

Recent Advances and Challenges in Task-oriented Dialog System

Zheng Zhang, Ryuichi Takanobu, Minlie Huang* & Xiaoyan Zhu

*Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China;
Institute for Artificial Intelligence, Tsinghua University (THUAI), Beijing 100084, China;
Beijing National Research Center for Information Science & Technology, Beijing 100084, China*

Abstract Due to the significance and value in human-computer interaction and natural language processing, task-oriented dialog systems are attracting more and more attention in both academic and industrial communities. In this paper, we survey recent advances and challenges in an issue-specific manner. We discuss three critical topics for task-oriented dialog systems: (1) improving data efficiency to facilitate dialog system modeling in low-resource settings, (2) modeling multi-turn dynamics for dialog policy learning to achieve better task-completion performance, and (3) integrating domain ontology knowledge into the dialog model in both pipeline and end-to-end models. We also review the recent progresses in dialog evaluation and some widely-used corpora. We believe that this survey can shed a light on future research in task-oriented dialog systems.

Keywords task-oriented dialog, low-resource, dialog state tracking, dialog policy, end-to-end model

Citation Zhang Z, Takanobu R, Huang M, Zhu X. Recent Advances and Challenges in Task-oriented Dialog System. *Sci China Inf Sci*, for review

1 Introduction

Building task-oriented (also referred to as goal-oriented) dialog systems has become a hot topic in the research community and the industry. A task-oriented dialog system aims to assist the user in completing certain tasks in a specific domain, such as restaurant booking, weather query, and flight booking, which makes it valuable for real-world business. Compared to open-domain dialog systems where the major goal is to maximize user engagement [1], task-oriented dialog systems are more targeting at accomplishing some specific tasks in one or multiple domains. Typically, the task-oriented dialog systems are built on top of a structured ontology, which defines the domain knowledge of the tasks.

1.1 General Framework

The architecture of task-oriented dialog systems can be roughly divided into two classes: pipeline and end-to-end approaches. In pipeline approaches, the model often consists of several components, including *Natural Language Understanding* (NLU), *Dialog State Tracking* (DST), *Dialog Policy*, and *Natural Language Generation* (NLG), which are combined in a pipeline manner as shown in Figure 1. The NLU, DST and NLG components are often trained individually before being aggregated together, while the dialog policy component is trained within the composed system. It is worth noting that although the NLU-DST-Policy-NLG framework is a typical configuration of the pipeline system, there are still some

* Corresponding author (email: aihuang@tsinghua.edu.cn)

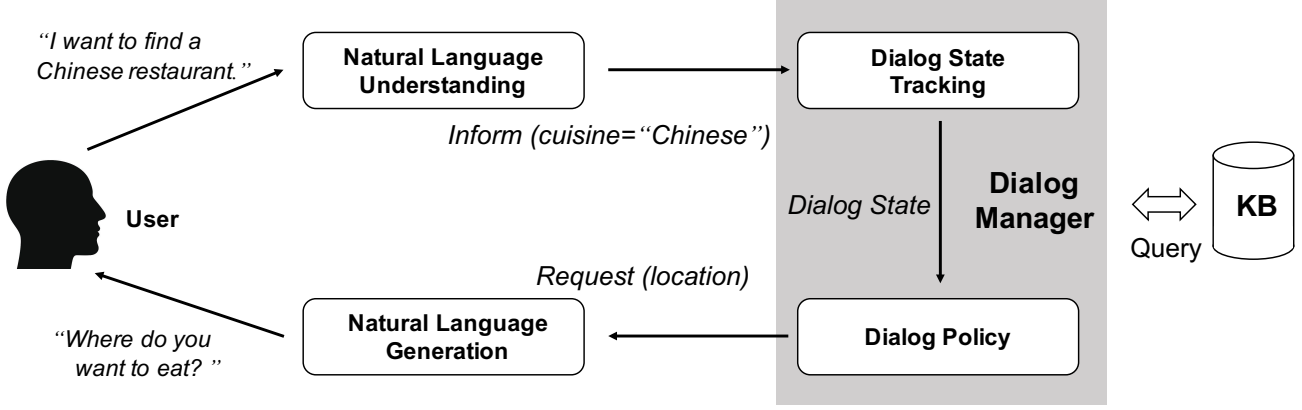


Figure 1 General framework of a pipeline task-oriented dialog system.

other kinds of configurations. Recently, there are works that merge some of the typical components, such as word-level DST and word-level policy, resulting in various pipeline configurations.

In end-to-end approaches, the dialog systems are trained in an end-to-end manner, without specifying each individual component. Commonly, the training process is formulated as generating a responding utterance given the dialog context and the backend knowledge base.

1.1.1 Natural Language Understanding

Given a user utterance, the natural language understanding (NLU) component maps it to a structured semantic representation. A popular schema for semantic representation is the dialog act, which consists of intent and slot-values, as illustrated in Table 1. The intent type is a high-level representation of an utterance, such as *Query* and *Inform*. Slot-value pairs are the task-specific semantic elements that are mentioned in an utterance. Based on the dialog act structure, the task of NLU can be further decomposed into two tasks: intent detection and slot-value extraction. The former is normally formulated as an intent classification task by taking the utterance as input, and the slot-value extraction task is often viewed as a sequence labeling problem:

$$p_{intent}(d|x_1, x_2, \dots, x_n) \quad (1)$$

$$p_{slot}(y_1, y_2, \dots, y_n|x_1, x_2, \dots, x_n) \quad (2)$$

where the d indicates intent class and y_1 to y_n are the labels of each token in the utterance $[x_1, x_2, \dots, x_n]$ in which x_i is a token and n means the number of tokens. p_{intent} and p_{slot} are often implemented using recurrent neural networks, such as LSTM, to predict the intent class d and the sequence label y_t respectively.

Table 1 An example of dialog act for an utterance in the restaurant reservation domain.

Utterance	How about a <i>British</i> restaurant in <i>north</i> part of town.
Intent	<i>Query</i>
Slot Value	Cuisine= <i>British</i> , Location= <i>North</i>

1.1.2 Dialog State Tracking

The dialog state tracker estimates the user's goal in each time step by taking the entire dialog context as input. The dialog state at time t can be regarded as an abstracted representation of the previous turns until t . Most existing works adopted belief state for dialog state representation, in which the state is composed of several probability distributions over the value vocabulary of each slot. Therefore, this

problem can be formulated as a multi-task classification task:

$$p_i(d_{i,t}|u_1, u_2, \dots, u_t) \quad (3)$$

where for each specific slot i , there is a tracker p_i . u_t represents the utterance in turn t . The class of slot i in the t -th turn is $d_{i,t}$. However, this approach falls short when facing previously unseen values at running time. To mitigate this issue, there are some generative approaches that generalize well on new domains and previous unseen values.

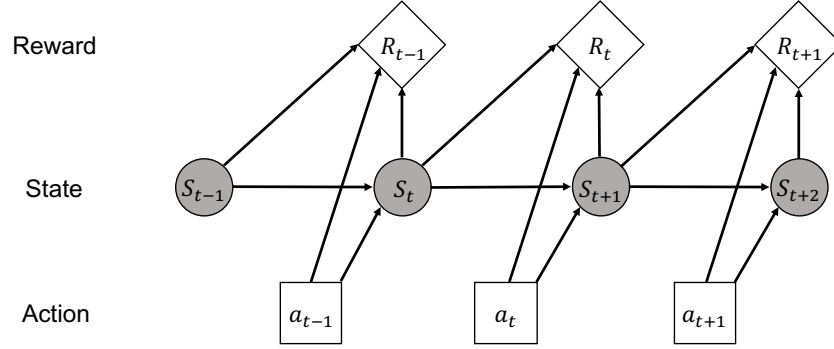


Figure 2 The framework of Markov Decision Process [2]. At time t , the system takes an action a_t , receiving a reward R_t and transferring to a new state S_{t+1} .

