SCIENCE CHINA
Technological Sciences

doi: 10.1007/s11432-016-0037-0

• REVIEW •

Recent Advances and Challenges in Task-oriented Dialog Systems

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, 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*

Received March 20, 2020; accepted April 20, 2020; published online May 20, 2020

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 task-oriented dialog systems. We also discuss three critical topics for task-oriented dialog systems: (1) improving data efficiency to facilitate dialog 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. Besides, we review the recent progresses in dialog evaluation and some widely-used corpora. We believe that this survey, though incomplete, can shed a light on future research in task-oriented dialog systems.

Task-oriented Dialog Systems, Natural Language Understanding, Dialog Policy, Dialog State Tracking, Natural Language Generation

Citation: Zhang Z, Takanobu R, Zhu Q, Huang M, Zhu X. Recent Advances and Challenges in Task-oriented Dialog Systems. Sci China Tech Sci, doi: 10.1007/s11432-016-0037-0

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 [2]. Typically, task-oriented dialog systems are built on top of a structured ontology, which defines the domain knowledge of the tasks.

Existing studies on task-oriented dialog systems can be broadly classified into two categories: pipeline and end-to-end methods. In the pipeline methods, the entire system is divided into several modules, including natural language understanding (NLU), dialog state tracking (DST), dialog policy (Policy) and natural language generation (NLG). There are also some other combination modes, such as word-level DST [3, 4] (coupling NLU and DST) and word-level policy [5, 6] (coupling Policy and NLG). While end-to-end methods build the system using a single model, which directly takes a natural language context as input and outputs a natural language response as well.

Building pipeline system often requires large-scale labeled dialog data to train each component. The modular structure makes the system more interpretable and stable than end-to-end counterparts. Therefore, most real-world commercial

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

systems are built in this manner. End-to-end systems require less annotations, making it more easily to build. However, the end-to-end structure makes it a black box, which is more uncontrollable [7].

To make a clear review of existing studies, we build a taxonomy for task-oriented dialog systems. As illustrated in Figure 1, for each individual component in pipeline and end-to-end methods, we list several key issues within which typical works are presented.

In pipeline methods, recent studies 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 are less interweaved to the other tasks 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 are 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 [8] and Go [9], 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 will be discussed in detail shortly:

- **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 are often hard to collect and expensive to annotate. Therefore, the problem of low-resource learning is one of the major challenges.

- **Multi-turn Dynamics** Compared to open-domain dialog, the core feature of task-oriented 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 for better policy learning, including model-based planning, reward estimation and end-to-end policy learning.

- **Ontology 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 bypass modular models which requires fine-grained annotation and domain expertise. However, this simplification makes it hard to construct a query since there is no explicit state representation.

This paper is structured as follows: In Section 2, we introduce the recent advances of each component in pipeline methods and end-to-end approaches. In Section 3, we discuss recent work on task-oriented dialog evaluation, including automatic, simulated, and human evaluation methods. In Section 4, we survey some widely-used corpus for task-oriented dialog. In Section 5, we review the approaches proposed to address the above three challenges. Finally in Section 6, we conclude the paper and discuss future research trends.

2 Modules and Approaches

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 2. 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 some other kinds of configurations. Recently, there are studies that merge some of the typical components, such as word-level DST and word-level policy, resulting in various pipeline configurations [3–6].

