UBAR: Towards Fully End-to-End Task-Oriented Dialog Systems with GPT-2

Yunyi Yang,Yunhao Li, Xiaojun Quan
Sun Yat-sen University
{yangyy37, likun36}@mail2.sysu.edu.cn, quanxj3@mail.sysu.edu.cn

Abstract

This paper presents our task-oriented dialog system UBAR [1] which models task-oriented dialogs on a dialog session level. Specifically, UBAR is acquired by fine-tuning the large pre-trained unidirectional language model GPT-2 on the sequence of the entire dialog session which is composed of user utterance, belief state, database result, system act, and system response of every dialog turn. Additionally, UBAR is evaluated in a more realistic setting, where its dialog context has access to user utterances and all content it generated such as belief states, system acts, and system responses. Experimental results on the MultiWOZ datasets show that UBAR achieves state-of-the-art performances in multiple settings, improving the combined score of response generation, policy optimization, and end-to-end modeling by 4.7, 3.5, and 9.4 points respectively. Thorough analyses demonstrate that the session-level training sequence formulation and the generated dialog context are essential for UBAR to operate as a fully end-to-end task-oriented dialog system in real life. We also examine the transfer ability of UBAR to new domains with limited data and provide visualization and a case study to illustrate the advantages of UBAR in modeling on a dialog session level.[2]

Introduction

Task-oriented dialog (TOD) systems aim to assist users with various tasks such as hotel reservations and ticket booking through natural language conversations. Recent years have seen a rapid growth of interest in developing data-driven approaches for this task from both the research community and industry (Zhang et al. 2020b). The presence of wide range of domains requires TOD systems to have better transfer ability while remaining practical in real conversations.

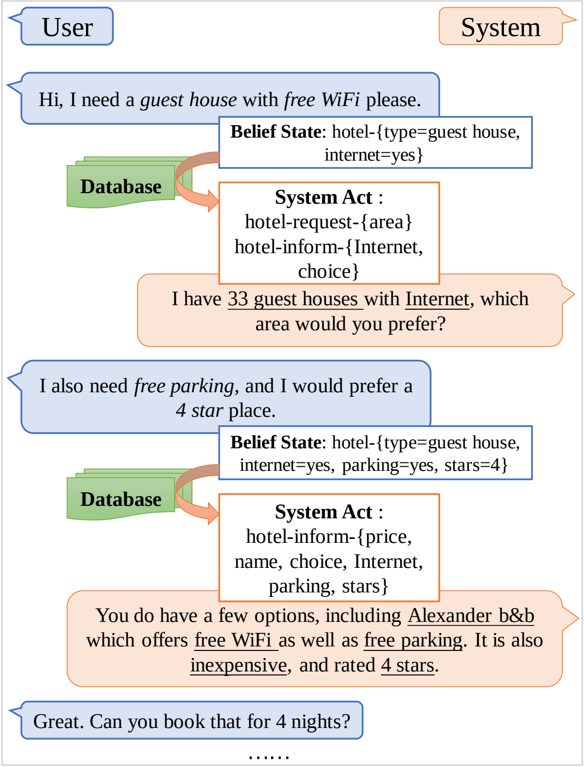

User
System
Hi, I need a *guest house* with *free WiFi* please.
Belief State: hotel-{type=guest house, internet=yes}
Database
System Act : hotel-request-{area} hotel-inform-{Internet, choice}
I have 33 guest houses with Internet, which area would you prefer?
I also need *free parking*, and I would prefer a *4 star* place.
Belief State: hotel-{type=guest house, internet=yes, parking=yes, stars=4}
Database
System Act : hotel-inform-{price, name, choice, Internet, parking, stars}
You do have a few options, including Alexander b&b which offers free WiFi as well as free parking. It is also inexpensive, and rated 4 stars.
Great. Can you book that for 4 nights?
……

Figure 1: An example of the first two-turn interactions between a user and a TOD system.

The functions of a task-oriented dialog system can be understood by introducing the traditional pipeline approach which consists of several consecutive modules. As shown in Figure 1, a dialog state tracker (DST) is equipped to estimate the belief state from the user utterance. The belief state can be used to query a task-related database (DB) for results such as the number of entities that match the user's goal. Then, a dialog policy learning module is applied to determine the next system act, followed by a natural language generation (NLG) module that maps the system act to a natural language response. These modules are often modeled and evaluated separately. The apparent drawback of the pipeline approach is that error propagation from the cascaded components can be detrimental to the subsequent subtasks (Liu and Lane 2018).

From a big picture perspective, the methodology for task-oriented dialog systems is gradually progressing from pipeline approaches to end-to-end modeling. Recently, some work attempts to incorporate the intermediate supervision,

[1] {**U**ser Utterance, **B**elief States, DB results, System **A**cts, System **R**esponse}

[2] Code and technical appendix available at https://github.com/TonyNemo/UBAR-MultiWOZ

i.e., the belief state and system act, and train systems in joint learning settings (Chen et al. 2019; Wang et al. 2020). They jointly generate system act and response, yet still using ground truth belief state. Some methods have come close to modeling TOD in an end-to-end manner, but they tend to use different decoders for each component. For example, Lei et al. (2018) use a seq2seq model to generate belief spans and responses. Zhang, Ou, and Yu (2020) propose multiple decoders to generate belief spans, act spans, and responses.