1.1.3 Dialog Policy

Conditioned on the dialog state, the dialog policy generates the next system action. Since the dialog acts in a session are generated sequentially, it is often formulated as a Markov Decision Process (MDP), which can be addressed by Reinforcement Learning (RL). As illustrated in Figure 2, at a specific time step t , the user takes an action a_t , receiving a reward R_t and the state is updated to S_t .

A typical approach is to first train the dialog policy off-line through supervised learning or imitation learning based on the dialog corpus, and then fine-tune the model through RL with real users. Since real user dialogs are costly, user simulation techniques are introduced to provide affordable training dialogs.

1.1.4 Natural Language Generation

Given the dialog act generated by the dialog policy, the natural language generation component maps the act to a natural language utterance, which is often modeled as a conditioned language generation task [3]. To improve user experience, the generated utterance should (1) fully convey the semantics of a dialog act for task-completion, and (2) be natural, specific, and informative, analogous to human language.

1.1.5 End-to-end Methods

The end-to-end approaches for task-oriented dialog systems are inspired by the researches on open-domain dialog systems, which use neural models to build the system in an end-to-end manner without modular design, as shown in Figure 3. Most of these methods utilized seq2seq neural networks as the infrastructural framework. End-to-end formulation can avoid the problem of error propagation within cascaded components during training.

1.2 Main Challenges

Recent advances in task-oriented dialog systems are dominated by neural approaches. These approaches can be roughly classified into two genres: pipeline and end-to-end methods.

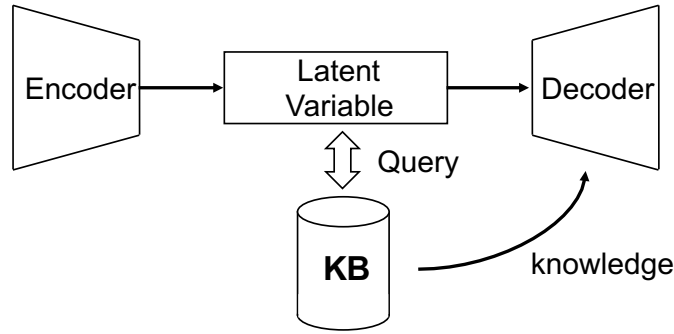


Figure 3 The framework of end-to-end dialog systems. It first encodes natural language context to get some latent variables, which can be used for KB query. Then based on the latent variables and query results, the decoder generates a natural language response.

In pipeline methods, recent researches focus more on the dialog state tracking and dialog policy components, which are also called *Dialog Management*. This is because both NLU and NLG components are standalone language processing tasks, which is less interweaved to the task in dialog systems. Based on the domain ontology, the DST task can be seen as a classification task by predicting the value of each slot. However, when the training data is not sufficient, such classification-based methods can suffer from the out-of-vocabulary (OOV) problem and can not be directly generalized to new domains. The dialog policy learning task is often considered as a reinforcement learning task. Nevertheless, different from other well-known RL tasks, such as playing video games and Go, the training of dialog policy requires real humans to serve as the environment, which is very costly. Furthermore, most existing methods used manually defined rewards, such as task-completion rate and session turn number, which cannot reliably evaluate the performance of a system.

For end-to-end methods, the data-hungry nature of the vanilla sequence-to-sequence model makes it difficult to learn the sophisticated slot filling mechanism in task-oriented dialog systems with a limited amount of domain-specific data. The knowledge base query issue requires the model to generate an intermediate query besides the encoder and the decoder, which is not straightforward. Another drawback is that the encoder-decoder framework utilizes a word-level strategy, which may lead to sub-optimal performance because the strategy and language functions are entangled together.

Based on the above analysis, we elaborate three key issues in task-oriented dialog systems which we will discuss in detail:

- **Data Efficiency** Most neural approaches are data-hungry, requiring a large amount of data to fully train the model. However, in task-oriented dialog systems, the domain-specific data is often hard to collect and expensive to annotate. Therefore, the problem of low-resource learning is one of the major challenges.
- **Multi-turn Dynamics** The core feature of task-oriented dialog as compared to open-domain dialog is its emphasis on goal-driven in multi-turn strategy. In each turn, the system action should be consistent with the dialog history and should guide the subsequent dialog to larger task reward. Nevertheless, the model-free RL methods which have shown superior performance on many tasks, can not be directly adopted to task-oriented dialog, due to the costly training environment and imperfect reward definition. Therefore, many solutions are proposed to tackle these problems in multi-turn interactive training to learn a better policy, including model-based planning, reward estimation and end-to-end policy learning.
- **Knowledge Integration** A task-oriented dialog system has to query the knowledge base (KB) to retrieve some entities for response generation. In pipeline methods, the KB query is mostly constructed according to DST results. Compared to pipeline models, the end-to-end approaches

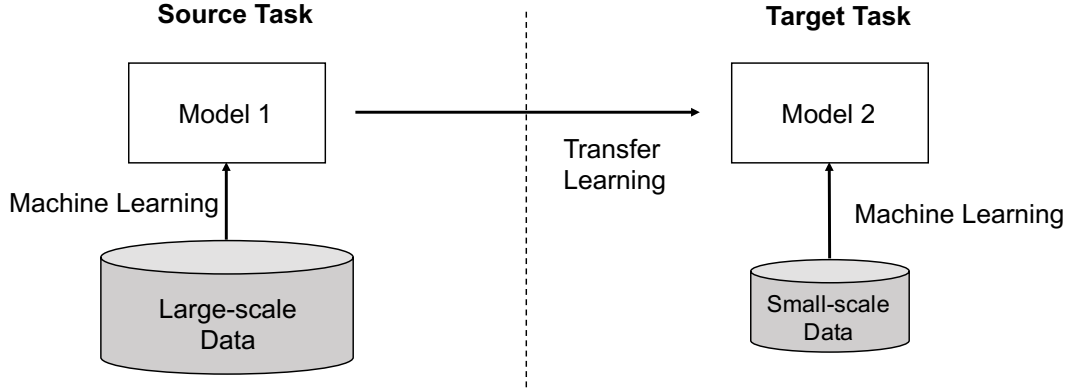


Figure 4 Transfer learning from a high-resource target task to a low-resource target task.

bypass modular models whrequireires fine-grained annotation and domain expertise. However, this simplification is questionable since there is no explicit state representation to construct a query.

2 Data Efficiency

Different from the research in open-domain dialog systems, data-driven approaches for task-oriented dialog systems often require fine-grained annotations to learn the dialog model in a specific domain, e.g., dialog act and state labels. However, it is often difficult to obtain a large-scale annotated corpus in a specific domain since (1) collecting a domain-specific corpus is more difficult than in open-domain setting due to its task-specific features, and (2) annotating fine-grained labels requires a large amount of human resources which is very expensive and time-consuming. Therefore, we often have to face the problem of improving the data efficiency of building task-oriented dialog systems, particularly in a low-resource setting.

In this section, we review some recent approaches proposed to mitigate this issue. We first review transfer learning methods that acquire prior knowledge from large-scale data or reliable trained models from other tasks. Then, we introduce some unsupervised methods, which can directly learn in a low-resource setting with few annotations through heuristic rules. In addition, we also review recent efforts on building data-driven user simulators.

2.1 Dialog Transfer Learning

One major assumption of machine learning is that the training and test data should have the same distribution. However, in many real-world scenarios, this may not hold when we have only limited data in the target task but sufficient data in another task with different data distribution. Transfer learning is thus proposed to mitigate this problem by transferring knowledge from the source task to the target task.

