Simulating Inconsistencies in Task-oriented Dialog

Anonymous ACL submission

Abstract

Most existing dialog models are trained on static dialog datasets or in an interactive way with user simulators, and evaluated in the same way. Such methods mostly make an ideal hypothesis that the user behaves consistently to the goal. Nevertheless, inconsistent behaviors are often observed from real users due to unpredictable mind changes or language understanding errors. In this paper, we give a systematic investigation of the inconsistent problem in real-world dialog systems and introduce three kinds of inconsistencies, namely *Goal Change*, *Action Disloyalty* and *Understanding Deviation*. We propose a user model to simulate those three kinds of inconsistencies, which can be used to examine the model robustness. The simulation model is further utilized to support Reinforcement Learning and inconsistent data augmentation, which boosts the performance of pipeline and end-to-end dialog models under inconsistent situation.

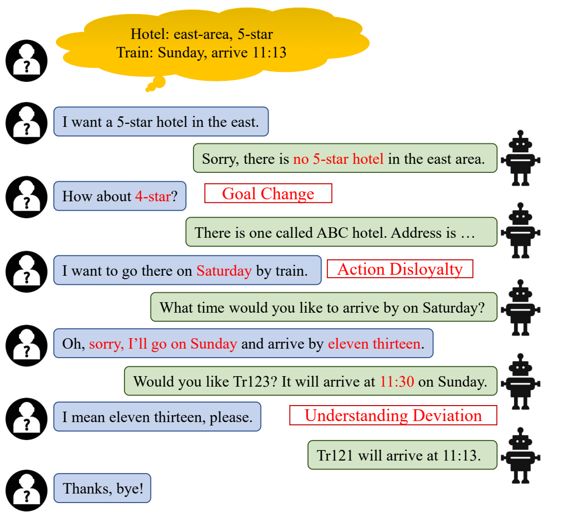


Figure 1: Inconsistencies when talking with real users. Real human may change their mind or make wrong actions. There are also understanding deviation problems such as ASR error.

1 Introduction

Recently, there has been a growing interest in task-oriented dialog system in both academic and industrial circles (Chen et al., 2017; Zhang et al., 2020b). Besides modularized tasks such as language understanding (Zheng et al., 2020), dialog state tracking (Henderson et al., 2014; Wu et al., 2019a) and policy learning (Takanobu et al., 2019, 2020a), the evaluation of dialog system has a pivotal role in improving system performance and robustness.

Existing evaluation methods for task-oriented dialogs can be categorized into three paradigms, including static evaluation based on collected datasets, interactive evaluation with human and user simulators. Evaluation with static datasets cannot handle previously unseen cases and does not take the diversity and interactiveness of dialogs into account (Lei et al., 2018; Wu et al., 2019a). Human evaluation can give more reliable judgments, but is rather costly and not scalable with large numbers of interactions required (Su et al., 2016b; Lipton et al., 2018). Automatic interactive evaluation with user simulators could both save cost and ensure diversity, but the discrepancy between simulated and real users still remains a short-coming (Peng et al., 2018; Wu et al., 2019c).

Although each evaluation method has its own advantage and disadvantage, they all depend on an ideal hypothesis that the users behave consistently in the dialog. Under such hypothesis, the user goal remain unchanged in existing interactive evaluation methods, and the users (real or simulated) are required to give responses which is consistent to the constant goal (Peng et al., 2018; Takanobu et al., 2019). However, the real world users do not necessarily act in this way. As shown in Figure 1, users could adaptively adjust their goals according to their needs and information obtained in the dialog context. Inevitably, the users will sometimes make mistakes and give actions which are irrelevant to their goals and the current topic. The systems may also make mistakes in understanding users' utterances, leading to irrelevant user intents. Therefore, it is important to take the user characteristics into consideration when evaluating a system, since different types of user behaviors can be extremely different and can affect model performance. Nevertheless, existing studies based on dialog-act level interactions neglect this issue (Peng et al., 2018; Takanobu et al., 2019).