On the other hand, the large pre-trained language model GPT-2 (Radford et al. 2019) is shown to be capable of modeling the dialog pipeline in a unified way. Initially, Ham et al. (2020) propose to train a unified language model for task-oriented dialogs with a single sequence in format of dialog history (all previous user utterances and responses), user utterance, belief state, system act, response of the current turn, and evaluate for DST and policy optimization. SimpleTOD (Hosseini-Asl et al. 2020) and SOLOIST (Peng et al. 2020a) further generalize this idea to an end-to-end setting where the belief states are also generated instead of using ground truth values. They also incorporate database results into the training process. In spite of the promising results from leveraging pre-trained language models like GPT-2 for end-to-end TOD systems, these methods do not fully explore the process of training and evaluating towards a real-life task-oriented dialog setting. Specifically, these GPT-2-based TOD systems are trained and evaluated on a dialog turn level instead of the dialog session level, which has several limitations. First, the dialog history of these methods only consists of user utterances and system responses but leaves out the intermediate information such as belief states and system acts of the previous turns. These intermediate information could be a helpful reference for the generation of the current turn. Second, they use the ground truth responses from annotations in the dialog history, which makes the generation of a dialog turn independent of other turns in a dialog session. Third, the assumption of having access to the ground truth system responses is invalid in real conversations.

To address the aforementioned limitations and advance towards a fully end-to-end TOD system, we propose UBAR to model task-oriented dialogs on a dialog session level. We fine-tune GPT-2 on the sequence of the entire dialog session consisting of user utterance, belief state, database result, system act, and system response of every dialog turn. Such training data formation resembles the workflow of a real-life task-oriented dialog session, which allows UBAR to learn task completion and language generation over the course of a dialog session. UBAR is able to condition on the previous belief states and system acts in the dialog context, making the process of inference and planning easier for the current turn. Since in real conversations, a TOD system should be able to access the belief states it predicted and the system acts and responses it generated throughout the entire dialog session. We further propose to evaluate UBAR with the dialog context of generated content instead of the ground truth. This encourages UBAR to adaptively supplement and make amends in response to the current user utterance in order to stay consistent and coherent during the entire session, and ultimately contribute to the task completion goal.

We conduct experiments on the MultiWOZ datasets (Budzianowski et al. 2018; Eric et al. 2019) in multiple settings including response generation, policy optimization, end-to-end modeling and dialog state tracking, and compare UBAR with its GPT-2-based predecessors and other strong baselines. UBAR achieves state-of-the-art performances in all compared settings. We perform thorough analysis to show that the session-level training sequence formulation and all-generated dialog context are essential for UBAR to operate as a fully end-to-end TOD system in real life. We also examine the transfer ability of UBAR to new domains given limited data, and provide visualization and case study to illustrate the advantages of modeling task-oriented dialogs on a session level.

Related Work

Towards End-to-End Task-Oriented Dialog

With the emergence of large-scale multi-domain TOD datasets (Budzianowski et al. 2018; Shah et al. 2018; Peskov et al. 2019), the methodology for task-oriented dialog systems can be roughly seen to gradually progress from discriminative and modularized modeling to generative and end-to-end modeling over the recent years. Early methods for DST are commonly formulated as a classification task, where the dialog state representation maintains a distribution over all possible states for each slot (Henderson, Thomson, and Young 2013, 2014; Zhang et al. 2019b). To generalize to tracking unknown slot values and multi-domain settings, generative methods are proposed to extract slot values for DST (Zhong, Xiong, and Socher 2018; Xu and Hu 2018; Wu et al. 2019a). Similarly for dialog policy learning, system acts are originally encoded in vector representations such one-hot vectors and used for response generation (Chen et al. 2019; Zhao, Xie, and Eskenazi 2019; Wen et al. 2017). Then, they are jointly trained and generated with system responses (Wang et al. 2020; Zhang, Ou, and Yu 2020). For end-to-end modeling, Lei et al. (2018) propose a two-stage CopyNet (Gu et al. 2016) that generates belief spans and system response jointly via a single seq2seq architecture. Zhang, Ou, and Yu (2020) propose a domain-aware multi-decoder model that uses separate decoders to generate belief spans, act spans and responses. Recently, pre-trained language model like GPT-2 is also leveraged for end-to-end modeling in a unified way (Peng et al. 2020a; Hosseini-Asl et al. 2020). Besides, end-to-end TOD systems that directly operate on dialog history and interact with knowledge base without any intermediate supervision (Eric and Manning 2017; Madotto, Wu, and Fung 2018; Wu, Socher, and Xiong 2019) also receive growing attention, but are not within the scope of our discussion.

Pre-trained Language Models for Dialog Systems

Large pre-trained language models have shown superior performance on a wide range of NLP tasks (Peters et al. 2018; Radford et al. 2018; Devlin et al. 2019), and GPT-2 (Radford et al. 2019) is especially good at language generation tasks. Some work extends GPT-2 (Radford et al. 2019) to generate responses in chit-chat dialog (Zhang et al. 2020a; Wu

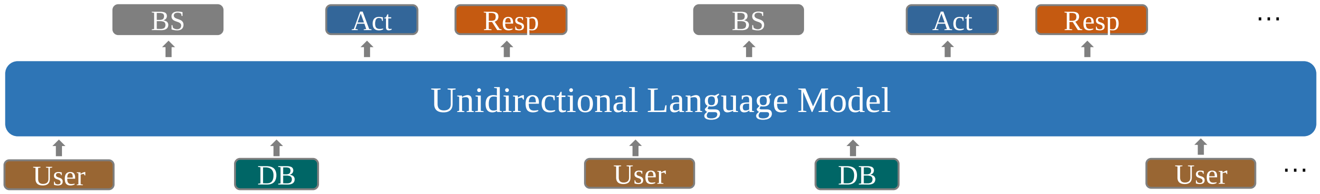


Figure 2: An overview of UBAR.