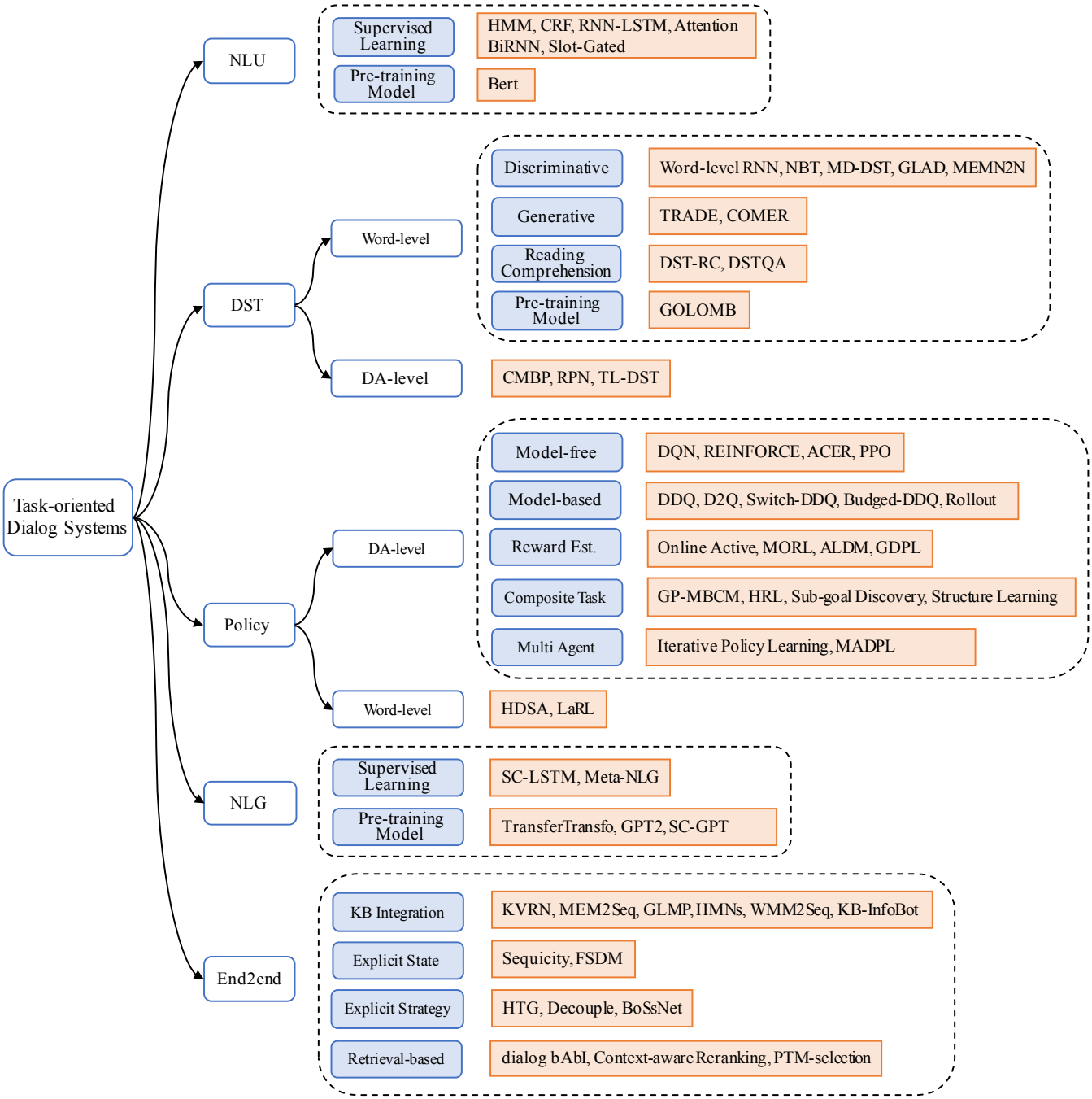

Figure 1 Taxonomy of task-oriented dialog systems.

In end-to-end approaches, 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.

2.1 Natural Language Understanding

Given a user utterance, the natural language understanding (NLU) component maps the utterance 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 classification of an utterance, such as *Query* and *Inform*, which indicates the function of the utterance. Slot-value pairs are the task-specific semantic elements that are mentioned in the utterance. Note that both intent type and slot-value pairs are task-specific, which are related to the ontology and can be used to query knowledge base.

Based on the dialog act structure, the task of NLU can be

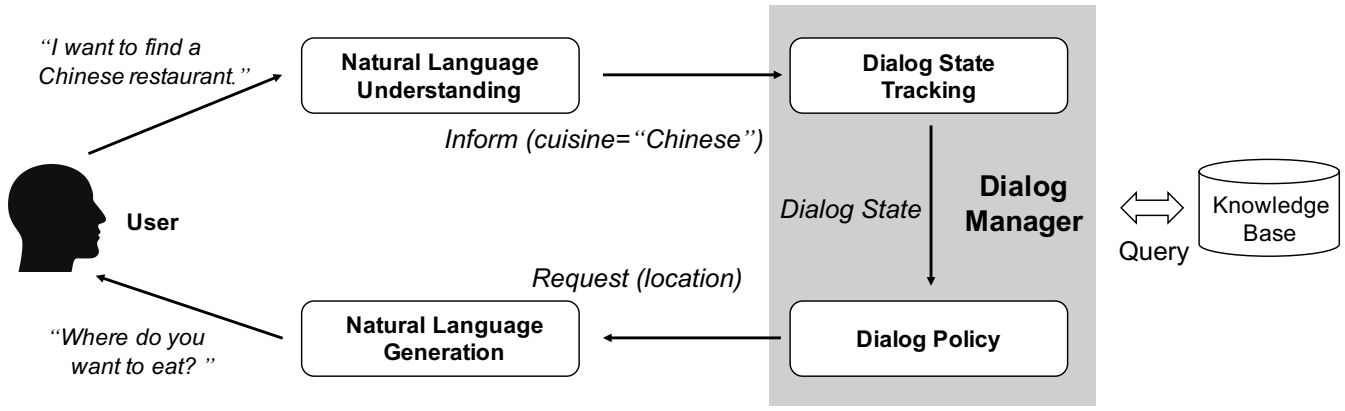

Figure 2 General framework of a pipeline task-oriented dialog system.

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, while the slot-value recognition task is often viewed as a sequence labeling problem:

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

$$p_{s-v}(y_1, y_2, ..., y_n|x_1, x_2, ..., x_n) \quad (2)$$

where the d indicates intent class and y_1 to y_n are the labels of each token in the utterance $[x_1, x_2, ..., x_n]$ in which x_i is a token and n means the number of tokens.

Due to the strong ability of sequence modeling, RNN and its variants have been widely used in intent detection and slot-value extraction [10–12]. These models used the hidden state of each token to predict the corresponding label y_i and used the final hidden state to recognize the sentence intent d. Other neural network structures such as recursive neural network [13] and CNN [14] have also been explored. Conditional random field, which is frequently used by traditional sequence tagging models, was combined with RNN [15] and CNN [14] to improve the performance. Recently pre-training model BERT [16] has been another popular choice [17, 18].