To better understand the interactive robustness in real world dialogs, we systematically summarize the ideal consistency hypothesis into three types: *Goal Consistency*, *Action Loyalty* and *Understanding Correctness*, which are breakable in real world interactions with humans. We need to explore how the performance of dialog system changes when those hypotheses are broken to give a more systematic evaluation to the dialog models. We therefore simulates three types of inconsistent phenomenons: *Goal Change*, *Action Disloyalty* and *Understanding Deviation*, which are targeted to the above hypotheses. We integrate a user simulation model with the ability to induce inconsistent interactions, and use it to interact with dialog systems to simulate the environment with inconsistent phenomenons.

We conduct experiment on the MultiWOZ dataset (Budzianowski et al., 2018), which is a widely used large-scale multi-domain task-oriented dialog benchmark, to investigate the robustness of different system models when dealing with the inconsistent problems. An in-depth analysis is given on how each kind of inconsistencies influences the system performance. By building a user model with inconsistent simulation ability, we can realize a more realistic training environment for the interactive reinforcement learning tasks to improve the robustness of pipeline dialog system. Meanwhile, we collected an augmented dialog dataset using the inconsistency simulator, which can be used to boost the performance of end-to-end models.

The contribution of this study can be summarized as follows:

- We give a systematic definition of three types of inconsistencies in real-world dialog.
- In allusion to these inconsistencies, we build a user simulator with the ability to induce inconsistent interactions to test the model robustness under such situations.
- We further utilize the user model to provide an RL environment and a data augmentation toolkit to improve the model robustness of pipeline and end-to-end dialog models under inconsistent situations.

2 Related Work

2.1 Dialog System

Existing task-oriented dialog systems mainly fall into two categories: pipeline systems and end-to-end systems. Pipeline agents are constructed with several components including Nature Language Understanding (NLU) (Hakkani-Tür et al., 2016; Devlin et al., 2019), Dialog State Tracking (DST) (Wu et al., 2019b; Heck et al., 2020), Dialog Policy (Shah et al., 2016; Schulman et al., 2017), and Nature Language Generation (NLG) (Wen et al., 2015; Peng et al., 2020b). End-to-end agents (Zhang et al., 2020a) use a single model to generate textual response directly. Recently, pre-trained language models for end-to-end dialog modeling (Peng et al., 2020a; Hosseini-Asl et al., 2020) are developed and demonstrate favorable performance.

2.2 Dialog Evaluation

Evaluation has always been an important topic of task-oriented dialog. As evaluating with static data does not take the diversity and interactiveness of dialog into account, interactive evaluation would be a better approach. Evaluating by communicating with human (Ultes et al., 2013; Su et al., 2016a; Schmitt and Ultes, 2015) can get reliable results but is extremely costly. Automatic evaluation by interacting with user simulator (Araki and Doshita, 1996; Eckert et al., 1997; Schatzmann et al., 2007; Asri et al., 2016) is able to save both time and resource but the gap between simulators and real users still remains a shortcoming. In-depth analysis (Takanobu et al., 2020b) and evaluation platforms such as pydial (Ultes et al., 2017) and Convlab (Lee et al., 2019; Zhu et al., 2020) have been presented to evaluate dialog evaluation systemically.

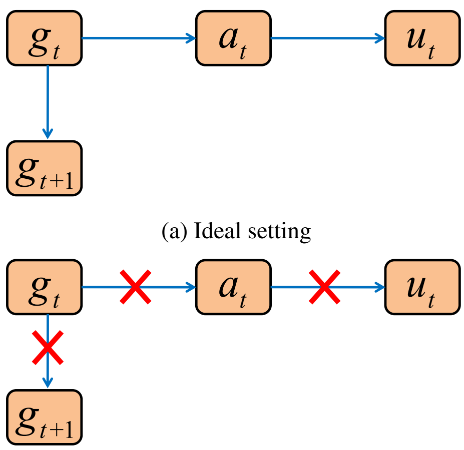
2.3 Dialog Robustness

Recently, researchers have shown an increased interest in the robustness of task-oriented system is attracting more and more research attention. Ganhotra et al. (2020) introduce naturalistic variation and investigate the impact of it on dialog systems. Irregular human behaviors like user goal change (Ma, 2013; Ma and Fosler-Lussier, 2014) are also important issues. For textual noises, Liu et al.

