurant-day:sunday, restaurant-time:16:30*	D: *restaurant-people:8, restaurant-day:sunday, restaurant-time:16:30*
S: Your reference number is 00000013 .	S: All set . Your reference number is 00000013 .
U: I would like to keep it in the moderate range , please . Oh , i need parking.	U: Hello , I would like to find a hotel that includes free parking . I also need a moderate place to stay .
D: *hotel-parking:yes, hotel-pricerange:moderate*	D: *hotel-parking:yes*
S: I have 14 different options for you ! How about hamilton lodge ? Fits your request perfectly . It has free wifi .	S: Reference number is : 00000013 .
U: That sounds fine , I just need the postcode though .	U: What is the postcode ?
D:	D:
S: The postal code for that hotel is cb41da .	S: Thank you for using our services .
U: I need a room starting on sunday . There is going to be 8 of us. How about only 2 nights .	U: That sounds fine , I just need the postcode though .
D: *hotel-day:sunday, hotel-people:8, hotel-stay:2*	D:
S: Here is the booking information : Booking was successful . Reference number is : 00000001.	S: Glad to have been of help . Thank you for using the service . Enjoy the rest of your day !
U: Thank you very much , goodbye .	U: I just need the postcode .
D:	D:
S: You are very welcome. Goodbye.	S: Ok , thank you . Have a good day .
Success	**Failure:** System fails to track the value of price range in the dialog state and does not answer the postcode.

Table 6: A sample dialog session comparison between *SYSTEM-1* and *SYSTEM-6* in simulated evaluation. For simplicity, we only present the updated slots and values at each turn, instead of the entire dialog state (lines start with D).

Train			**Attraction**	
Info	Reqt	Book	Info	Reqt

destination=peterborough
day=saturday
departure=cambridge
leaveAt=13:00
	duration
	trainID
		people=7	area=centre	phone
				postcode

SYSTEM-1	**SYSTEM-6**
U: I am looking for a train that leaves at 13:00	U: Hi! I'm looking for a train departing from Cambridge after 13:00
S: What day would you like ? Where would you like your train to take you ? Where did you want to depart from ?	S: What day did you have in mind ? Where are you heading to ?
U: I would like to leave on Saturday and depart from Cambridge	U: Saturday and heading to peterborough
S: What is your destination ?	S: There is a train arriving at 13:00 would you like me to book tickets for that one ? I can book you on saturday. Would you like me to book a train to peterborough for you ? Would you like me to book a train from cambridge for you ?
U: I need to go to Peterborough	U: Yes, please book for 7 people
S: We can book you for the train leaving at 13:00 . I can book you on Saturday. Would you like me to book a train to peterborough for you ? Would you like me to book a train from cambridge for you ?	S: All set . Your reference number is 00000066 .
U: Yes. I need the train ID and booking for 7 people	U: Thanks! I'm looking forward to seeing local tourist attractions
S: Reference number is : 00000054 .	S: Thank you for using our services
U: I want to see the tourist attractions so it should be near the Centre	U: Do you know of any attractions in the centre?
S: There are 44 . Would you like me to recommend one for you ? Would you like vue cinema ?	S: We have 44 options , can i recommend for you ? I would suggest sheep's green and lammas land park fen causeway .
U: Yes. I also want the phone number and postcode for the cinema	U: Can I get the phone number for Sheep's Green?
S: The phone number is 08712240240 and postcode is cb11ps . They are located within the centre .	S: What time would you like to leave ?
U: Do you have me booked for a party of 7 people?	U: Around noon
S: Here is the booking information : Booking was successful . Reference number is : 00000054.	S: Have a good day .
U: Are there 7 seats booked?	U: Can you give me the phone number of the tourist attraction?
S: Reference number is : 00000054 .	S: Thank you for contacting the help desk . Have a great day .
U: Thank You for your help and Goodbye!	U: Can you give me a list of attractions?
S: You are welcome . Have a good day !	S: Have a good day .
Success	**Failure:** System does not answer the phone number of the requested tourist attraction.

Table 7: A sample dialog session comparison between *SYSTEM-1* and *SYSTEM-6* in human evaluation.