There are also some models strengthening the connection between intent classification and slot tagging. [19] used an intent gate to direct the slot tagging process, while [20] applied attention mechanism to allow the interaction between word and sentence representations.

2.2 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. Early works assumed some fixed sets of dialog state, and modeled the state transition during interaction as a Markov Decision Process (MDP). POMDP further assumes that the observation is partially observable, which makes it more robust in sophisticated situations [21–24]. Most recent works adopted belief state for dialog state representation, in which the state is composed of slot-value pairs that represent the user's goal. Therefore, this problem can be formulated as a multi-task classification task [3, 25–27]:

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

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 run time. Besides, there are also some works formulating the DST task as a reading comprehension task [28, 29].

In more recent methods, slots can be divided into two types: free-form and fixed vocabulary [30]. The former type does not assume a fixed vocabulary for the slot, which means the model cannot predict the values by classification. For free-form slot, one could generate the value directly [4, 31] or predict the span of the value in the utterance [28, 32]. In generative methods, they often use a decoder to generate the value of a slot word by word from a large vocabulary. However, for rare words, this method can also fail since the vocabulary is limited. While for span-based methods, the model assumes that the value are shown in the context, and predicts the start and end position of that span.

2.3 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 3, 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.

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

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

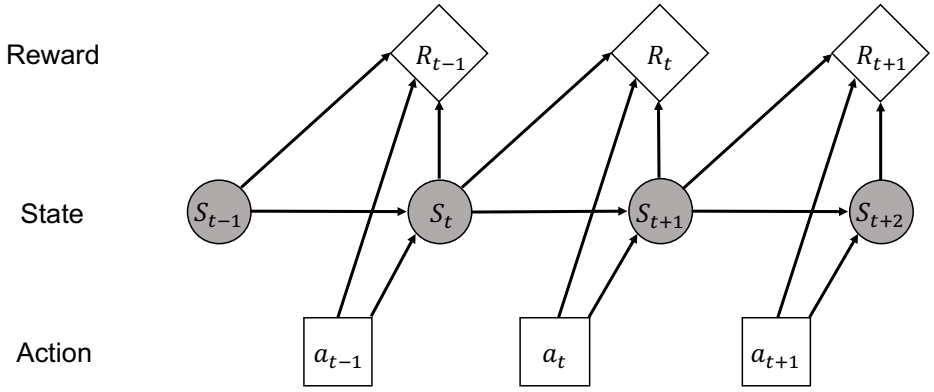

Figure 3 Framework of Markov Decision Process [33]. At time t, the system takes an action a_t, receiving a reward R_t and transferring to a new state S_{t+1}.

A typical approach is to first train the dialog policy off-line through supervised learning or imitation learning based on a 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.

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.

Model-free RL methods dominated the early studies of neural dialog policy by learning through interaction with real users, such as DQN and Policy Gradient methods [34–36]. For complex multi-domain dialogs, hierarchical RL models are introduced to first decide which is the domain of current turn and then select an action of that domain [37]. Training a RL policy model requires a large amount of interactions . One common solution is to use user simulators [24, 38, 39], which is another dialog system acting like a human user to provide training and evaluating environment. 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 poorly in real human conversation. To alleviate these problems, model-based RL methods are proposed to model the environment, enabling planning for dialog policy learning [40–42]. In model-based RL approaches, the environment is modeled to 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. Some other works jointly train a system policy and a user policy simultaneously [43, 44].

2.4 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 [45]. The task takes dialog act as input and generates the natural language response. 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. Another problem is how to build a robust NLG with limited training data. Peng et al. [46] proposed SC-GPT by first pre-training GPT with large-scale NLG corpus collected from existing publicly available dialog datasets, and then fine-tuning the model on target NLG tasks with few training instance.

2.5 End-to-end Methods

Generally speaking, the components in a pipeline system are optimized separately. This modularized structure leads to complex model design, and the performance of each individual component does not necessarily translate to the advance of the whole system [7]. 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 mod-

ular design, as shown in Figure 4. Most of these methods utilized sequence to sequence models as the infrastructural framework, which is end-to-end differentiable and can be optimized by gradient-based methods [47].

In most existing end-to-end approaches, the models are trained to maximize the prediction probability of response in the collected data. Wen et al. [48] proposed a modularized end-to-end model in which each component is modeled using neural networks, which makes the model end-to-end differentiable. Bordes et al. [49] formalized the task-oriented dialog as a reading comprehension task by regarding the dialog history as context, user utterance as the question, and system response as the answer. In this work, they utilized end-to-end memory networks for multi-turn inference. Madotto et al. [50] took a similar approach and further feed the knowledge base information into the memory networks. In [51] a new memory network structure named key-value memory networks is introduced to extract relevant information from KB through key-value retrieval. Lei et al. [52] proposed a two-step seq2seq generation model which bypassed the structured dialog act representation, and only retain the dialog state representation. In their method, the model first encodes the dialog history and then generates a dialog state using LSTM and CopyNet. Given the state, the model then generates the final natural language response.

One major drawback of the above methods is that they often require large amounts of training data, which is expensive to obtain. Furthermore, they cannot fully explore the state-action space since the model only observes examples in the data. Therefore, reinforcement learning methods are introduced to mitigate these issues [52–57]. In [53], there is an end-to-end model that takes the natural language utterance as input and generates system dialog act as a response. In this method, there is no explicit state representation. Instead, they used LSTM to encode the dialog history into a state vector and then use DQN to select an action. Williams et al. [54] proposed LSTM-based hybrid code networks (HCN), which supports self-defined software.