(2021) systemically studies the robustness of NLU models. Great efforts have also been made to develop training methods to improve the robustness (Su et al., 2018; Fazel-Zarandi et al., 2017; Pereira et al., 2021) of dialog models. However, very little research has been carried out on how the inconsistent interactions affect system robustness.

3 Inconsistencies

In this paper, we define and implement three types of inconsistencies in task-oriented dialog: Goal Change, Action Disloyalty and Understanding Deviation. Fig 2 illustrates the differences between ideal setting and inconsistent setting. In ideal evaluation setting, user goal is consistent and it determines user actions while system can understand user actions perfectly. In other words, $g_{t+1} = g_t$, $a_t \in g_t$ and $u_t = a_t$ hold for any dialog turn t. However, in practical conversations, those consistencies are often broken. We present an inconsistent evaluation environment with $g_{t+1} \neq g_t$, $a_t \notin g_t$ and $u_t \neq a_t$ to test models' robustness when facing such challenge.



(a) Ideal setting

(b) Inconsistent setting

Figure 2: Difference between ideal setting and inconsistent setting. g, a and u stand for user goal, user action and system understanding while t denotes the number of dialog turn.

3.1 Goal Change

According to user's motivation, we introduce two types of goal change: passive and active goal change.

Passive Goal Change Sometimes, user query could not be satisfied according to the knowledge base and system will return a negative response. For example, the user wants to book a 5-star hotel in a certain area where such hotel does not exist. In these cases, after knowing that their constraints are unreasonable, users have to change their goals. We simulate such passive goal change by first sampling an unsolvable initial goal g_{init} and change it to a new solvable goal $g_{init} \rightarrow g_{new}$ when get the negative feedback from system.

Active Goal Change Real users have their free will and they have the possibility to change their mind actively at any time in the real dialog. To simulate this behaviour, goal change $g_{init} \rightarrow g_{new}$ is allowed to occur at any turn t regardless of whether the system gives a negative response. In order to ensure the naturalness and completion of the dialog, only one constraint of the goal will be changed and the new goal g_{new} is guaranteed to be solvable.

3.2 Action Disloyalty

There are mainly two kinds of disloyal actions. One is relevant to user goal but is wrong while the other is totally not relevant to the goal.

Wrong Action Sometimes, users may make some mistakes on some information which is relevant to their goal and current topic. For instance, user may may provide an incorrect departure date when booking a ticket. We simulate such user behaviour by perturbing an action $a_t = a_{wrong} \notin g_t$ at a random sampled turn. Then the mistake will be corrected at a following turn $a_{t+k} = a_{correct} \in g_t$.

Irrelevant Action As real users are not necessary to follow the pre-defined dialog schema strictly, they may talk about something irrelevant to the goal. We simulate this behaviour by randomly inserting utterances with irrelevant actions $a_t = a_{irr} \notin g_t$. Inserted action a_{irr} is randomly sampled from other domains which are not in user goal.

3.3 Understanding Deviation

According to the source of mistake, we defined two types of understanding deviation: natural mistake and extra perturbation.

Natural Mistake As we know, no existing NLU model could guarantee a 100% performance even without any extra perturbation. As a result, comparing to ideal setting where user and system communicate at action-level, there always exist understanding mistakes when system need to understand textual utterances. We call those NLU mistakes natural mistakes. We monitor those understanding deviation during the dialog and study how they affect the whole dialog.

Extra Perturbation As there is more text-level noise in real dialog than in collected datasets, it is necessary to study system's robustness to those extra perturbation. We introduce four perturbation methods from a text augmentation toolkit LAUG[1]: word perturbation, text paraphrasing, speech recognition and speech disfluency. Those perturbation methods are randomly applied on some utterances and we study how robust system models are to these extra understanding mistakes.

4 Experimental Setup

4.1 Dataset

# Training Dialogs	8,438
# Validation Dialogs	1,000
# Test Dialogs	1,000
# Domains	7
# Entities in Database	3116
Avg. # Turns per Dialog	13.7

Table 1: Statistics of MultiWOZ 2.1.