et al. 2020). In task-oriented dialog domain, Budzianowski and Vulić (2019) first point out the possibility of fine-tuning all necessary information in simple text on GPT-2 which inspires a line of improved and simplified design of task-oriented dialog systems. SC-GPT (Peng et al. 2020b) is a pre-trained model that converts ground-truth system acts into responses. Ham et al. (2020) fine-tune GPT-2 in a similar fashion for DST and policy optimization, but employ heuristic rules to handle different database query results. SimpleTOD (Hosseini-Asl et al. 2020) incorporates the database results into the training process and is evaluated for end-to-end modeling where belief state and system act are generated. SOLOIST (Peng et al. 2020a) follows a pre-train and fine-tune paradigm where it first undergoes pre-training on a large number of out-of-domain dialog turns, then fine-tune on the data of new domains. It does not require the annotation of system acts. This work follows its GPT-2-based predecessors and progresses for a fully end-to-end TOD system by operating in terms of a whole dialog session instead of a dialog turn during training and evaluating.

Method

In this section, we describe how UBAR models on a dialog session level and how we prepare the dialog data to be trained in sequence. Figure 2 is an overview of UBAR.

Modeling on a Dialog Session Level

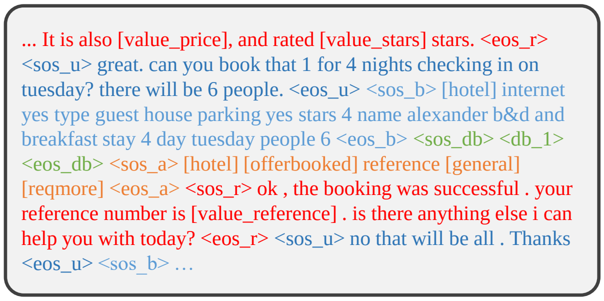
The workflow of a TOD system interacting with a user naturally produces a sequence as it reads user utterances, tracks dialog states and generates acts and responses over the turns of a dialog session.

Given a dialog session composed of multiple turns, we show how UBAR models the process of a task-oriented dialog session. In the first turn $t = 0$, the user inputs user utterance U_0, UBAR generates a belief state B_0 based on U_0. This belief state is used to query a database to retrieve the matched number of entities that satisfy the constraint imposed by the belief state, which is the database search result D_0. Conditioned on $\{U_0, B_0, D_0\}$, UBAR then generates system act A_0 and the delexicalized response R_0, completing the interaction of the first turn. As the dialog proceeds to turn t, UBAR generates B_t, A_t and R_t based on context of user utterances and all previous generated outputs $\{U_0, B_0, D_0, A_0, R_0, ..., U_{t-1}, B_{t-1}, D_{t-1}, A_{t-1}, R_{t-1}, U_t\}$, eventually completing the entire dialog session. Therefore, a single training sequence for a dialog session with T turns can be formulated as $\{U_0, B_0, D_0, A_0, R_0, ..., U_T, B_T, D_T, A_T, R_T\}$.

Note that UBAR is different from previous methods that are trained and evaluated in turn-level sequences where they are based on dialog history $\{U_0, R_0, ..., U_{t-1}, R_{t-1}, U_t\}$ to generate the response in turn t and their responses in the dialog history are ground truth. While UBAR incorporates the intermediate information B, D and A in the context.

Domain-Adaptive Pre-processing

We pre-process the dialog data into the form of simple text by delexicalizing the responses and converting intermediate dialog information to spans. All sequences are surrounded by special <*sos_?*> and <*eos_?*>[3] tokens indicating the beginning and end of a component. Figure 3 is an exemplar pre-processed training sequence.



... It is also [value_price], and rated [value_stars] stars. <eos_r> <sos_u> great. can you book that 1 for 4 nights checking in on tuesday? there will be 6 people. <eos_u> <sos_b> [hotel] internet yes type guest house parking yes stars 4 name alexander b&d and breakfast stay 4 day tuesday people 6 <eos_b> <sos_db> <db_1> <eos_db> <sos_a> [hotel] [offerbooked] reference [general] [reqmore] <eos_a> <sos_r> ok , the booking was successful . your reference number is [value_reference] . is there anything else i can help you with today? <eos_r> <sos_u> no that will be all . Thanks <eos_u> <sos_b> …

Figure 3: Example of different (colored) components in a dialog session concatenated into a single training sequence.

Delexicalization It is important to generate delexicalized responses for task-oriented generation, which allows the model to learn value-independent parameters (Wen et al. 2015). Delexicalization replaces specific slot values by their corresponding placeholders, which can be filled according to database search results afterwards. We adopt a domain-adaptive delexicalization scheme (Zhang, Ou, and Yu 2020) to decouple the domain and slot name of placeholders. For example, a hotel name in the generated response is just *<value_name>* instead of *<hotel-value_name>*.

Belief State and System Act Spans Belief states are originally represented in *domain-slot, value* pairs. We decouple domain and slot names to allow generalization cross different domains that share the same ontology. A schematic example of belief state span with two mentioned domains

[3] The *?* represents a component from {u, b, db, a, r}.

would be {*[domain1] slot value slot value [domain2] slot value*}. Each domain can follow several slot value pairs specifying user's goal.