3 Evaluation

The evaluation of a dialog agent is crucial for the progress of task-oriented dialog systems. Most evaluation studies follow the PARADISE [58] framework. It estimates the user satisfaction from two aspects. 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.

3.1 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 language understanding, *slot F1* and *intent accuracy* are used. 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. Detailed definition of these metrics can be found in [59]. 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.

3.2 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-like 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 [60] and ConvLab [61,62]. 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 [63].

3.3 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 [64] and the quality of response generation [48], to

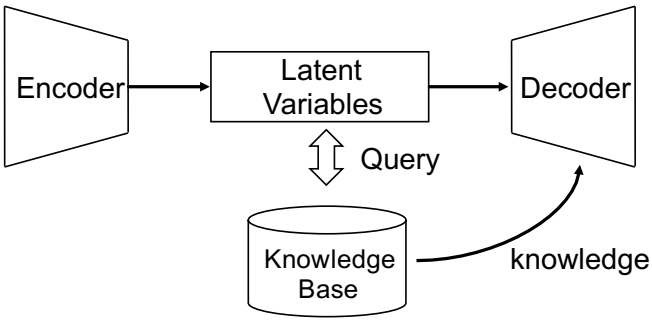

Figure 4 Framework of end-to-end dialog systems. It first encodes natural language context to obtain some latent variables, which can be used for KB query. Then based on the latent variables and query results, the decoder generates a natural language response.

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 crowdsourcing 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 [39] or give their preference among different systems [65] according to each metric. The other one is *direct evaluation* that the participants are asked to interact with the system to complete 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 [66].

4 Corpora

A number of corpora with various domains and annotation granularity have been collected to facilitate the research on task-oriented dialog systems. Some datasets contain single-domain conversations [48, 51, 67, 68]. With the increasing demands to handle various tasks in real-world applications, some large-scale multi-domain corpora [69–71] have been collected recently. These datasets have higher language variation and task complexity. While most datasets are in English, Zhu et al. [72] propose the first large-scale Chinese task-oriented dataset with rich annotations to facilitate the research of Chinese and cross-lingual dialog modeling. An incomplete survey on these dialog datasets is presented in Table 2.

With respect to data annotation, the DSTC corpus [73] provides the first common testbed and evaluation suite for dialog state tracking. DSTC2 [67] contains additional details on the ontology including a list of attributes termed *informable slots* and *requestable slots* for NLU tasks. User goals and a database of matching entities during the conversation are provided in some corpora [48, 69, 74] as well, which can be utilized for modeling multi-turn interactions. It is worth noting that, the schema of dialog state annotation is often different across these datasets. For example, *search methods* representing user intents are included in DSTC2, and a schema listing the supported slots and intents along with their natural language descriptions is provided in SGD [75].

There are mainly three modes in data collection. The first one is human-to-machine (H2M) where the data is collected via human users talking to a deployed machine-based system. The second mode is machine-to-machine (M2M) where two systems play the user and system roles respectively and interact to each other to generate conversations. Shah et al. [76] bootstrap the data collection process by first generating dialog templates at dialog act level using the M2M mode, and then converting these templates to natural language using crowd sourcing. The advantage of this method lies in that semantic annotations can be obtained automatically, which is thus cost-effective and error-resistant since translating templates to sentences is relatively simple for crowd-sourcing workers. However, the task complexity and language diversity is often restricted because the dialog simulation is performed using heuristic rules. The third mode is human-to-human (H2H), most following the Wizard-of-Oz (WoZ) paradigm [77] which collects real conversations between two crowd-sourcing workers who play a role of an agent (system) and a client (user) respectively. Each worker is given a task description about their goals and how they should act before the dialog is launched. While such a framework yields natural and diverse dialogs, it raises the difficulty of data annotation, especially when the annotation scheme is fine-grained.

5 Challenges

5.1 Data Efficiency

Different from the research in open-domain dialog systems, data-driven approaches for task-oriented dialog systems often

Name	Task	Method	Size	Statistics	Labels/Ontologies
DSTC[73]	Bus timetable	H2M	15K	14 turns/dialog	dialog states user/system dialog acts
DSTC2[67]	Restaurant booking	H2M	3.2K	14.49 turns/dialog 8.54 tokens/turn	dialog states user/system dialog acts database
bAbI[49]	Restaurant booking	M2M	3K	5 tasks	dialog level database
CamRest[48]	Restaurant booking	H2H	676	7.4 turns/dialog	dialog states user/system dialog acts database
WOZ[3]	Restaurant booking	H2H	1.2K	7.45 turns/dialog 11.24 tokens/turn	dialog states user/system dialog acts database
KVReT[51]	Car assistant	H2H	3K	5.25 turns/dialog 8.02 tokens/turn	dialog states dialog level database
Frames[68]	Flight/Hotel booking	H2H	1.4K	14.6 turns/dialog 12.6 tokens/turn	semantic frame user/system dialog acts dialog level database
SimD[78]	Restaurant/Movie booking	M2M	3K	9.86 turns/dialog 8.24 tokens/turn	dialog states user/system dialog acts
AirD[79]	Flight booking	M2M/H2H	40K	14.1 turns/dialog 8.17 tokens/turn	dialog states database context pairs
MultiWOZ[69]	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
MDC[74]	Movie/Restaurant/Taxi booking	H2H	10K	7.5 turns/dialog	user/system dialog acts database user goals
CoSQL[80]	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
Taskmaster[71]	Multi-domain booking (Repair, Drinks, etc.)	H2H/self	13K	6 domains 22.9 turns/dialog 8.1 tokens/turn	API calls and arguments
SGD[75]	Multi-domain booking (Movie, Flight, etc.)	M2M	23K	17 domains 20.44 turns/dialog 9.75 tokens/turn	schema-guided dialog states user/system dialog acts services
CrossWOZ[72]	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.