We instantiate our inconsistent simulating on MultiWOZ corpus. MultiWOZ is a widely used multi-domain task-oriented dialog dataset. We use the 2.1 version of MultiWOZ[2] and Table 1 shows its statistics. There are seven domains including train, taxi, restaurant, hotel, attraction, hospital and police with over three thousand entities in database. We choose it as the representative dataset for our experiments due to its challenging multi-domain setting and rich knowledge base.

4.2 Base Models

Type	Name
Pipeline	MLE
Pipeline	PG(Shah et al., 2016)
Pipeline	PPO(Schulman et al., 2017)
E2E	DAMD(Zhang et al., 2020a)
E2E	SOLIST(Peng et al., 2020a)

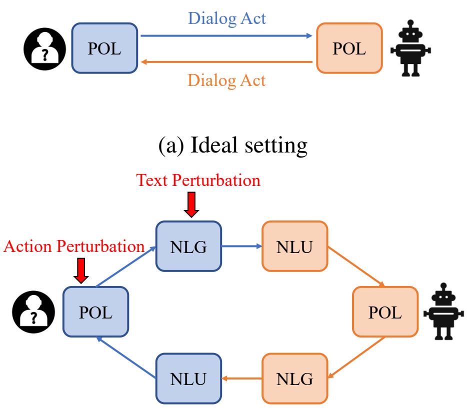
Table 2: List of the base dialog models in our experiments.

[1] https://github.com/thu-coai/LAUG

[2] https://github.com/budzianowski/multiwoz/blob/master/data/MultiWOZ_2.1.zip

As Table 2 shows, we conduct our experiments based on five dialog system models which can be divided into two main categories: pipeline models and end-to-end models. We evaluate and compare their robustness under inconsistent setting. As such robustness is mainly on policy-level, we name the pipeline systems using the name of policy method while keeping other components the same. We adopt BERT-NLU model for both user and system sides. Dialog states are updated by rule according to the output of NLU. As linguistic diversity is not the main target of our tests, user and system NLG are set as template-based to reduce unstable facts. As for end-to-end agents, system NLU and NLG are not required while user side follows the settings above.

The implements for BERT-NLU, rule-DST and template-NLG are from Convlab2. We adopt a T5-base model for SOLOIST while the other four models are also Convlab2 version.



(a) Ideal setting

(b) Inconsistent setting

Figure 3: Evaluation frameworks of pipeline models. Blue parts stand for user side while orange ones are for system.

4.3 Evaluation Setup

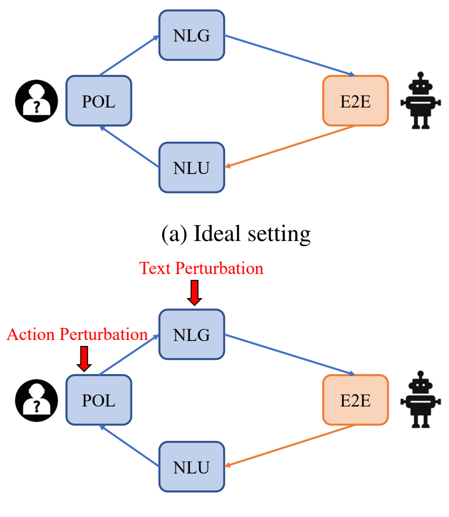
Our evaluation framework is mainly based on Convlab2[3]. System models are automatically evaluated by interacting with a user simulator. User goals are sampled by a goal model which simulating the data distribution and user actions are generated by a agenda-based user policy according to the user goal. We report the complete rate, success rate, and inform F1 in our experiments. Among these metrics, we pay the most attention to success rate because we want to evaluate whether the dialog can