For database search results, we use special tokens to indicate the number of matched entities under the constraints of the belief state of the current turn.

System acts are originally represented as *domain-act, slot* pairs which aim to inform or request information about the slot of a certain domain. We also decouple the domain and act for system act span: {*[domain] [inform] slot ... [request] slot ...*}. The decoupling of domains allows dialog ontology as well as expressions to be learned across relevant domains. The domains, acts and slot values are all bracketed as additional special tokens so that they can be learned specifically.

Architecture and Training Objective

GPT-2 (Radford et al. 2019) is a powerful pre-trained unidirectional language model. It is a large Transformer decoder (Vaswani et al. 2017) that is trained on large corpora of web text and can generate realistic and coherent natural language. By fine-tuning GPT-2 on session-level task-oriented dialog data, UBAR learns to ground generation with ontology knowledge and decision making ability.

The training objective for UBAR is the language modeling objective (Bengio et al. 2003), which maximizes the probability of next word prediction: $L = \sum_i logP(w_i|w_{<i})$. UBAR does not require additional training objectives such as next-utterance classification.

Experiments

Dataset and Evaluation Metrics

MultiWOZ 2.0 (Budzianowski et al. 2018) is a large-scale human-to-human multi-domain task-oriented dialog dataset consisting of 8438 dialogues spanning over seven domains (*attraction, hospital, police, hotel, restaurant, taxi, train*). It provides additional validation set and test set each of 1000 dialogues, excluding hospital and police. Each dialog session contains 1 to 3 domains and multiple domains might be mentioned in a single turn (more dataset details in appendix). MultiWOZ 2.1 (Eric et al. 2019) is an improved version of MultiWOZ 2.0 by fixing some noisy state annotations. We conduct experiments and analyses on the 2.0 version and also report results on the 2.1 version.

We follow the automatic evaluation metrics to evaluate task completion and response quality: **Inform** measures whether a system has provided a correct entity, **Success** measures whether it has answered all the requested information, and **BLEU** (Papineni et al. 2002) is used to measure the fluency of the generated responses (Budzianowski et al. 2018). A combined score: $(\textbf{Inform}+\textbf{Success})\times 0.5+\textbf{BLEU}$ is also reported as an overall quality measure suggested in Mehri, Srinivasan, and Eskenazi (2019). We also use the joint goal accuracy to evaluate dialog state tracking (DST).

Implementation Details

We implement UBAR with HuggingFace's Transformers (Wolf et al. 2019) and DistilGPT2 (Sanh et al. 2019), a distilled version of GPT-2. The model is trained on session-level sequences with a max sequence length of 1024. Sequences that exceed 1024 tokens are pre-truncated. We use the AdamW optimizer and standard greedy decoding method with temperature of 0.7. We select the best performing model on validation set through hyperparameters search of learning rate and batch size, then evaluate on test set to get the final results. We also report the performances of UBAR on validation set in technical appendix. Code and models are included in the supplement and will be released.

Baselines

We compare UBAR with SimpleTOD (Hosseini-Asl et al. 2020) and SOLOIST (Peng et al. 2020a), the GPT-2-based methods that are trained on turn-level data without generated belief state and system act in dialog history (Hosseini-Asl et al. 2020; Peng et al. 2020a), and other several competitive methods HDSA (Chen et al. 2019), SFN+RL (Mehri, Srinivasan, and Eskenazi 2019), ARDM (Wu et al. 2019b), and DAMD (Zhang, Ou, and Yu 2020). UBAR is evaluated and compared in three context-to-response settings: response generation based on ground truth belief state and system act, policy optimization to generate system act and response based on ground truth belief state, and end-to-end modeling to generate belief state, system act and response. Experiments with ground truth belief state use ground truth database search result. All content UBAR generated during a dialog session will remain in the dialog context for the generation the current turn.

Since the proposed UBAR can generate belief state throughout the entire dialog session, we compare the performance of UBAR on dialog state tracking with GPT-2-based model SimpleTOD (Hosseini-Asl et al. 2020) and other state-of-the-art methods such as TRADE (Wu et al. 2019a), DSTQA (Zhou and Small 2019), DST-Picklist (Zhang et al. 2019a), SST (Chen et al. 2020). As DST requires extracting slot values from non-delexicalized responses, we train a new DST-UBAR using non-delexicalized responses for DST evaluation.

Overall Results

Response Generation with Ground truth Belief State and System Act The first group in Table 1 shows the results of response generation based on the ground truth belief state and system act. UBAR applies the same domain-adaptive delexicalization and domain-aware belief, act spans as previous state-of-the-art DAMD, yet outperforms all compared methods in response generation in terms of inform rate, success rate and combined score, including DAMD. The BLEU score is slightly lower but inform rate and success rate are much higher than HDSA, which indicates UBAR is more grounded in task completion than language surface.