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 the open-domain setting due to its task-specific nature, and (2) annotating fine-grained labels requires a large amount of human resources which is very expensive and time-consuming. Therefore, we have to face the problem of improving the data efficiency of building task-oriented dialog systems, particularly in low-resource settings.

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 adapt 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.

5.1.1 Transfer Learning

One major assumption of machine learning is that the training and test data have the same distribution. However, in many real-world scenarios, this does not hold when we have only limited data in the target task but sufficient data in another task, with different data distributions. Transfer learning is thus proposed to mitigate this problem by transferring knowledge from a source task to a 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. [26] 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 [81], Ilievski et al. proposed to directly transfer the parameters of shared slots from the source domain model to initialize the target model. Chen et al. [82] proposed to model dialog agent using several slot-dependent agents and a slot-independent agent to track the private and public slots across different domains. In [83, 84], the parameters of DST models are shared across domains and is independent of pre-defined value sets. Therefore, the model is able to transfer to previously unseen domains. Wu et al. [4] further decoupled the domain and slot from the model parameters by taking domain and slot names as inputs to the DST model.

For transferring across disjoint tasks, Mo et al. [85] proposed to transfer the dialog policy model between domains by learning act and state transfer functions where there are no shared slots, which directly maps from the source feature space to the target space.

For personalized knowledge transfer, in [86], 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. [87] 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 proposed. In [88], 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 [89]. 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 [90, 91]. Madotto et al. [92] 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.

Besides the above methods which transfer knowledge from a source model, there are also some works improving data efficiency by directly endowing the model or algorithms with prior knowledge to decrease data usage. For example, improved RL methods including ACER [93] and BBQ-Networks [36] are proposed to enhance sample efficiency. In [94], the action selection process is decomposed into master action and primitive action selection, and the two actions are designed according to the domain ontology.

5.1.2 Unsupervised Methods

A crucial issue in dialog policy learning is to estimate reward signal, which is hard to be obtained in real-world applications. Therefore, building a reward estimation model is necessary for dialog policy learning, particularly during RL training. 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. [64] first used GAN to learn a binary reward function by discriminating simulated from real user dialogs. Xu et al. [95] extended this idea for detecting dialog failure by using the predicted reward as an indicator of failure. Su et al. [96] 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 failure cases. In their experiment, the requirement for human intervention dramatically decreases with the reduction in the uncertainty of reward estimation, which remarkably relieve manual annotation.

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. [97] 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 data 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. Takanobu et al. [98] proposed an RL method for topic segmentation and labeling in task-oriented dialog systems, 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). Wolf et al. [99] 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. Budzianowski et al. [100] 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.

5.1.3 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) [24], which is commonly used in building task-oriented dialog systems. 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 linguistic variations of human dialogs, which may lead to suboptimal performance in real applications.

Recently, building user simulators in a data-driven fashion is proposed to alleviate the above issues. Asri et al. [101] proposed a dialog act level seq2seq user simulation model that takes into account the dialog context. Crook et al. [102] presented another seq2seq model which takes as input natural language contexts and outputs natural language responses. Kreyssig et al. [103] introduced a neural user simulator (NUS), which mimics the user behavior of the corpus and generates word-level user responses. Gur et al. [104] proposed a hierarchical seq2seq user simulator (HUS) that first encodes the user goal and system turns, and then generates user dialog act. To generate more diverse user acts, they extended 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. [57] 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 [40–42], 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.

The user simulators in the above methods are trained based on the human-agent dialog data. In addition to this, the human can also assist dialog policy learning by providing human demonstrations. Since the human guidance is expensive, Chang et al. [105] compared various teaching schemes answering the question how and when to teach, to use the teaching budget more economically. Chen et al. [106] further proposed companion learning (CL) framework, which integrates rule-based policy and RL-based policy. Since the rule teacher is not as good as a human teacher, an uncertainty estimation is introduced to control the timing of consultation and learning.

5.2 Multi-turn Dynamics

Compared to open-domain dialog systems, one major feature of task-oriented dialog systems is the emphasis on multi-turn state-action dynamics, which is mainly related to dialog management (DST and Policy). In open-domain dialog systems, the research focuses more on generating reasonable, consistent, and inter-personal responses to maximize user engagement [1]. While for task-oriented dialog systems, although the above issues are still important, the completion of a specific task has been viewed as more critical. Therefore, the research on dialog management, which is responsible for tracking the dialog state and flow of the conversation, acts as the pillar of a dialog system.