[3] https://github.com/thu-coai/ConvLab-2

System	Consistent			Inconsistent		
	Comp	Succ	Inform	Comp	Succ	Inform
MLE-Pipeline	53.6	52.1	66.4	33.8(-36.9%)	30.1(-42.2%)	50.4(-24.1%)
PG-Pipeline	53.7	52.6	65.8	34.1(-36.4%)	31.0(-41.0%)	49.5(-24.7%)
PPO-Pipeline	71.1	**69.5**	**68.4**	55.5(-21.9%)	**44.1**(-36.5%)	**56.1**(-18.0%)
DAMD-E2E	37.9	33.4	54.0	33.0(-12.9%)	22.2(-33.5%)	45.5(-15.7%)
SOLOIST-E2E	**74.4**	36.8	62.8	**60.1**(-19.2%)	22.7(-38.5%)	45.9(-26.9%)

Table 3: Main results of robustness evaluation. We present the complete rate, success rate and inform F1 performance. The percentage in brackets represents the relative performance decline rate. The bold results represent the highest performance, and the green ones stand for the most robust model.

finally succeed under the interference of inconsistent perturbations throughout the dialog process.



(a) Ideal setting

(b) Inconsistent setting

Figure 4: Evaluation frameworks of end-to-end models.

For pipeline agents, the ideal setting is directly communicating with user policy at action-level as Fig 3 shows where there is no action perturbation or understanding deviation. In inconsistent setting, NLG and NLU are adopted to introduce natural understanding mistakes. Extra text perturbations are four methods (word perturbation, text paraphrasing,speech recognition and speech disfluency) from LAUG toolkit, with a possibility of 10% for each method per utterance. Action-level perturbations including goal change and action disloyalty are applied on user agenda policy. 50% dialogs contain passive goal changes while the other 50% contain active changes. Each user turn has a 20% probability of becoming an inserted utterance with irrelevant action. Although wrong action and active goal change have different user motivation, their implements are the same (perturb one action and then correct it) because user goal is invisible to system agent. So we do not present experimental results about wrong action because it is duplicated.

Fig 4 demonstrate the evaluation frameworks of end-to-end agents. Because end-to-end agents can only input and output text, user NLG and NLU are necessary. Thus, the ideal setting of end-to-end models can not avoid natural understanding mistakes. For inconsistent setting, we use the same text-level and action-level perturbations to pipeline settings.

5 Robustness Evaluation

5.1 Main Results

We conducted experiments in both the ideal and inconsistent setting to study the robustness of the base models. Table 3 is the is the result of the comparison of the two settings. Each result is an average of 5 runs with 1000 dilogs at a time. Significant performance decline can be observed on all models. In general, the pipeline models have high performance in the ideal setting while the end-to-end models are more robust in the inconsistent setting. But please note that due to the difference the evaluation frameworks of pipeline and end-to-end models, their results could not be compared directly in ideal setting. Among the three pipeline models, PPO has both the best and the most robust performance while MLE and PG have similar performance on all metrics and settings. Although the result of DAMD is the lowest, its relative performance drop under inconsistent setting is the least. SOLOIST has the highest complete rate and a better success rate than DAMD but its robustness is relatively poor. Among the three metrics, the success rate of the model decreased the most which indicate that addressing the inconsistent issues in dialog process and finally successfully completing them is challenging.

5.2 Ablation Studies

In order to investigate how each type of inconsistency influences the performance of models, we conduct in-depth ablation studies. We adopt goal change, action disloyalty and understanding deviation separately and analyze their ablation results.

model	Comp	Succ	Inform
MLE	40.6 (-24.2%)	37.7 (-27.6%)	56.7 (-14.6%)
PG	40.4 (-24.7%)	39.5 (-24.9%)	56.3 (-14.4%)
PPO	61.4 (-13.6%)	**58.3** (-16.1%)	57.1 (-16.5%)
DAMD	32.1 (-15.3%)	26.8 (-19.7%)	52.9 (-2.0%)
SOLOIST	**71.3** (-4.2%)	30.0 (-18.5%)	**60.1** (-4.3%)

(a) Passive goal change only.

model	Comp	Succ	Inform
MLE	46.8 (-12.7%)	45.3 (-13.1%)	58.7 (-11.6%)
PG	47.8 (-11.0%)	47.5 (-9.7%)	60.3 (-11.8%)
PPO	**65.8** (-14.7%)	**61.9** (-10.9%)	**63.5** (-7.2%)
DAMD	32.9 (-13.2%)	27.4 (-18.0%)	50.0 (-7.4%)
SOLOIST	61.6 (-17.2%)	32.1 (-12.8%)	60.3 (-4.0%)

(b) Active goal change only.