The same issue often occurs in task-oriented dialog systems. For example, how can a dialog system for restaurant reservation be adapted to hotel booking when there are only limited data in the hotel domain. In such a situation, the two domains' ontologies are similar, sharing many dialog acts and slots. In this setting, transfer learning can considerably reduce the amount of target data required for this adaptation. Besides domain-level transfer, knowledge can also be transferred in many other dimensions, including inter-person and cross-lingual transfer. For domain transfer, Mrkšić et al. [4] proposed to learn the dialog state tracking model through multi-task learning on multiple domain datasets to transfer knowledge across domains, which can improve the performance on all tasks. In [5], Ilievski et al. proposed to directly transfer the parameters of shared slots from the source domain model to initialize the target model. For transferring across disjoint tasks, Mo et al. [6] proposed to transfer the dialog policy model

between domains where there are no shared slots by learning act and state transfer functions, which directly maps from the source feature space to the target space.

For personalized knowledge transfer, in [7], a hybrid DQN policy is proposed to transfer knowledge across different customers, in which there is a general Q-function for all customers and a personalized one for each specific customer. When transferring to a new person, only a small amount of data is required to learn the personalized Q-function. Mo et al. [8] further transfers finer granularity phrase-level knowledge between different persons while keeping personal preferences of each user intact by designing a novel personal control gate within the RNN decoder framework.

The research on cross-lingual transfer is recently motivated by the demand of multinational corporations. In [9], three cross-lingual methods are studied: (1) Translating the training data to the target language, (2) Pre-training cross-lingual embeddings and (3) Using a multilingual machine translation encoder to share knowledge for contextual word representations.

Model-agnostic methods are also proposed for transfer learning in dialog systems, which are mostly inspired by the Model-Agnostic Meta-Learning (MAML) framework [10]. The MAML framework can learn a good initialized model by simulating the train-test procedure during learning. By applying such methods on NLG, the model can get better results in a low-resource setting and show better domain generalization [11, 12]. [13] further extended this method for personalized dialog systems by leveraging only a few dialogue samples collected from the target user without using the persona-specific descriptions.

2.2 Unsupervised Methods

A crucial issue in dialog policy learning is the reward supervision, which is hard to be obtained in real-world applications. Therefore, building a reward estimation model to provide reward signal during RL training is necessary. By regarding the dialog policy as a generator and the reward function as a discriminator, generative adversarial nets (GAN) can be employed to learn the reward function in an unsupervised manner. Liu et al. [14] first used GAN to learn a binary reward function by discriminating simulated from real user dialogs. [15] extended this idea for failure dialog detection by using the predicted reward as an indicator of failure. Su et al. [16] used another way for reward estimation using Gaussian Process. By modeling the uncertainty of predicted reward, the model can actively require human intervention on potential failed cases. In their experiment, the requirement for human intervention dramatically decreases with the reduction in reward estimation uncertainty, which notably lightens manpower.

In most studies, the ontology of a dialog system is built by human experts through elaborate domain engineering. Another line of work is to assist the human experts in this process by learning the dialog structure from unlabeled corpus automatically. Shi et al. [17] proposed to learn a finite state machine of the dialog procedure through a variational autoencoder (VAE) based approach. They first pre-trained a VAE based dialog model using raw dialog corpus without intermediate annotations. Then several dialog states can be discovered according to the latent variables. After that, a state transition diagram can be built by estimating the transition probabilities between states. There are also some works analyzing the structure of task-oriented to facilitate language understanding. [18] proposed an RL method for topic segmentation and labeling in task-oriented dialog, which aims to detect topic boundaries among dialogue turns and assign topic labels to them.

Recently, pre-training methods show superior performance on many NLP tasks. In such approaches, extensive linguistic features can be transferred from large-scale unlabeled corpora using unsupervised pre-training tasks, such as mask language modeling (MLM) and next sentence prediction (NSP). [19] followed this way by first pre-training a transformer model on large-scale dialog data and then fine-tuning the model on a personalized dialog task with multi-task learning. [20] further explored this idea to task-oriented dialog without explicit standalone dialogue policy and generation modules. In this work, the belief state and database state are first converted to natural language text and then taken as input to the transformer decoder besides the context.

2.3 User Simulation

User simulation techniques alleviate the data-hungry issue of the RL-based dialog policy model by providing a theoretically infinite number of training interactions. Early approaches focused on agenda-based user simulator (ABUS) [21], which is commonly used in building task-oriented dialog. It maintains a stack-like structure representing the user’s goal with some heuristics. Building an agenda-based simulator requires the human expert to define the agenda and heuristics rules explicitly. However, for more complex tasks, it is not feasible to define an explicit agenda structure. Utterances from ABUS also lack the linguistic variations of human dialogs, which may lead to suboptimal performance in real applications.

Recently, people studied building user simulators in a data-driven fashion to alleviate the above issues. Asri et al. [22] proposed a dialog act level seq2seq user simulation model that takes into account the dialog context. Crook et al. [23] presented another seq2seq model which takes as input natural language context and outputs natural language response. Kreyssig et al. [24] introduced a neural user simulator (NUS), which mimics the user behavior of the corpus and generates word-level user responses. Gur et al. [25] proposed hierarchical seq2seq user simulator (HUS) that first encodes the user goal and system turns, and generates user dialog act. To generate more diverse user acts, they extend HUS to a variational version (VHUS) where the user turn is generated from an unobservable latent variable.

Another line of data-driven user simulators trains the simulator together with the target dialog system, which can be regarded as a multi-agent fashion. Liu et al. [26] proposed to first train the dialog system and the simulator based on the dialog corpus through supervised learning, and then fine-tune both models by reinforcement learning. In this work, the system and the simulator are trained cooperatively, in which both agents share the same reward function. The world model in the Deep Dyna-Q (DDQ) based dialog planning framework [27–29], which is updated during training, can also be regarded as a simulator. However, different from RL-based co-training, the world model in DDQ is updated through supervised learning using real experience.

3 Multi-turn Dynamics

Compared to open-domain dialog systems, one major feature of task-oriented dialog research is its emphasis on multi-turn state-action dynamics. In open-domain dialog systems, the research focuses more on generating reasonable, consistent, and semantically-rich responses to maximize user engagement. While for task-oriented dialog systems, although the above issues are still important, the completion of a specific task has viewed as more critical. Therefore, the research on dialog management, which is responsible for tracking the dialog state and flow of the conversation, and acts as the pillar of a dialog system.

Human conversation can be formulated as a Markov Decision Process (MDP): at each time step, the system transits from some state s to a new state s' by taking certain action a . Therefore, reinforcement learning is often applied to solve such an MDP problem in the dialog systems. Recent studies on the dynamics of task-oriented dialog systems are mainly focused on the following topics: (1) planning for dialog policy learning for better sample efficiency, (2) reward estimation for the reward sparsity issue, and (3) end-to-end dialog policy learning.

3.1 Planning for Dialog Policy Learning

Model-free RL methods dominated the early studies of neural dialog policy by learning through interaction with real users. It is data-hungry, requiring a large amount of interactions to train a policy model effectively. One common solution is to use user simulators. However, the user simulator is not able to fully mimic real human conversation behaviors, and its inductive bias may lead to sub-optimal models that perform badly in real human conversation.