Policy Optimization with Ground truth Belief State In the setting of policy optimization, the context of UBAR consists of ground truth belief states and database results and generated act and responses. As shown in the second group of Table 1, UBAR achieves the best performance in terms of inform rate, success rate and combined score, improving the previous state-of-the-art SOLOIST by a large margin (3.5 points on the combined score). Note that SOLOIST is

Model	Belief State	System Act	Inform	Success	BLEU	Combined
HDSA	oracle	oracle	87.9	78.0	**30.4**	113.4
DAMD	oracle	oracle	95.4	87.2	27.3	118.5
SimpleTOD	oracle	oracle	92.3	85.8	18.67	107.7
UBAR (ours)	oracle	oracle	**96.9**	**92.2**	28.6	**123.2**
SFN+RL	oracle	generated	82.7	72.1	16.3	93.7
HDSA	oracle	generated	82.9	68.9	**23.6**	99.5
ARDM	oracle	-	87.4	72.8	20.6	100.7
DAMD	oracle	generated	89.2	77.9	18.6	102.2
SimpleTOD	oracle	generated	88.9	67.1	16.9	94.9
SOLOIST	oracle	-	89.6	79.3	18.0	102.5
UBAR (ours)	oracle	generated	**94.0**	**83.6**	17.2	**106.0**
SFN+RL	generated	generated	73.8	58.6	16.9	83.0
DAMD	generated	generated	76.3	60.4	16.6	85.0
SimpleTOD	generated	generated	84.4	70.1	15.0	92.3
SOLOIST	generated	-	85.5	72.9	16.5	95.7
UBAR (ours)	generated	generated	**95.4**	**80.7**	**17.0**	**105.1**

Table 1: Comparison of generation results on MultiWOZ 2.0. The oracle/generated denotes either using ground truth or generated intermediate information. The results are grouped according to how belief state and system act are modeled.

first pre-trained on large-scale task-oriented dialog data and then fine-tuned on MultiWOZ, and they did not leverage the system act. UBAR achieves higher inform and success rate than SOLOIST without additional pre-training data. These results show that UBAR can effectively learn dialog policy and response generation.

End-to-end Modeling The third group in Table 1 shows results in end-to-end modeling setting, where UBAR has to generate belief state, query database result with the generated belief state, and then generates act and response. UBAR achieves the state-of-the-art performance on all metrics and lifts almost 10 points on the combined score. Like SimpleTOD and SOLOIST, UBAR has a very simple architecture and is trained on sequences with language modeling objective. Unlike SimpleTOD and SOLOIST, UBAR uses all generated content in dialog context instead of ground truth responses. UBAR demonstrates incredible ability in modeling a complete task-oriented dialog session in an arguably fully end-to-end fashion, much closer to a task-oriented conversation in real life.

Results on MultiWOZ 2.1 We also report the performance of UBAR in the three settings on MultiWOZ 2.1 for future comparison. As shown in Table 2, the results are consistent with that on MultiWOZ 2.0.

Belief	Act	Inf.	Succ.	BLEU	Comb.
oracle	oracle	95.4	91.4	28.8	122.2
oracle	generated	92.7	81.0	16.7	103.6
generated	generated	95.7	81.8	16.5	105.7

Table 2: UBAR in different settings on MultiWOZ 2.1.

Dialog State Tracking As shown in Table 1, The small gap between the performances in policy optimization (106.0) and end-to-end modeling (105.1) suggests that UBAR is very good at generating belief states. Even though UBAR is designated for response generation and end-to-end modeling, we evaluate the design of UBAR on dialog state tracking task by training a DST-UBAR with training data consisting of user utterances, belief states, database results, system acts and **non-delexicalized** responses of dialog sessions. The belief states are generated by DST-UBAR based on dialog context of user utterances, generated belief states and ground truth system acts and non-delexicalized responses. Table 3 shows that DST-UBAR also outperforms the state-of-the-art methods on both versions of MultiWOZ.

Model	Joint Accuracy (%)	
	MultiWOZ 2.0	MultiWOZ 2.1
TRADE	48.62	45.60
DSTQA	51.44	51.17
DST-Picklist	-	53.3
SST	-	55.23
SimpleTOD	-	55.72
DST-UBAR	**53.04**	**56.20**

Table 3: Comparison of Dialog state tracking (DST) on MultiWOZ 2.0 and 2.1.

Analysis and Discussion

In this section, we try to answer three questions: (1) How much and what kind of dialog context does UBAR need for end-to-end modeling? (2) What are the advantages of the proposed training and evaluating on a dialog session level over turn level? (3) How well can UBAR transfer to unseen domains in end-to-end modeling?

Dialog Context

A large portion of the information of user's goal is stored in belief states. Since UBAR incorporates belief states in the context, it can figure out the new belief states based on just the previous turn and user utterance of the current turn. As shown in Table 4, UBAR based on the previous turn underperforms UBAR based on all previous turns, but still outperforms other state-of-the-art methods. Therefore, UBAR can operate properly with much shorter context length than turn-level methods that require full dialog history, which is more computationally efficient. On the other hand, if UBAR is granted with ground truth belief states in the context, the results would increase slightly. This is because the ground truth belief states in the context make generating belief states of the current turn easier. However, if UBAR takes all ground truth content in the context including system acts and responses, the results actually drop quiet a lot. This is somewhat unexpected yet understandable given the acts and responses in the context are not committed by UBAR, and could mislead UBAR to think that it already committed such acts and responses. We will discuss more on this in the case study next section. In a realistic setting, a TOD system can have access to the context it generated, but not any ground truth, which is why we report all-generated results for overall comparison.

#Turns	Belief	Act	Inf.	Succ.	BLEU	Comb.
All	GT	GT	88.4	76.6	**17.6**	100.1
All	GT	Gen	95.4	**82.3**	17.2	**106.1**
All	Gen	Gen	**95.4**	80.7	17.0	105.1
Prev	GT	GT	87.2	75.3	**16.8**	98.0
Prev	GT	Gen	92.7	**79.0**	16.6	**102.5**
Prev	Gen	Gen	**92.7**	77.7	16.4	101.6

Table 4: Results of UBAR evaluated with different kinds of dialog context in end-to-end setting. *#Turns* denotes the number of previous turns in context, *All* means all previous turn, *Prev* means just the last turn. *GT* or *Gen* denotes if the belief states and system acts in the context are ground truth or generated. We provide a more comprehensive evaluation of UBAR with different kinds of dialog context in multiple settings in the technical appendix.