Table 4: Results when include goal change only. Results of passive and active goal change are represented separately. The possibility of both type are set to 50% which is same to the settings of main experiment.

Goal Change Table 4 shows the results when include goal change only. On the whole, larger performance decline occurs on passive goal change. Though active goal is more unpredictable, it only changes one slot constraint per time due to our setting. While passive goal change may change multiple constraint in order to get a new solvable goal which could be the reason why it is more challenging. According to the success rate results, end-to-end models are relatively more robust to passive goal change while pipeline agents are more robust to active changes. PPO is the most robust model against passive changes among the pipeline models while PG is very robust to active changes. Both type of goal change have little impact on Inform F1 metrics to the two end-to-end models. SOLOIST only loses very little complete rate when facing passive goal change and it is quite robust to active change in the perspective of success rate.

Action Disloyalty As Sec 4.3 states, we only present the results of irrelevant action. PPO has the highest performance on all three metrics while

model	Comp	Succ	Inform
MLE	40.6 (-24.2%)	37.7 (-7.6%)	56.7 (-14.6%)
PG	52.0 (-3.2%)	51.5 (-2.1%)	62.7 (-4.7%)
PPO	**70.3** (-11.2%)	**66.9** (-3.7%)	**65.3** (-4.5%)
DAMD	37.5 (-1.1%)	33.0 (-1.2%)	51.6 (-4.4%)
SOLOIST	50.2 (-32.5%)	26.2 (-28.8%)	45.5 (-27.5%)

Table 5: Irrelevant Action only.

DAMD is the most robust to irrelevant perturbation. Apart from DAMD, PG and PPO also have only small performance decline. However, SOLOIST is particularly not robust to irrelevant actions which could be the biggest weak point of SOLOIST.

model	Comp	Succ	Inform
MLE	52.5 (-2.1%)	48.9 (-6.1%)	65.2 (-1.8%)
PG	48.7 (-9.3%)	45.6 (-13.3%)	60.7 (-7.8%)
PPO	**71.1** (-0.0%)	**62.4** (-10.2%)	**66.8** (-2.3%)

(a) Natural Mistake change only.

model	Comp	Succ	Inform
MLE	51.9 (-3.2%)	47.7 (-8.4%)	64.3 (-3.2%)
PG	47.9 (-10.8%)	44.7 (-15.0%)	60.5 (-8.1%)
PPO	**69.4** (-2.4%)	**61.7** (-11.2%)	**66.0** (-3.5%)

(b) Nature Mistake and Extra Perturbation.

Table 6: Ablation results of understanding deviation.

Understanding Deviation Table 6 shows the results of pipeline agents when only understanding deviation exists. As the understanding ability of end-to-end models can not be ablated, we only study pipeline models here. All three models are quite robust to understanding deviation. Relatively, MLE is the most robust one while PG is most sensitive to understanding mistakes.

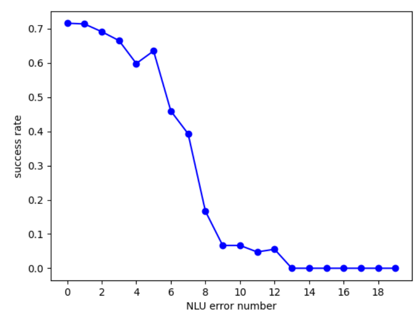


Figure 5: Relationship between the number of NLU errors and the success rate of PPO.