Human conversation can be broadly formulated as a Markov Decision Process (MDP): at each time step, the system transits from a certain state s to a new state s' by taking an action a. Therefore, reinforcement learning (RL) is often applied to solve such an MDP problem in the dialog systems. Recent studies on the dialog management of task-oriented dialog systems are mainly focused on the following topics: (1) generative DST with value decoder for free-form slots, (2) dialog planning for better sample efficiency in policy learning, and (3) user goal estimation for predicting task success and user satisfaction.

5.2.1 Generative DST

Dialog state tracker plays a central role in task-oriented dialog systems by tracking of a structured dialog state representation at each turn. Most recent DST studies applied a word-level structure by taking natural language as input without NLU, which may avoid the errors propagated from the NLU component. In early neural DST methods, *belief state* is widely adopted for dialog state representation [25], which maintains a distribution over all possible values for each slot. Therefore, early methods commonly formulated DST as a classification task [3, 26, 107–109]. Matthew et al. [107] 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. [3] 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. [110] 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. [52] 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 [111] 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. [112] proposed to use pointer network [113] to extract unknown slot values, which showed superior performance over discriminative DST methods. A more practical way is to use both extractive and discriminative methods to handle different type of slots [30]. For the free-form slots, such as *hotel name* and *departure date*, their value should be extracted from the utterance. While for those fixed-vocabulary slots like *hotel star* and *room category*, it is better to predict their value using a classifier.

Recently, some multi-domain datasets are proposed to promote the research in this direction [69, 75]. Compared to single-domain tasks, the DST in multi-domain scenario has to predict the domain of slot values. Wu et al. proposed TRADE [4], a transferable multi-domain DST using seq2seq model with CopyNet [111] to predict values. The parameters are shared across domains, enabling zero-shot DST for unseen domains. COMER [31] further decreases the computation complexity of value decoding by first deciding the domain and slot, and then decoding the value. In the decoding of the above methods, they first input the domain and slot names to the decoder, and then decode the value. If we take the domain and slot names as a form of "question", then the model can be regarded as a question answering model by taking the previous turns as context, domain-slot names as question and the value as answer. DSTQA [32] added more elements into the "question" in addition to the names, such as the description text of domain and slots, values of fixed-vocabulary slots. They also encoded the intermediate dialog state graph using GNN to alleviate value decoding. In [114], Chen et al. proposed to use graph attention neural networks to model the relations across slots.

5.2.2 Dialog Planning

Model-free RL methods dominated the early studies of neural dialog policy by learning through interaction with real users [34, 54, 55, 115]. It is data-hungry, requiring a large amount of interactions to train a policy model effectively. One common solution is to use user simulators [24, 38]. 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 poorly in real human conversation [39].

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 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 [116]. Peng et al. [40] proposed the Deep Dyna-Q (DDQ) framework, which first integrates model-based planning for task-oriented dialog systems.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 [41,42] 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. The simulated future steps can provide extra hints to facilitate decision making. 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. Some works [117,118] introduced *dialog rollout* planning into negotiation dialogs, 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 will be taken. Instead of completing the dialogs and obtaining explicit rewards, Jiang et al. [119] proposed to look only several limited steps ahead and use those steps as additional features for the policy model to alleviate decision making.

5.2.3 User Goal Estimation

In RL-based dialog models, the user's goal is crucial for policy learning. Reward signal is an indirect reflect of the user's goal since it gives the user's satisfaction of a dialog. 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 [120]. 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 judged 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 [121]. 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 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 [96]. 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 estimating 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 user dialogs [64,65,95].

5.3 Ontology Integration

One major issue in task-oriented dialog systems is to integrate the ontology of dialog into the dialog model, including domain schema and knowledge base. In most previous methods, the domain schema is pre-defined and highly dependent on the corpus they use, e.g., the slots of restaurant domain contain address area, cuisine type, price range, etc.. As querying the database and retrieving the results are essential for a task-oriented dialog system to make decisions and produce appropriate responses, there are also many efforts to integrate external database or API calls recently.

However, ontology integration for task-oriented dialog models becomes more challenging, because of the large scale of task domains. Though the pre-defined ontology can be considered into model design, these approaches are coupled with domain schema and can not be easily transferred to a new task. While increasing end-to-end models are proposed to alleviate the schema integration problem, 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) dialog task schema integration and (2) knowledge base integration in task-oriented dialog models.

5.3.1 Schema Integration

Integrating the schema into a dialog model is critical for task-oriented dialog, since the value prediction of NLU and DST, and the action selection in Policy are highly dependent on the domain schema. Early methods for NLU use classification for intent detection and sequence labeling for slot-value recognition. Therefore, the schema integration are mainly reflected in the model output layer design, e.g., one class for each intent. Early DST methods utilized a similar way by giving a value probability distribution on the value vocabulary for each slot (also known as belief state). For NLG methods, the inputs are often structured dialog acts, and the encoder input structure is highly dependent on the representation structure.