Session-Level vs. Turn-Level

The main difference between UBAR and other GPT-2-based models is that UBAR is trained on session-level sequences with intermediate information such as belief states and system acts in the context, while others are trained on turn-level sequences with only dialog history of user utterances and system responses. To study the effect of session-level training and the incorporation of belief states and system acts in the context via ablation, we implement a model URUR trained on turn-level sequences. We evaluate URUR in the end-to-end modeling setting where every turn it makes generation conditioned on previous user utterances and system responses. As shown in Table 5, the turn-level URUR underperforms UBAR. Specifically, URUR with ground truth responses in dialog history has comparable performance with SimpleTOD and SOLOIST. What's more, URUR with generated responses in history shows significant improvement over ground truth responses, which suggests that SimpleTOD and SOLOIST miss out a convenient boost evaluating on a turn-level. They could achieve better performances by simply using generated responses in their dialog context.

On the other hand, we constrain UBAR to generate based on the context consisted of only user utterances and responses or only belief states and system acts. With B&A outperforming U&R as well as URUR, we confirm that the belief states and system acts are more important than user utterances and responses in dialog context and that it is more difficult for models to infer belief states and system acts from the dialog history of every turn.

Model	Context	Inf.	Succ.	BLEU	Comb.
URUR	GT	82.6	73.1	**17.0**	94.8
URUR	Gen	**91.2**	**79.5**	16.5	**101.8**
UBAR	U&R	92.5	70.8	14.3	95.9
UBAR	B&A	**94.1**	**77.1**	**16.3**	**101.9**

Table 5: Results in the end-to-end setting. URUR is trained in turn-level. *GT* or *Gen* means it uses ground truth or generated responses in its context. *U&R* denotes the context of UBAR only consists of user utterances and generated responses. *B&A* denotes the context of UBAR only consists of belief states, database results and system acts.

Domain Transfer

The ontology are often shared across domains. For example, *Hotel* and *Restaurant* share the same requestable slots such as address, postcode, price range. Therefore, it is possible for UBAR to generalize to new domains.

To examine the transfer ability of UBAR to generalize to unseen domains, we run zero-shot and few-shot end-to-end modeling experiments by excluding one domain out of the five domains that are available in validation and test set, and training UBAR on other four domains. As shown in Table 6, after trained on 4 domains, the base model (BM) performs generally well in-domain. In zero-shot setting, the performances vary across different domains. The *Train* domain achieves the highest combined score, while *Restaurant* performs badly. This is because *Train* domain has a high overlap in ontology with other domains, while *Restaurant* has a unique food slot which is mentioned frequently.

In few-shot setting, the base model is fine-tuned with 100 dialog sessions from the held-out domain. The few-shot BM is evaluated on the held-out domain and achieves better performance than zero-shot BM, improving the combined score by 20 points in average, which demonstrates the transfer ability of UBAR. However, we also see a drop in the performance of the few-shot BM evaluated on the original four domains, which indicates catastrophic forgetting to some extent. What's more, the big gap between few-shot BM and UBAR trained on all domains underscores the data hungry nature of end-to-end task-oriented dialog modeling.

Visualization and Case Study

In this section we provide an attention weights visualization to demonstrate how UBAR generates belief states based on user utterance and the belief states of the previous turn, and

Evaluation on 4 Domains	Except Hotel	Except Train	Except Attraction	Except Restaurant	Except Taxi
Base Model trained in-domain	99.03	99.40	99.68	**101.87**	95.30
Few-shot BM on new domain	84.88	84.66	96.96	**100.79**	88.14
Evaluation on New Domain	Hotel	Train	Attraction	Restaurant	Taxi
Zero-shot BM	58.40	**64.19**	49.96	40.55	59.03
Few-shot BM on new domain	78.48	70.52	**79.51**	74.19	74.02
UBAR on all domains	103.06	**106.40**	102.24	104.44	103.20

Table 6: Results of domain transfer. The first row is the base model trained on the four domains and evaluated in-domain. The second row is the results of the base model fine-tuned with 100 new domain examples on the four domains. The last three rows are evaluations on the new domains with zero-shot or few-shot BM or UBAR trained on full data, respectively.

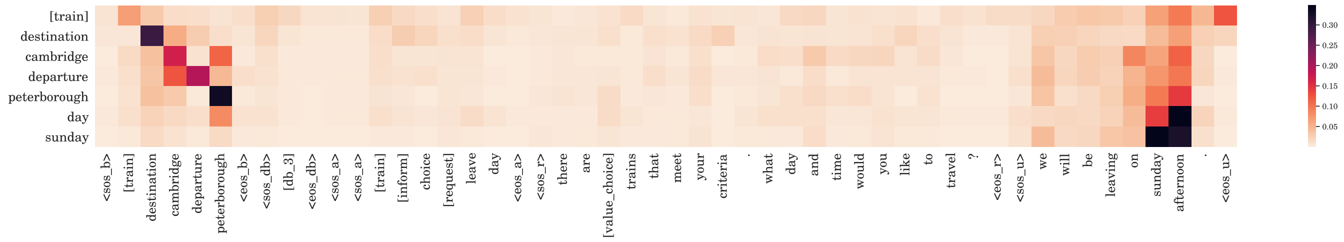


Figure 4: Visualization of attention weights of the generated belief states attending to the context. The X-axis is the belief states, response from the previous turn and the current user utterance. The Y-axis is the generated belief states.