In order to investigate why models are robust to

Speaker	Utterance
User	Well, I'm looking for a hotel. I need a place to stay in the uh, the east please. It should have free wifi.
System	What area of town would you like to be in? Is there a price range you prefer? I have 6 different options for you!
User	In the east preferably. I don't care about the price of the hotel.
System	I'd recommend Express by Holiday Inn Cambridge. Would you like some information on it? Their address in our system is listed as 15 - 17 Norman way, Coldhams business park.
User	I'd like their phone number if you have it available, please.
System	Here is the hotel phone number, 01223866800.
User	I need find information about restaurant, can you help that with? Are there any Thai restaurants? I'd like a moderate priced one.
System	I don't have anything meeting that criteria. Can I look for something else?
User	I want to find a expensive priced restaurant.
System	There's a place called Sala Thong. It is in the west area.
User	I want to leave by 05:00.
System	Is there anything else I can help you with?
User	May I also get the address for the restaurant?
System	Their address is 35 Newnham Road Newnham. 01223323178 is the restaurant phone number.
User	Actually , I'm all set. Thank you! Bye!

Table 7: Dialog cases of PPO agent under inconsistent setting. As the original dialogs are too long, we only present part of them, and some utterances are abbreviated. The colored text are the inconsistent parts.

understanding mistakes, we make a statistics of the number of NLU errors which is shown in Fig 5. We found that the success rate is relatively stable when NLU error number is less than 5 while it drops rapidly when there are more than 5 errors. This result indicates that system policies has the ability to save the dialog from several NLU mistakes which is the reason why they are robust to understanding deviation even extra perturbations are added. For example, system agent can request for some information again when fail to understand it and user may reply to it in a way that can be successfully understood.

5.3 Case Study

Tab 7 present some dialog cases with typical inconsistencies. The red part in the first dialog is an understanding deviation case. System NLU fails to recognize the area information "east" at the first time. System agent ask for it in the following turn and get the answer again. In the green parts, user simulator changes its price constraint after knowing there's no such restaurant. The blue utterance is an inserted irrelevant action. System agent ignores it and they continue to talk about restaurant in the following turns.

6 Augmentation

6.1 Data Augmentation

For supervised methods, training with targeted additional data is a straightforward but effective way to improve robustness. We present an Inconsistency Enhanced version of training data: MultiWOZ-ICE by data augmentation. Step one, we run and record 15000 interactive conversations between inconsistent user simulator and a rule-based system policy from Convlab2 with high performance. Step two, we discard all failed dialogs and there are 11027 dialogs remaining. These dialogs are divided into 9:1 for training and validation. Note that text perturbations are not included in step one. Instead, we adopt LAUG to introduce text-level noise at the final step three because we want to get noisy text with correct dialog act labels. All the action-level and text-level perturbation of augmentation are using the same settings to the main experiment.

model	Comp	Succ	Inform
MLE	38.9(+25.8%)	**34.8**(+21.4%)	**52.7**(+12.5%)
DAMD	35.8(+57.1%)	23.9(+15.2%)	48.7(+37.6%)
SOLOIST	**69.4**(+65.0%)	24.3(+11.3%)	51.0(+32.5%)

Table 8: Results under inconsistent setting after training on MultiWOZ-ICE. Numbers in brackets are performance recover rates (performance_recover / performance_drop). Red stand for the biggest recover.

After we construct the augmented training data, we fine-tune the three supervised models on it. Note that for MLE agent, the NLU model is also enhanced by perturbed data augmented by LAUG toolkit. Table 8 shows the performance recover of MLE, DAMD and SOLOIST after training on the augmented data. In terms of success rate, MLE recovers the most because the policy model is directly tuned by augmented dialog act data. The two end-to-end models recover less on success rate but more on complete rate and Inform F1. As the success rate does not increase much and the training with the auto-generated data will affect the naturalness of the output utterance, data augmentation may not be a very suitable method to enhance the pre-trained end-to-end model like SOLOIST.

6.2 Environment Augmentation

For reinforcement learning based models, we train the policy interactively in an inconsistent environment to enhance their robustness to it. As PG and PPO are pipeline agents, their NLU models are also the LAUG-enhanced version as MLE. The results of them are shown in Tab 9. The recover rates of PG and PPO are similar to MLE, which indicates that pipeline agents could recover more success rate through augmentation. PPO has a higher recover rate of complete and success than PG.

model	Comp	Succ	Inform
PG	38.8 (+24.0%)	34.9 (+18.6%)	52.4 (+17.8%)
PPO	**60.3** (+30.8%)	**49.2** (+20.1%)	**63.0** (+16.3%)

Table 9: Results of inconsistent setting after reinforcement learning in augmented environment.