The above schema integration methods basically couple the schema and model design together, yield poor scalability and domain generalization. Recently, there are many methods trying to untie the domain scheme and model design. Convlab [61] provides additional user dialog act annotation in the MultiWOZ [69] dataset to enable developers to apply NLU models in multi-domain, multi-intent settings. While most DST makes assumption that a slot in a belief state can only be mapped to a single value within a single turn, COMER [31] extends the representation of dialog states with priority operator that considers the user's preference on slot values. Other works [32, 122] use question answering methods for DST by taking domain-slot descriptions as question. The values are regarded as answers, which are predicted by either extraction or generation based methods. In such methods, the model design is decoupled from the domain schema, and the schema of domain is represented by a natural language text, which makes it easy to transfer to new domain. For NLG task, Peng et al. [46] proposed SC-GPT, which treats the structured dialog act as a sequence of tokens, and feeds the sequence to the generation model. By pre-training on large-scale da-response pairs, the model is able to capture the semantic structure of the sequence-based dialog act representation. When extending to a new domain, only a small amount of training instances (50) are required to achieve satisfactory performance. ZSDG [123] learns a cross-domain embedding space that models the semantics of dialog responses so that it can instantly generalize to new situations with minimal data. Each service (domain) in SGD [75] 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, making models applicable in a zero-shot setting.

5.3.2 Knowledge Base Integration

It is critical for a task-oriented dialog system to query the external knowledge base to get user's inquired information. Early models or pipeline systems retrieved entries from the knowledge base by issuing a query based on the current dialog state during conversational interaction, which requires some manual effort. Training an end-to-end dialog system without intermediate supervision will be more appealing due to the growing task complexities in task-oriented scenarios. However, 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. [124] 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 [50, 125]. 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. [126] 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 [127], 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. [48] 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.

6 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 [16, 128, 129]. 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 [46, 99, 100, 130, 131].

- **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 neural 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 [132], 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 is to combine robust rule-based methods with neural models, such as Neural Symbolic Machine [133, 134], 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 [48], 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 [52], 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.

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).

1 Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems*, 2020.

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

3 Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, 2017.

4 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.

5 Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. *arXiv preprint arXiv:1902.08858*, 2019.

6 Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866*, 2019.

7 Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374, 2018.

8 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

9 David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

10 Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. In *INTERSPEECH*, 2013.

11 Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. Spoken language understanding using long short-term memory neural networks. In *IEEE Spoken Language Technology Workshop*, 2014.

12 Dilek Hakkani-Tür, Gökhan Tür, Asli Çelikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In *INTERSPEECH*, 2016.

13 Daniel Guo, Gökhan Tür, Wen-tau Yih, and Geoffrey Zweig. Joint semantic utterance classification and slot filling with recursive neural networks. In *IEEE Spoken Language Technology Workshop*, 2014.

14 Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.

15 Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. Recurrent conditional random field for language understanding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

16 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 2019.

17 Qian Chen, Zhu Zhuo, and Wen Wang. BERT for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019.

18 Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. Multi-lingual intent detection and slot filling in a joint bert-based model. *arXiv preprint arXiv:1907.02884*, 2019.

19 Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. Slot-gated modeling for joint slot filling and intent prediction. In *NAACL-HLT*, 2018.

20 Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*, 2016.

21 Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

22 Steve Young. Using pomdps for dialog management. In *2006 IEEE Spoken Language Technology Workshop*, pages 8–13. IEEE, 2006.

23 Jason D Williams and Steve Young. Scaling up pomdps for dialog management: The"summary pomdp"method. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 177–182. IEEE, 2005.

24 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.

25 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.

26 Nikola Mrkšić, 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.

27 Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. Sumbt: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5478–5483, 2019.

28 Shuyang Gao, Abhishek Sethi, Sanchit Aggarwal, Tagyoung Chung, and Dilek Hakkani-Tur. Dialog state tracking: A neural reading comprehension approach. *arXiv preprint arXiv:1908.01946*, 2019.

29 Julien Perez. Machine reading method for dialog state tracking, January 21 2020. US Patent 10,540,967.

30 Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*, 2019.

31 Liliang Ren, Jianmo Ni, and Julian McAuley. Scalable and accurate dialogue state tracking via hierarchical sequence generation. *arXiv preprint arXiv:1909.00754*, 2019.

32 Li Zhou and Kevin Small. Multi-domain dialogue state tracking

as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*, 2019.

33 David L Poole and Alan K Mackworth. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

34 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

35 Li Zhou, Kevin Small, Oleg Rokhlenko, and Charles Elkan. End-to-end offline goal-oriented dialog policy learning via policy gradient. *arXiv preprint arXiv:1712.02838*, 2017.

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

37 Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. *arXiv preprint arXiv:1704.03084*, 2017.

38 Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*, 2016.

39 Weiyan Shi, Kun Qian, Xuewei Wang, and Zhou Yu. How to build user simulators to train rl-based dialog systems. *arXiv preprint arXiv:1909.01388*, 2019.

40 Baolin Peng, Xiujun 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.

41 Yuexin Wu, Xiujun 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.

42 Shang-Yu Su, Xiujun 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.

43 Alexandros Papangelis, Yi-Chia Wang, Piero Molino, and Gokhan Tur. Collaborative multi-agent dialogue model training via reinforcement learning. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 92–102, 2019.

44 Ryuichi Takanobu, Runze Liang, and Minlie Huang. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. *arXiv preprint arXiv:2004.03809*, 2020.

45 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.

46 Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*, 2020.

47 Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

48 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.

49 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.

50 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.

51 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.

52 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.

53 Tiancheng Zhao and Maxine Eskenazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. *arXiv preprint arXiv:1606.02560*, 2016.

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 Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777*, 2016.

56 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.

57 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.

58 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.