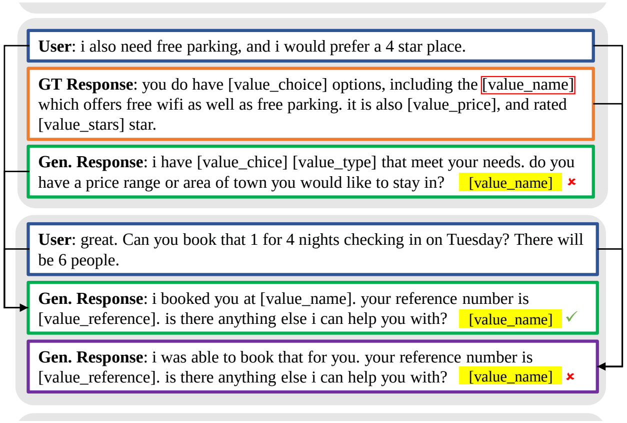
a case study to explain why UBAR can improve task completion by conditioning on the generated context.

Figure 4 visualizes the attention weights of the last layer of the transformer blocks in UBAR, demonstrating that UBAR appropriately attends to the belief state from the previous turn and copies those that remain consistent, which are the destination and departure of a train. Then it attends to user utterance for an update with new specifications, which is the day of the train.

It makes sense to keep track of the belief states and add in or make adjustment to them given incoming user utterance, instead of going through all previous user utterances and responses every turn. We provide more visualizations of the process of generating the belief states, system acts and responses in technical appendix.

Figure 5 shows a case where the user wants to book a 4-stars hotel with free parking. In the first turn, UBAR did not provide the hotel entity, but asked for additional information about the hotel's price range or area. However in the second turn, the user directly requests to book the unmentioned hotel. If using the generated content, UBAR would provide a specific hotel entity in response to the user utterance in the second turn. While if using the ground truth response which contains the hotel entity as the dialog context, UBAR would be mistaken that it had already provided such entity, failing to mention the important entity.

This explains why UBAR with generated context can outperform UBAR with ground truth context, as UBAR can adaptively supplement and make amends in response to the current user utterance in order to stay consistent and coherent throughout the entire session, ultimately improving task completion. More case studies are given in the technical appendix.



User: i also need free parking, and i would prefer a 4 star place.

GT Response: you do have [value_choice] options, including the [value_name] which offers free wifi as well as free parking. it is also [value_price], and rated [value_stars] star.

Gen. Response: i have [value_chice] [value_type] that meet your needs. do you have a price range or area of town you would like to stay in? [value_name] ×

User: great. Can you book that 1 for 4 nights checking in on Tuesday? There will be 6 people.

Gen. Response: i booked you at [value_name]. your reference number is [value_reference]. is there anything else i can help you with? [value_name] ✓

Gen. Response: i was able to book that for you. your reference number is [value_reference]. is there anything else i can help you with? [value_name] ×

Figure 5: Two consecutive dialog turns in dialog session SNG0855 from MultiWOZ2.0. The green boxes and purple box indicate responses of UBAR based on generated context and response based on oracle context, respectively. The task related entities are highlighted in yellow.

Conclusion

In this paper, we attempt to approach end-to-end task-oriented dialog system in a more realistic setting. The proposed UBAR is trained and evaluated on a dialog session level. It generates belief states, system acts and responses based on the user utterances and all content it generated. We conduct extensive experiments and analyses to demonstrate the superiority of the modeling on a dialog session level and the power of GPT-2. We hope that UBAR can inspire more future work to model task-oriented dialog system on a session level.

Acknowledgments

We thank the anonymous reviewers for their constructive reviews. This work was supported by the Fundamental Research Funds for the Central Universities (No.19lgpy220) and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.2017ZT07X355).

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3(6): 1137–1155.

Budzianowski, P.; and Vulić, I. 2019. Hello, It's 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*, 15–22.

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gasic, M. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *EMNLP 2018: 2018 Conference on Empirical Methods in Natural Language Processing*, 5016–5026.

Chen, L.; Lv, B.; Wang, C.; Zhu, S.; Tan, B.; and Yu, K. 2020. Schema-Guided Multi-Domain Dialogue State Tracking with Graph Attention Neural Networks. *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence* 34(5): 7521–7528.

Chen, W.; Chen, J.; Qin, P.; Yan, X.; and Wang, W. Y. 2019. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *ACL 2019 : The 57th Annual Meeting of the Association for Computational Linguistics*, 3696–3709.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 4171–4186.

Eric, M.; Goel, R.; Paul, S.; Kumar, A.; Sethi, A.; Ku, P.; Goyal, A. K.; Agarwal, S.; Gao, S.; and Hakkani-Tur, D. 2019. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In *LREC*, 422–428.

Eric, M.; and Manning, C. D. 2017. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414* .

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1631–1640.

Ham, D.; Lee, J.-G.; Jang, Y.; and Kim, K.-E. 2020. End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In *ACL 2020: 58th annual meeting of the Association for Computational Linguistics*, 583–592.

Henderson, M.; Thomson, B.; and Young, S. 2013. Deep Neural Network Approach for the Dialog State Tracking Challenge. In *Proceedings of the SIGDIAL 2013 Conference*, 467–471.

Henderson, M.; Thomson, B.; and Young, S. 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)*, 292–299.