7 Conclusion and Discussion

In this paper, we investigate the effect of breaking the ideal consistencies in task-oriented dialog. We define and simulate three types of inconsistencies including *Goal Change*, *Action Disloyalty* and *Understanding Deviation* along with two sub-type for each of them. We conduct a robustness evaluation and the significant performance decline when models facing those inconsistent challenges indicates that it is dangerous to use clean and ideal data and settings for the training and evaluation of task-oriented dialog system. We also carry out an in-depth ablation study to investigate the model robustness to different types of inconsistencies. We find that pipeline agents and end-to-end models have different characteristics when under different inconsistencies. We augment the data and environment to fine-tune dialog models and experimental results demonstrate that these augmentation methods can improve the robustness of them. However, such improvement is quite limited, so more future efforts are supposed to made in order to develop more effective approaches for addressing the naturalistic dialog inconsistencies.

Acknowledgements

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.

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*.

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.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.

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.

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

Maryam Fazel-Zarandi, Shang-Wen Li, Jin Cao, Jared Casale, Peter Henderson, David Whitney, and Alborz Geramifard. 2017. Learning robust dialog policies in noisy environments. *arXiv preprint arXiv:1712.04034*.

Jatin Ganhotra, Robert C Moore, Sachindra Joshi, and Kahini Wadhawan. 2020. Effects of naturalistic variation in goal-oriented dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4013–4020.

Dilek Hakkani-Tür, Gokhan Tur, 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. *Interspeech 2016*, pages 715–719.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

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

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Xiang Li, Yaoqin Zhang, Zheng Zhang, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, et al. 2019. Convlab: Multi-domain end-to-end dialog system platform. *arXiv preprint arXiv:1904.08637*.

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, Melbourne, Australia. Association for Computational Linguistics.

Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. 2018. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021. Robustness testing of language understanding in task-oriented dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2467–2480.

Yi Ma. 2013. User goal change model for spoken dialog state tracking. In *Proceedings of the 2013 NAACL HLT Student Research Workshop*, pages 91–97.

Yi Ma and Eric Fosler-Lussier. 2014. A discriminative sequence model for dialog state tracking using user goal change detection. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 318–323. IEEE.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020a. Soloist: Few-shot task-oriented dialog with a single pretrained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Deep Dyna-Q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192, Melbourne, Australia. Association for Computational Linguistics.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020b. Few-shot natural language generation for task-oriented dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 172–182.

Lis Pereira, Xiaodong Liu, Hao Cheng, Hoifung Poon, Jianfeng Gao, and Ichiro Kobayashi. 2021. Targeted adversarial training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5385–5393.

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 experts—and how it relates to user satisfaction. *Speech Communication*, 74:12–36.

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

Pararth Shah, Dilek Hakkani-Tur, and Larry Heck. 2016. Interactive reinforcement learning for task-oriented dialogue management.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016a. On-line active reward learning for policy optimisation in spoken dialogue systems. *arXiv preprint arXiv:1605.07669*.

Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016b. 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, Berlin, Germany. Association for Computational Linguistics.

Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 2018. Discriminative deep dyna-q: Robust planning for dialogue policy learning. *arXiv preprint arXiv:1808.09442*.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020a. 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*, pages 625–638, Online. 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, Hong Kong, China. Association for Computational Linguistics.

Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020b. Is your goal-oriented dialog model performing really well? empirical analysis of system-wise evaluation. *arXiv preprint arXiv:2005.07362*.

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 ACL 2017, 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.

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

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019a. 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, Florence, Italy. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019b. 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.

Yuexin Wu, Xiujun Li, Jingjing Liu, Jianfeng Gao, and Yiming Yang. 2019c. Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7289–7296.

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 AAAI Conference on Artificial Intelligence*, volume 34, pages 9604–9611.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020b. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027.

Yinhe Zheng, Guanyi Chen, and Minlie Huang. 2020. Out-of-domain detection for natural language understanding in dialog systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1198–1209.

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. *arXiv preprint arXiv:2002.04793*.

A Example Appendix

This is an appendix.