To alleviate these problems, model-based RL methods are proposed to model the environment, enabling planning for dialog policy learning. In model-based RL approaches, the environment is modeled to

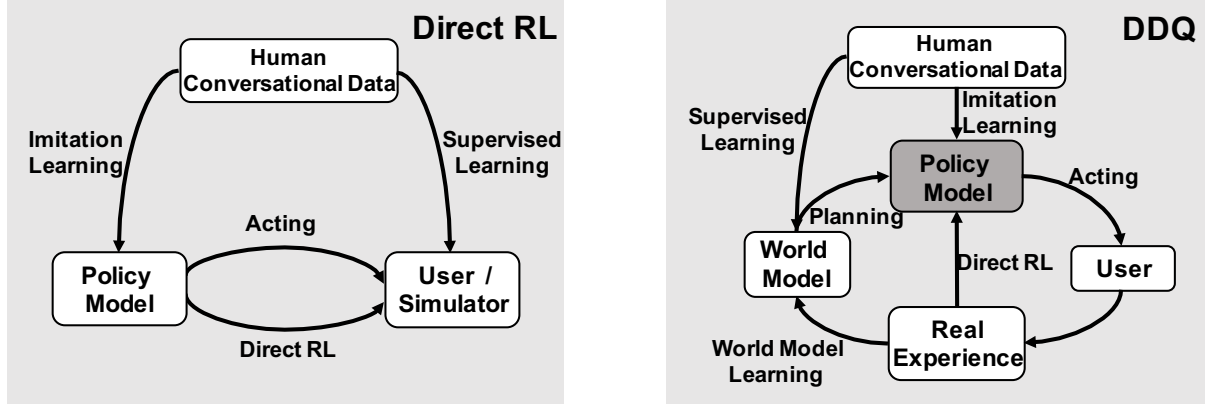


Figure 5 The framework of Deep Dyna-Q (DDQ) [27] planning as compared to direct RL framework. In DDQ, a world model is presented which can learn from real experience and provide simulated experience for policy learning.

simulate the dynamics of the conversation. Then in the RL training phase, the dialog policy is alternately trained through learning from real users and planning with the environment model. Peng et al. [27] proposed the Deep Dyna-Q (DDQ) framework, which first integrates model-based planning for task-oriented dialog systems, as illustrated in Figure 5. In the DDQ framework, there is a *world model*, which is trained on real user experience to capture the dynamics of the environment. The dialog policy is trained through both direct RL with real user and simulated RL with the world model. During training, the world model is also updated through supervised learning based on the increasing real experience. The performance of the world model, which is crucial for policy learning, continues to improve during training. However, the ratio of real vs. simulated experience used for Q-learning is fixed in the original DDQ framework. Therefore, controlled planning [28,29] is proposed to mitigate this issue by dynamically adjusting the ratio of real to simulated experiences according to the performance of the world model.

The above methods for planning are referred to as *background planning*, which improves the policy through training on simulated experience with the world model. Another line of planning-based research is *decision time planning*, which directly decides which action to take in a specific state S_t based on some simulated experience. Planning used in this way can look much deeper than one-step ahead at decision time, which is common in human activities. Taking the chess game for example, the players often conduct mental simulation by looking several steps ahead and then decide how to move the pieces. [30,31] introduced *dialog rollout* planning for negotiation, in which the agent simulates complete dialogues in a specific state S_t for several candidate responses to get their expected reward, and the response with the highest reward is taken. Instead of completing the dialogs and obtaining explicit rewards, [32] proposed to look only several limited steps ahead and use those steps as additional features for the policy model to make a decision.

3.2 Reward Estimation

In RL-based dialog models, the reward is crucial for policy learning. One typical approach of reward function definition is to assign a large positive reward at the end of a successful session and a small negative penalty for each turn to encourage short conversations [33]. However, in real-world applications where the user goal is not available, this reward can not be estimated effectively. Another problem is that the reward signals are not consistent when they are objectively calculated by predefined rules or subjectively given by real users. To alleviate the above issues, there are some studies that learn an independent reward function to provide a reliable supervision signal.

One method for reward estimation is off-line learning with annotated data [34]. By taking the dialog utterances and intermediate annotations as input features, reward learning can be formulated as a supervised regression or classification task. The annotated reward can be obtained from either human annotation or user simulator. However, since the input feature space is complicated, a large amount of

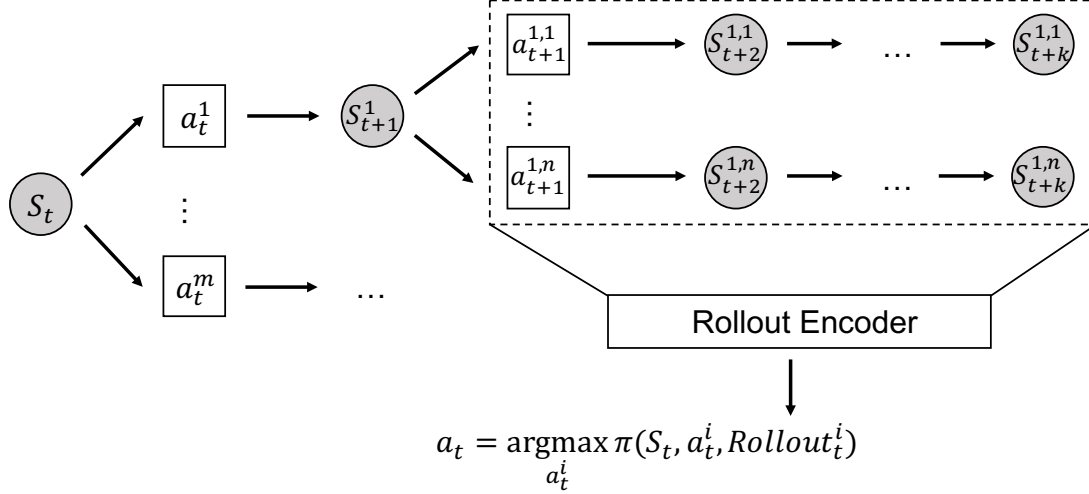


Figure 6 The framework of decision time planning using dialog rollout. At a specific state S_t , the policy generates m candidate actions. For each candidate action a_t^i , the model rollout n sessions for k steps (or until the end of dialog). The rollout turns are further encoded as additional features Rollout_t^i of the dialog policy.

manual annotation is required, which is too costly.

To resolve the above problems, there is another line of work using on-line learning for reward estimation [16]. Reward estimation is often formulated as a Gaussian Process regression task, which can additionally provide an uncertainty measure of its estimation. In this setting, active learning is adopted to reduce the demand for real reward signals in which the users are only asked to provide feedback when the uncertainty score exceeds a threshold. In other cases, when the estimation uncertainty is small, the estimated reward is utilized.

Instead of estimating the reward signals through annotated labels, Inverse RL (IRL) aims to recover the reward function by observing expert demonstrations. Adversarial learning is often adopted for dialog reward estimation through distinguishing simulated and real dialogs [14, 15, 35].

3.3 End-to-end Dialog Policy Learning

End-to-end dialog methods break down the component barriers and model the entire system in an end-to-end fashion. It often adopts word-level policy, which couples the strategy and language generation together as a decoder [30, 36–38] like in many open-domain approaches.

However, compared to the open-domain dialog, the strategy in task-oriented dialog is crucial for task-completion, and coupling the two operations together can harm the performances on both sides. For a specific training example, the loss can come from either strategy or language fault, but the backpropagation on parameters influence both sides. Therefore, there works trying to disentangle the learning of strategy and language in the end-to-end framework. [31] proposed to decouple the semantics of the dialogue utterance from its linguistic realization with a latent variable model. He et al. [39] presents a more explicable method by explicit modeling the current and future dialog acts and generates responses based on the estimated acts. In each turn, the model first decides the act of current utterance and predicts the next act that the system should take, which is similar to the policy component in the pipeline framework but is trained in an end-to-end fashion.

In the original seq2seq models, the dialog state is the encoder output, which is a distributed vector representation. This representation, if learned successfully, should capture the semantics of dialog context. However, by training the dialog model in an end-to-end manner without intermediate supervision, it is hard to be learned. Another drawback of implicit state representation is that explicit knowledge query is hard to conduct. Therefore, there are some works leveraging the advantage of both classic pipeline and seq2seq models. Lei et al. [40] presented Sequicity, in which an explicit state representation, namely

belief span, is produced before decoding. A two-stage CopyNet is proposed, in which a belief span is first generated through coping from user input, and then the system response is generated by copying from the belief span. In [41], a composite state representation is proposed for enabling soft-KB query. For each specific slot, there is a distributed vector representing its state in each turn.