Hosseini-Asl, E.; McCann, B.; Wu, C.-S.; Yavuz, S.; and Socher, R. 2020. A Simple Language Model for Task-Oriented Dialogue. *arXiv preprint arXiv:2005.00796* .

Lei, W.; Jin, X.; Kan, M.-Y.; Ren, Z.; He, X.; and Yin, D. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *ACL 2018: 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, 1437–1447.

Liu, B.; and Lane, I. 2018. End-to-End Learning of Task-Oriented Dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, 67–73.

Madotto, A.; Wu, C.-S.; and Fung, P. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217* .

Mehri, S.; Srinivasan, T.; and Eskenazi, M. 2019. Structured Fusion Networks for Dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, 165–177.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, 311–318.

Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Liden, L.; and Gao, J. 2020a. SOLOIST: Few-shot Task-Oriented Dialog with A Single Pre-trained Auto-regressive Model. *arXiv preprint arXiv:2005.05298* .

Peng, B.; Zhu, C.; Li, C.; Li, X.; Li, J.; Zeng, M.; and Gao, J. 2020b. Few-shot Natural Language Generation for Task-Oriented Dialog. *arXiv preprint arXiv:2002.12328* .

Peskov, D.; Clarke, N. E.; Krone, J.; Fodor, B.; Zhang, Y.; Youssef, A.; and Diab, M. T. 2019. Multi-Domain Goal-Oriented Dialogues (MultiDoGO): Strategies toward Curating and Annotating Large Scale Dialogue Data. 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)*, 4525–4535.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. DEEP CONTEXTUALIZED WORD REPRESENTATIONS. In *NAACL HLT 2018: 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, 2227–2237.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8): 9.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* .

Shah, P.; Hakkani-Tur, D.; Liu, B.; and Tur, G. 2018. Bootstrapping a Neural Conversational Agent with Dialogue Self-Play, Crowdsourcing and On-Line Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, volume 3, 41–51.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5998–6008.

Wang, K.; Tian, J.; Wang, R.; Quan, X.; and Yu, J. 2020. Multi-Domain Dialogue Acts and Response Co-Generation. In *ACL 2020: 58th annual meeting of the Association for Computational Linguistics*, 7125–7134.

Wen, T.-H.; Gasic, M.; Mrkšić, N.; Su, P.-H.; Vandyke, D.; and Young, S. 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*, 1711–1721.

Wen, T.-H.; Miao, Y.; Blunsom, P.; and Young, S. 2017. Latent Intention Dialogue Models. In *ICML'17 Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 3732–3741.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; and Brew, J. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771* .

Wu, C.-S.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; and Fung, P. 2019a. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *ACL 2019 : The 57th Annual Meeting of the Association for Computational Linguistics*, 808–819.

Wu, C.-S.; Socher, R.; and Xiong, C. 2019. Global-to-local memory pointer networks for task-oriented dialogue. *arXiv preprint arXiv:1901.04713* .

Wu, Q.; Zhang, Y.; Li, Y.; and Yu, Z. 2019b. Alternating Recurrent Dialog Model with Large-Scale Pre-Trained Language Models. *arXiv preprint arXiv:1910.03756* .

Wu, Z.; Galley, M.; Brockett, C.; Zhang, Y.; Gao, X.; Quirk, C.; Koncel-Kedziorski, R.; Gao, J.; Hajishirzi, H.; Ostendorf, M.; and Dolan, B. 2020. A Controllable Model of Grounded Response Generation. *arXiv preprint arXiv:2005.00613* .

Xu, P.; and Hu, Q. 2018. An End-to-end Approach for Handling Unknown Slot Values in Dialogue State Tracking. In *ACL 2018: 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, 1448–1457.

Zhang, J.; Hashimoto, K.; Wu, C.-S.; Wan, Y.; Yu, P. S.; Socher, R.; and Xiong, C. 2019a. Find or Classify? Dual Strategy for Slot-Value Predictions on Multi-Domain Dialog State Tracking. *arXiv preprint arXiv:1910.03544* .

Zhang, Y.; Ou, Z.; and Yu, Z. 2020. Task-Oriented Dialog Systems that Consider Multiple Appropriate Responses under the Same Context. *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence* 34(5): 9604–9611.

Zhang, Y.; Sun, S.; Galley, M.; Chen, Y.-C.; Brockett, C.; Gao, X.; Gao, J.; Liu, J.; and Dolan, B. 2020a. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 270–278.

Zhang, Z.; Liao, L.; Huang, M.; Zhu, X.; and Chua, T.-S. 2019b. Neural Multimodal Belief Tracker with Adaptive Attention for Dialogue Systems. In *The World Wide Web Conference on*, 2401–2412.

Zhang, Z.; Takanobu, R.; Huang, M.; and Zhu, X. 2020b. Recent Advances and Challenges in Task-oriented Dialog System. *arXiv preprint arXiv:2003.07490* .

Zhao, T.; Xie, K.; and Eskenazi, M. 2019. Rethinking Action Spaces for Reinforcement Learning in End-to-end Dialog Agents with Latent Variable Models. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 1208–1218.

Zhong, V.; Xiong, C.; and Socher, R. 2018. Global-Locally Self-Attentive Encoder for Dialogue State Tracking. In *ACL 2018: 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, 1458–1467.

Zhou, L.; and Small, K. 2019. Multi-domain Dialogue State Tracking as Dynamic Knowledge Graph Enhanced Question Answering. *arXiv preprint arXiv:1911.06192* .