4 Knowledge Integration

One major issue in task-oriented dialog systems is to integrate domain knowledge into the dialog model, as illustrated in Figure 7. In the pipeline framework, the knowledge is first extracted from the knowledge base by queries that are formulated according to the DST results. Then, in the NLG process, the inquired entities are integrated into the natural language response by filling the delexicalized placeholders. Therefore, the knowledge integration of pipeline systems is more closely related to DST. Traditional neural methods formulated DST as a classification task, which can only handle slot values in the ontology. However, the discriminative DST falls short when facing OOV values and transferring to new domains where there are previously unseen slots.

One drawback of pipeline approaches is that it is hard to be scaled to new domains, as each component in the dialog system should be rebuild. Recently, there are many efforts to build end-to-end models for task-oriented dialog systems. In these approaches, the entire model can be trained in an end-to-end fashion, thus avoid error propagation from cascaded components. However, knowledge integration for end-to-end models becomes more challenging, because it is not trivial to involve context information and knowledge base since, different from pipeline methods, there is no explicit dialog state representation to generate an explicit knowledge base query.

In this section, we introduce some recent advances on (1) generative methods for DST for better generalization and (2) knowledge integration in end-to-end systems.

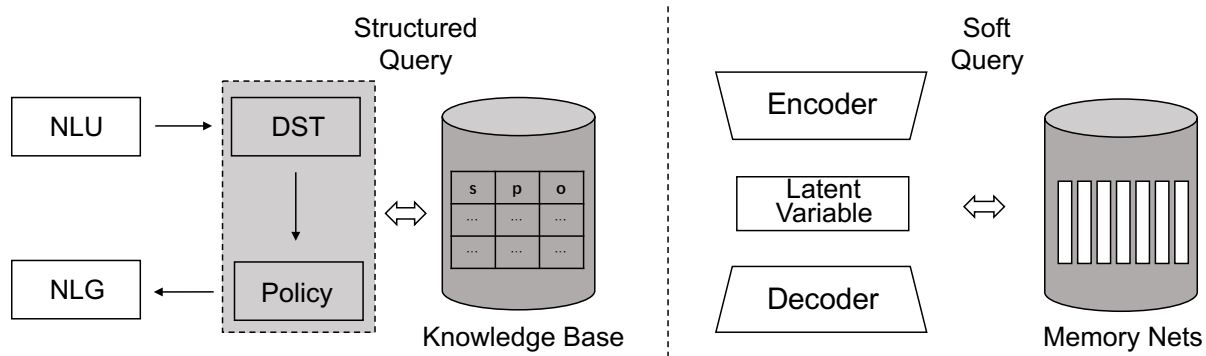


Figure 7 The knowledge base integration frameworks of the pipeline methods and end-to-end methods. In the pipeline methods, a structured query is generated based on the dialog management components. The end-to-end methods apply soft-lookup on the knowledge base using their intermediate latent variables.

4.1 Generative DST

Dialog state tracker plays a central role in task-oriented dialog system by keeping track of a structured dialog state representation in each turn. Most recent DST studies applied a word-level structure by taking natural language as input without NLU, avoiding the error propagated from the NLU component. In early neural DST methods, *belief state* is adopted for dialog state representation [42], which maintains a distribution over all possible states for each slot. Therefore, early methods commonly formulated DST as a classification task [4, 43–46]. Matthew et al. [43] first proposed to use recurrent neural networks for word-level dialog state tracking by taking both natural language utterances and ASR scores as input

features. Nikola et al. [44] proposed Neural Belief Tracker (NBT), a word-level dialog state tracker that directly reads from natural language utterances. NBT explicitly modeled the *system request* and *system confirm* operations through a gating mechanism. However, these approaches can only deal with pre-defined slot values in the domain ontology vocabulary, which generally fall short in tracking unknown slot values during inference.

Zhong et al. [47] proposed to share parameters across slots and learn slot-specific features through a globally-locally self-attention mechanism, which can generalize to rare values with few training data. However, the rare values are still in-vocabulary words. Lei et al. [40] use a seq2seq model with two-stage CopyNet to generate belief spans and response at the same time, which obtain satisfactory results in OOV cases. In the first stage, a belief state CopyNet takes the user utterance as input and generates a belief state span. Then in the second stage, based on the utterance and belief span, another CopyNet generates the response utterance. Hu et al. [48] proposed to use pointer network [49] to extract unknown slot values, which showed superior performance over discriminative DST methods.

Recently, some multi-domain datasets are proposed to promote the research in this direction. Compared to single-domain tasks, the DST in multi-domain scenario has to predict the domain of slot values. Wu et al. [50] propose to use a seq2seq model with CopyNet [51] for state tracking. The parameters are shared across domains, enabling zero-shot DST for unseen domains.

4.2 End-to-end Knowledge Integration

Different from pipeline approaches, there is no explicit structured dialog state representation in end-to-end methods. Therefore, the knowledge base interaction is conducted by using intermediate latent representation of the model and trained seamlessly through end-to-end training. CopyNet and end-to-end memory networks are widely used for integrating knowledge into dialog systems through the attention mechanism. The copy mechanism, however, can also be regarded as a memory network in which the encoder hidden states consist of the memory units. Eric et al. [36] presented a copy-based method depending on the latent neural embedding to attend to dialog history and copy relevant prior context for decoding. However, they can only generate entities that are mentioned in the context. More recent works use memory networks for prior dialog context and knowledge integration [37, 52]. In such approaches, the dialog context and knowledge base are modeled into two memory nets. Then in the decoding phase, the decoder's hidden state is used to selectively query and copy information from those memory nets. A key problem in such a method is that dialog context and knowledge base are heterogeneous information from different sources. Lin et al. [53] proposed to model heterogeneous information using historical information, which is stored in a context-aware memory, and the knowledge base tuples are stored in a context-free memory. In [38], a two-step KB retrieval is proposed to improve the entities' consistency by first deciding the entity row and then selecting the most relevant KB column.

Besides fully end-to-end methods with few intermediate supervision, there are also some end-to-end models integrating domain prior knowledge into the model through dialog act and belief state annotations. Williams et al. [54] proposed hybrid code networks (HCNs), which combines an RNN with domain knowledge encoded as software and templates, which can considerably reduce the training data required. Wen et al. [55] presented a modularized end-to-end task-oriented dialog model by combining several pre-trained components together, and then fine-tuning the model using RL in an end-to-end fashion. However, compared to seq2seq models, these methods are more like simplified versions of the pipeline model.

5 Evaluation

The evaluation of a dialog agent is crucial for the progress of task-oriented dialog systems. Most evaluation studies follow the PARADISE [56] framework. It estimates the user satisfaction by measuring two types of factors. One is *dialog cost* that measures the cost incurred in the dialog, such as the number of turns.

The other one is *task success* that evaluates whether the system successfully solves the user’s problem. The approaches to evaluate a task-oriented dialog system can be roughly grouped into the following three lines.

5.1 Automatic Evaluation

Automatic evaluation is widely advocated since it is quick, cheap, and objective. A bunch of well-defined automatic metrics have been designed for different components in the system. For dialog state tracking, the evaluation metrics include *slot accuracy* and *joint state accuracy* in general. For policy optimization, *inform rate*, *match rate* and *task success rate* are used. For language generation, metrics such as *BLEU* and *perplexity* are applicable. All the models can be optimized against these metrics via supervised learning. However, each component is trained or evaluated separately in this way. Moreover, it assumes that the model would be fed with the ground truth from upstream modules or last dialog turn in the training process, but this assumption is invalid in real conversation.

5.2 Simulated Evaluation

In addition to training RL-based agents, a user simulator mimicking user behaviors in the task-oriented dialog also enables us to evaluate a trained dialog system. This is because, distinct from open-domain dialog systems, user goals in task-oriented dialog systems are somehow “enumerable” so that it is feasible to exhaustively leverage domain expertise to build a user simulator, which can provide human-alike conversational interaction for simulated evaluation. The metrics used in the simulated evaluation includes task success rate, dialog length, average rewards, etc.

Simulated evaluation has been widely applied in the recently proposed dialog system platforms, such as PyDial [57], Rasa [58] and ConvLab [59,60]. The main advantage of simulated evaluation is that (1) the system can be evaluated in an end-to-end fashion; (2) multi-turn interaction is available during inference; (3) synthetic dialog data can be efficiently generated for evaluation at no cost. Similar to dialog policy optimization, the main challenge of employing simulated evaluation is to build a good user simulator that can mimic real user behaviors as much as possible. Meanwhile, how to evaluate the user simulator also remains an ongoing research direction [61].

5.3 Human Evaluation

Simulated evaluation is efficient to evaluate the system performance with automatic, simulated interactions. Even though having a perfect user simulator, we still require human judgement for more complete evaluation on, e.g. covariate shift between the simulated environment and real conversation [14] and the quality of response generation [55], to assess real user satisfaction. Human evaluation metrics include task success rate, irrelevant turn rate, redundant turn rate, user satisfaction score, etc.

The researchers generally hire human users on the crowd-sourcing platform, and human evaluation can be conducted in the following two ways. One is *indirect evaluation* that asking the annotators to read the simulated dialog between the dialog system and the user simulator, then rate the score [62] or give their preference among different systems [35] according to each metric. The other one is *direct evaluation* that the participants, each of whom is asked to interact with the system to achieve a certain task, give their ratings on the interaction experience. For example, *language understanding* that evaluates whether the dialog agent understands user input, and *response appropriateness* that evaluates whether the dialog response is appropriate during the conversation, are assessed in the DSTC8 competition [63].

6 Task-oriented Dialog Corpora

Since a task-oriented dialog agent requires precise language understanding and heavy domain expertise, a number of corpora with various topics and annotation granularity have been collected to help build

modular or end-to-end task-oriented dialog systems recently. An incomplete survey on these dialog datasets is presented in Table 2¹⁾.

- **DSTC** [64]: This corpus provides a first common testbed and evaluation suite for dialog state tracking. The task of a dialog state tracker here is to assign a score to each observed dialog act. Data discrepancy is explicitly created by choosing training and test data from different dialog systems to avoid performance overestimation.
- **DSTC2** [65]: This corpus introduces some additional features on the dialog state tracking task. The dialog state uses a richer representation, including not only the slot/value attributes of the user goal but also their *search method* representing user intents. It provides the details on the ontology including a list of attributes termed *informable slots* and *requestable slots*. There is a database (DB) of matching entities during the interaction. User goals are permitted to change as well.
- **Dialog bAbI** [66]: This corpus is designed for five goal-oriented dialog tasks. Grounded with an underlying knowledge base (KB), these tasks cover several dialog stages and evaluate if dialog models can learn various abilities such as performing dialog management, querying KBs, interpreting the output of such queries to continue the conversation or dealing with new entities not appearing in dialogs from the training set.
- **CamRest** [55]: This corpus applies a novel crowd-sourcing version of the Wizard-of-Oz (WOZ) paradigm. In order to enable large-scale parallel data collection, users and wizards are asked to contribute just a single turn to each dialog. To ensure coherence and consistency, users and wizards must review all previous turns in that dialog session before they contribute their own turns. The data is relatively clean and comes with good entity extraction methods.
- **Movie-Ticket Booking (MTB)** [67]: This corpus is collected via Amazon Mechanical Turk (AMT) and annotated by domain experts. The main contribution of this corpus is the agenda-based user simulator, which works at the dialog act level for reinforcement learning approaches and simulated evaluation.
- **KVRET** [68]: This corpus covers three different domains in the in-car personal assistant scenario: calendar scheduling, weather information retrieval, and point-of-interest navigation. The dialogs are grounded through KBs, which makes them ideal for building dialog architectures that can reason with world knowledge. The multi-domain nature of the dialogs in the corpus also makes this dataset an apt testbed for model generalization.
- **Frames** [69]: This corpus is developed to study the memory capabilities in goal-oriented dialog systems and how to provide the user with information on the DB. The authors introduce a task called *frame tracking*, which extends state tracking to a setting where several states are tracked simultaneously. A semantic frame is defined by the following four components: constraints, requests, user binary questions, and user comparison requests.
- **M2M** [70]: This corpus is collected via a novel framework combining automation and crowd-sourcing to rapidly bootstrap end-to-end dialog agents in arbitrary domains. It scales to new tasks with just a task schema and an API client from the dialog system developer, but is also customizable to support task-specific interactions. The entire data collection process can finish within a few hours.
- **Air Dialogue** [71]: This corpus defines a goal-driven dialog that is conditioned on a pair of contexts, with the goal of reaching the target state. A mapping is found in this corpus in order to generate the ground-truth state for each dialog context so that it is used as a mechanism to evaluate the generated dialog. It supports three dialog tasks: dialog generation, state tracking, and dialog self-play.

1) *H* stands for human, and *M* for machine in data collection *method*. E.g., H2M means that the data is collected via human users talking to machine-based systems.

Name	Task	Method	Size	Statistics	Labels/Ontologies
DSTC [64]	Bus timetable	H2M	15k	14 turns/dialog	informable slots user/system dialog acts
DSTC2 [65]	Restaurant booking	H2M	3.2k	14.49 turns/dialog 8.54 tokens/turn	informable/requestable slots user/system dialog acts database
bAbI [66]	Restaurant booking	M2M	3k	5 tasks	dialog level database
CamRest [55]	Restaurant booking	H2H	676	7.45 turns/dialog 11.24 tokens/turn	informable/requestable slots user/system dialog acts database
MTB [67]	Movie booking	H2H	280	11 turns/dialog	informable/requestable slots user/system dialog acts database user goals
KVRET [68]	Car assistant	H2H	3k	5.25 turns/dialog 8.02 tokens/turn	informable/requestable slots dialog level database
Frames [69]	Flight/Hotel booking	H2H	1.4k	14.6 turns/dialog 12.6 tokens/turn	semantic frame user/system dialog acts dialog level database
M2M [70]	Restaurant/Movie booking	M2M	3k	9.86 turns/dialog 8.24 tokens/turn	dialog states user/system dialog acts
AirDialogue [71]	Flight booking	M2M/H2H	40k	14.1 turns/dialog 8.17 tokens/turn	dialog states database context pairs
MultiWOZ [72]	Multi-domain booking (Restaurant, Train, etc.)	H2H	10k	7 domains 13.68 turns/dialog 13.18 tokens/turn	dialog states system dialog acts database user goals
CoSQL [73]	Multi-domain booking (College, Music, etc.)	H2H	3k	138 domains 10.36 turns/dialog 11.34 tokens/turn	sql queries user dialog acts database query goals
MultiDoGO [74]	Multi-domain booking (Finance, Media, etc.)	H2H	54k	6 domains 20.06 turns/dialog 12.16 tokens/turn	informable/requestable slots user/system dialog acts
SGD [75]	Multi-domain booking (Movie, Flight, etc.)	M2M	16k	17 domains 20.44 turns/dialog 9.75 tokens/turn	schema-guided dialog states user/system dialog acts services
CrossWOZ [76]	Multi-domain booking (Attraction, Hotel, etc.)	H2H	6k	5 domains 16.9 turns/dialog 16.3 tokens/turn	user/system dialog states user/system dialog acts database user goals

Table 2 Task-Oriented Dialog Corpora

- **MultiWOZ** [72]: This corpus is a large-scale conversational corpus spanning seven domains. It presents the data collection approach, a summary of the data structure, as well as the data

statistics. Each dialog is annotated with a sequence of dialog states and corresponding system dialog acts. Benchmark baselines of belief tracking, natural language generation, and end-to-end response generation have been conducted and reported.

- **CoSQL** [73]: This corpus is gathered for building general-purpose DB querying dialog systems. Each dialog simulates a real-world DB query scenario with a crowd worker as a user exploring the DB and a SQL expert. The expert retrieves answers with SQL, clarifies ambiguous questions, or otherwise informs of unanswerable questions during the dialog. It includes three tasks: SQL-grounded dialog state tracking, natural language response generation, and user dialog act prediction.
- **MultiDoGO** [74]: This corpus is a large-scale dialog dataset annotated with intent types and slot labels. In the devised annotation strategy, the authors distinguish between dialog speech acts for agents vs. customers. The data curation process is controlled with biases to ensure diversity in dialog flows, following variable dialog policies. Multiple levels of annotation granularity are investigated as well.
- **SGD** [75]: This corpus studies the schema-guided approach, an alternative to allow easy integration of new services and APIs. Each service provides a schema listing the supported slots and intents along with their natural language descriptions. These descriptions are used to obtain a semantic representation of these schema elements. The system employs a single unified model containing no domain or service-specific parameters to make predictions conditioned on these schema elements.
- **CrossWOZ** [76]: This is the first large-scale Chinese task-oriented dataset. It focuses on cross-domain user goals that favor inter-domain dependency and encourage natural transition across domains in conversation. It contains rich annotation of dialog states and dialog acts at both user and system sides, along with a user simulator and several benchmark models, which supports a variety of tasks in cross-domain dialog modeling.

7 Discussion and Future Trends

In this paper, we review the recent advancements on task-oriented dialog systems and discuss three critical topics: data efficiency, multi-turn dynamics, and knowledge integration. In addition, we also review some recent progresses on task-oriented dialog evaluation and widely-used corpora. Despite these topics, there are still some interesting and challenging problems. We conclude by discussing some future trends on task-oriented dialog systems:

- **Pre-training Methods for Dialog Systems.** Data scarcity is a critical challenge for building task-oriented dialog systems. On the one hand, collecting sufficient data for a specific domain is time-consuming and expensive. On the other hand, the task-oriented dialog system is a composite NLP task, which is expected to learn syntax, reasoning, decision making, and language generation from not only off-line data but also on-line interaction with users, presenting more requests for fine-grained data annotation and model design. Recently, pre-trained models have shown superior performance on many NLP tasks [77–79]. In this vein, a base model is first pre-trained on large-scale corpora by some unsupervised pre-training tasks, such as masked language model and next sentence prediction. During the pre-training phase, the base model can capture implicit language knowledge, learning from the large-scale corpora. Using such implicit knowledge, the base model can fast adapt to a target task by simply fine-tuning on the data for the target task. This idea can also be applied to task-oriented dialog systems to transfer general natural language knowledge from large-scale corpora to a specific dialog task. Some early studies have shown the possibility of using pre-training models to model task-oriented dialogs [19, 20].
- **Domain Adaptation.** Different from open-domain dialogs, the task-oriented conversations are based upon a well-defined domain ontology, which constrains the agent actions, slot values and

knowledge base for a specific task. Therefore, to accomplish a task, the models of a dialog system are highly dependent on the domain ontology. However, in most existing studies, such ontology knowledge is hard-coded into the model. For example, the dialog act types, slot value vocabularies and even slot-based belief states are all embedded into the model. Such hard-coded ontology embedding raises two problems: (1) Human experts are required to analyze the task and integrate the domain ontology into the model design, which is a time-consuming process. (2) An existing model cannot be easily transferred to another task. Therefore, decoupling the domain ontology and the dialog model to obtain better adaptation performance is a critical issue. One ultimate goal is to achieve zero-shot domain adaptation, which can directly build a dialog system given an ontology without any training data, just like humans do.

- **Robustness.** The robustness of deep learning based models has been a challenging problem since existing neural models are vulnerable to simple input perturbation. As for task-oriented dialog systems, robustness is also a critical issue, which mainly comes from two aspects: (1) On the one hand, the task-oriented dialogs are highly dependent on the domain ontology. Therefore, in many studies, the training data are constrained to only reasonable instances with few noises. However, models trained in such an ad hoc way often fall short in real applications where there are many out-of-domain or out-of-distribution inputs [80], such as previously unseen slot values. A robust dialog system should be able to handle noises and previously unseen inputs after deployment. (2) On the other hand, the decision making of a neural dialog policy model is not controllable, which is trained through off-line imitation learning and on-line RL. The robustness of decision making is rather important for its performance, especially for some special applications which have a low tolerance for mistakes, such as in medical and military areas. Therefore, improving the robustness of neural dialog models is an important issue. One possible approach can be combining robust rule-based methods with neural models, such as Neural Symbolic Machine [81,82], which may make the models not only more robust but also more explainable.

- **End-to-end Modeling.** Compared to pipeline approaches, end-to-end dialog system modeling is gaining more and more attention in recent years. The end-to-end model can be trained more easily without explicit modeling of dialog state and policy. However, existing end-to-end methods still require some intermediate supervision to boost the model performance. For example, in [83], a modular-based end-to-end framework is proposed by combining pre-trained components together and then fine-tuning all the components using RL in an end-to-end fashion, which still requires intermediate supervision such as dialog act and belief state at the pre-training phase. In [40], although a seq-to-seq framework is proposed to avoid component barriers, intermediate output named *belief spans* is still retained for explicit belief state modeling. Therefore, the problem of modeling task-oriented dialog in a fully end-to-end fashion, without intermediate supervision and can seamlessly interact with the knowledge base, is still an open problem.

Acknowledgements This work was supported by the National Science Foundation of China (Grant No.61936010/61876096) and the National Key R&D Program of China (Grant No. 2018YFC0830200).

References

- 1 Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *arXiv preprint arXiv:1905.05709*, 2019.
- 2 David L Poole and Alan K Mackworth. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.
- 3 Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.
- 4 Nikola Mrksić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Multi-domain dialog state tracking using recurrent neural networks. *arXiv preprint arXiv:1506.07190*, 2015.
- 5 Vladimir Ilievski, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. Goal-oriented chatbot dialog management bootstrapping with transfer learning. *arXiv preprint arXiv:1802.00500*, 2018.

- 6 Kaixiang Mo, Yu Zhang, Qiang Yang, and Pascale Fung. Cross-domain dialogue policy transfer via simultaneous speech-act and slot alignment. *arXiv preprint arXiv:1804.07691*, 2018.
- 7 Shuangyin Li Jiajun Li Qiang Yang, Kaixiang Mo, and Yu Zhang. Personalizing a dialogue system with transfer reinforcement learning. *Association for the Advancement of Artificial Intelligence*, 18, 2018.
- 8 Kaixiang Mo, Yu Zhang, Qiang Yang, and Pascale Fung. Fine grained knowledge transfer for personalized task-oriented dialogue systems. *arXiv preprint arXiv:1711.04079*, 2017.
- 9 Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. *arXiv preprint arXiv:1810.13327*, 2018.
- 10 Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.
- 11 Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. *arXiv preprint arXiv:1905.05644*, 2019.
- 12 Kun Qian and Zhou Yu. Domain adaptive dialog generation via meta learning. *arXiv preprint arXiv:1906.03520*, 2019.
- 13 Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, 2019.
- 14 Bing Liu and Ian Lane. Adversarial learning of task-oriented neural dialog models. *arXiv preprint arXiv:1805.11762*, 2018.
- 15 Xinnuo Xu, Yizhe Zhang, Lars Liden, and Sungjin Lee. Unsupervised dialogue spectrum generation for log dialogue ranking. In *Proceedings of the 20th Annual SIGDial Meeting on Discourse and Dialogue*, pages 143–154, 2019.
- 16 Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. On-line active reward learning for policy optimisation in spoken dialogue systems. *arXiv preprint arXiv:1605.07669*, 2016.
- 17 Weiyan Shi, Tiancheng Zhao, and Zhou Yu. Unsupervised dialog structure learning. *arXiv preprint arXiv:1904.03736*, 2019.
- 18 Ryuichi Takanobu, Minlie Huang, Zhongzhou Zhao, Feng-Lin Li, Haiqing Chen, Xiaoyan Zhu, and Liqiang Nie. A weakly supervised method for topic segmentation and labeling in goal-oriented dialogues via reinforcement learning. In *IJCAI*, pages 4403–4410, 2018.
- 19 Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*, 2019.
- 20 Pawel Budzianowski and Ivan Vulic. Hello, it’s gpt-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*, 2019.
- 21 Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 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. Association for Computational Linguistics, 2007.
- 22 Layla El Asri, Jing He, and Kaheer Suleman. A sequence-to-sequence model for user simulation in spoken dialogue systems. *arXiv preprint arXiv:1607.00070*, 2016.
- 23 Paul A Crook and Alex Marin. Sequence to sequence modeling for user simulation in dialog systems. In *INTER-SPEECH*, pages 1706–1710, 2017.
- 24 Florian Kreyssig, Iñigo Casanueva, Pawel Budzianowski, and Milica Gasic. Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 60–69, 2018.
- 25 Izzeddin Gür, Dilek Hakkani-Tür, Gokhan Tür, and Pararth Shah. User modeling for task oriented dialogues. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 900–906. IEEE, 2018.
- 26 Bing Liu and Ian Lane. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE, 2017.
- 27 Baolin Peng, Xiuju Li, Jianfeng Gao, Jingjing Liu, Kam-Fai Wong, and Shang-Yu Su. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*, 2018.
- 28 Shang-Yu Su, Xiuju Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. Discriminative deep dyna-q: Robust planning for dialogue policy learning. *arXiv preprint arXiv:1808.09442*, 2018.
- 29 Yuexin Wu, Xiuju Li, Jingjing Liu, Jianfeng Gao, and Yiming Yang. 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, 2019.
- 30 Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, 2017.
- 31 Denis Yarats and Mike Lewis. Hierarchical text generation and planning for strategic dialogue. *arXiv preprint arXiv:1712.05846*, 2017.
- 32 Zhuoxuan Jiang, Xian-Ling Mao, Ziming Huang, Jie Ma, and Shaochun Li. Towards end-to-end learning for efficient dialogue agent by modeling looking-ahead ability. *arXiv preprint arXiv:1908.05408*, 2019.
- 33 Pei-Hao Su, Milica Gasić, and Steve Young. Reward estimation for dialogue policy optimisation. *Computer Speech & Language*, 51:24–43, 2018.

- 34 Zhaojun Yang, Gina-Anne Levow, and Helen Meng. Predicting user satisfaction in spoken dialog system evaluation with collaborative filtering. *IEEE Journal of Selected Topics in Signal Processing*, 6(8):971–981, 2012.
- 35 Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 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, 2019.
- 36 Mihail Eric and Christopher D Manning. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. *arXiv preprint arXiv:1701.04024*, 2017.
- 37 Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217*, 2018.
- 38 Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. Entity-consistent end-to-end task-oriented dialogue system with kb retriever. *arXiv preprint arXiv:1909.06762*, 2019.
- 39 He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, 2018.
- 40 Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 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, 2018.
- 41 Haoyang Wen, Yijia Liu, Wanxiang Che, Libo Qin, and Ting Liu. Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation. *arXiv preprint arXiv:1806.04441*, 2018.
- 42 Matthew Henderson, Blaise Thomson, and Steve Young. 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, 2014.
- 43 Matthew Henderson, Blaise Thomson, and Steve Young. Deep neural network approach for the dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 467–471, 2013.
- 44 Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016.
- 45 Zheng Zhang, Minlie Huang, Zhongzhou Zhao, Feng Ji, Haiqing Chen, and Xiaoyan Zhu. Memory-augmented dialogue management for task-oriented dialogue systems. *ACM Transactions on Information Systems (TOIS)*, 37(3):1–30, 2019.
- 46 Zheng Zhang, Lizi Liao, Minlie Huang, Xiaoyan Zhu, and Tat-Seng Chua. Neural multimodal belief tracker with adaptive attention for dialogue systems. In *The World Wide Web Conference*, pages 2401–2412, 2019.
- 47 Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*, 2018.
- 48 Puyang Xu and Qi Hu. An end-to-end approach for handling unknown slot values in dialogue state tracking. *arXiv preprint arXiv:1805.01555*, 2018.
- 49 Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700, 2015.
- 50 Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*, 2019.
- 51 Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016.
- 52 Chien-Sheng Wu, Richard Socher, and Caiming Xiong. Global-to-local memory pointer networks for task-oriented dialogue. *arXiv preprint arXiv:1901.04713*, 2019.
- 53 Zehao Lin, Xinjing Huang, Feng Ji, Haiqing Chen, and Ying Zhang. Task-oriented conversation generation using heterogeneous memory networks. *arXiv preprint arXiv:1909.11287*, 2019.
- 54 Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *arXiv preprint arXiv:1702.03274*, 2017.
- 55 Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 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, 2017.
- 56 Marilyn Walker, Diane Litman, Candace A Kamm, and Alicia Abella. Paradise: A framework for evaluating spoken dialogue agents. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 271–280, 1997.
- 57 Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Pawel Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, 2017.
- 58 Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*, 2017.
- 59 Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujuan Li, Minlie Huang, et al. 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, 2019.
- 60 Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. *arXiv preprint arXiv:2002.04793*, 2020.

- 61 Olivier Pietquin and Helen Hastie. A survey on metrics for the evaluation of user simulations. *The knowledge engineering review*, 28(1):59–73, 2013.
- 62 Weiyang Shi, Kun Qian, Xuwei Wang, and Zhou Yu. How to build user simulators to train rl-based dialog systems. 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 1990–2000, 2019.
- 63 Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. 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*, 2020.
- 64 Jason D Williams, Antoine Raux, Deepak Ramachandran, and Alan W Black. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, 2013.
- 65 Matthew Henderson, Blaise Thomson, and Jason D Williams. 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, 2014.
- 66 Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- 67 Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 733–743, 2017.
- 68 Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, 2017.
- 69 Layla El Asri, Hannes Schulz, Shikhar Kr Sarma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, 2017.
- 70 Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 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)*, pages 41–51, 2018.
- 71 Wei Wei, Quoc Le, Andrew Dai, and Jia Li. Airdialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854, 2018.
- 72 Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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, 2018.
- 73 Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. 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 1962–1979, 2019.
- 74 Denis Peskov, Nancy Clarke, Jason Krone, Brigi Fodor, Yi Zhang, Adel Youssef, and Mona Diab. 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)*, pages 4518–4528, 2019.
- 75 Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2020.
- 76 Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. Crosswoz: A large-scale chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 2020.
- 77 Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- 78 Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf, 2018.
- 79 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- 80 Yinhe Zheng, Guanyi Chen, and Minlie Huang. Out-of-domain detection for natural language understanding in dialog systems. *arXiv preprint arXiv:1909.03862*, 2019.
- 81 Chen Liang, Jonathan Berant, Quoc Le, Kenneth D Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *arXiv preprint arXiv:1611.00020*, 2016.
- 82 Marwin HS Segler and Mark P Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry—A European Journal*, 23(25):5966–5971, 2017.
- 83 Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.