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

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

61 Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujun 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.

62 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.

63 Olivier Pietquin and Helen Hastie. A survey on metrics for the evaluation of user simulations. *The knowledge engineering review*, 28(1):59–73, 2013.

64 Bing Liu and Ian Lane. Adversarial learning of task-oriented neural dialog models. *arXiv preprint arXiv:1805.11762*, 2018.

65 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.

66 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.

67 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.

68 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.

69 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.

70 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.

71 Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. Taskmaster-1: Toward a realistic and diverse dialog dataset. 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 4506–4517, 2019.

72 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.

73 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.

74 Xiujun Li, Yu Wang, Siqi Sun, Sarah Panda, Jingjing Liu, and Jianfeng Gao. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*, 2018.

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 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.

77 John F Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41, 1984.

78 Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*, 2018.

79 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.

80 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.

81 Vladimir Ilievski, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. Goal-oriented chatbot dialog management bootstrapping with transfer learning. *arXiv preprint arXiv:1802.00500*, 2018.

82 Lu Chen, Cheng Chang, Zhi Chen, Bowen Tan, Milica Gašić, and Kai Yu. Policy adaptation for deep reinforcement learning-based dialogue management. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6074–6078. IEEE, 2018.

83 Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE, 2017.

84 Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. Towards universal dialogue state tracking. *arXiv preprint arXiv:1810.09587*, 2018.

85 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.

86 Kaixiang Mo, Yu Zhang, Shuangyin Li, Jiajun Li, and Qiang Yang. Personalizing a dialogue system with transfer reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

87 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.

88 Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. *arXiv preprint arXiv:1810.13327*, 2018.

89 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.

90 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.

91 Kun Qian and Zhou Yu. Domain adaptive dialog generation via meta learning. *arXiv preprint arXiv:1906.03520*, 2019.

92 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.

93 Gellért Weisz, Paweł Budzianowski, Pei-Hao Su, and Milica Gašić. Sample efficient deep reinforcement learning for dialogue systems with large action spaces. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11):2083–2097, 2018.

94 Inigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Stefan Ultes, Lina Rojas-Barahona, Bo-Hsiang Tseng, and Milica Gašić. Feudal reinforcement learning for dialogue management in large domains. *arXiv preprint arXiv:1803.03232*, 2018.

95 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.

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

97 Weiyan Shi, Tiancheng Zhao, and Zhou Yu. Unsupervised dialog

structure learning. *arXiv preprint arXiv:1904.03736*, 2019.

98 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.

99 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.

100 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.

101 Layla El Asri, Jing He, and Kaheer Suleman. A sequence-to-sequence model for user simulation in spoken dialogue systems. *Interspeech 2016*, pages 1151–1155, 2016.

102 Paul A Crook and Alex Marin. Sequence to sequence modeling for user simulation in dialog systems. In *INTERSPEECH*, pages 1706–1710, 2017.

103 Florian Kreyssig, Iñigo Casanueva, Paweł 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.

104 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.

105 Cheng Chang, Runzhe Yang, Lu Chen, Xiang Zhou, and Kai Yu. Affordable on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2200–2209, 2017.

106 Lu Chen, Xiang Zhou, Cheng Chang, Runzhe Yang, and Kai Yu. Agent-aware dropout dqn for safe and efficient on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2454–2464, 2017.

107 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.

108 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.

109 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.

110 Victor Zhong, Caiming Xiong, and Richard Socher. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*, 2018.

111 Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016.

112 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.

113 Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700, 2015.

114 Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *AAAI*, pages 7521–7528, 2020.

115 Heriberto Cuayáhuitl. Simpleds: A simple deep reinforcement learning dialogue system. In *Dialogues with social robots*, pages 109–118. Springer, 2017.

116 Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

117 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.

118 Denis Yarats and Mike Lewis. Hierarchical text generation and planning for strategic dialogue. *arXiv preprint arXiv:1712.05846*, 2017.

119 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.

120 Pei-Hao Su, Milica Gašić, and Steve Young. Reward estimation for dialogue policy optimisation. *Computer Speech & Language*, 51:24–43, 2018.

121 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.

122 Pavel Gulyaev, Eugenia Elistratova, Vasily Konovalov, Yuri Kuratov, Leonid Pugachev, and Mikhail Burtsev. Goal-oriented multi-task bert-based dialogue state tracker. *arXiv preprint arXiv:2002.02450*, 2020.

123 Tiancheng Zhao and Maxine Eskenazi. Zero-shot dialog generation with cross-domain latent actions. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 1–10, 2018.

124 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.

125 Chien-Sheng Wu, Richard Socher, and Caiming Xiong. Global-to-local memory pointer networks for task-oriented dialogue. *arXiv preprint arXiv:1901.04713*, 2019.

126 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.

127 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.

128 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.

129 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.

130 Shikib Mehri, Evgeniia Razumovsakaia, Tiancheng Zhao, and Maxine Eskenazi. Pretraining methods for dialog context representation learning. *arXiv preprint arXiv:1906.00414*, 2019.

131 Shikib Mehri and Maxine Eskenazi. Multi-granularity representations of dialog. *arXiv preprint arXiv:1908.09890*, 2019.

132 Yinhe Zheng, Guanyi Chen, and Minlie Huang. Out-of-domain detection for natural language understanding in dialog systems. *arXiv preprint arXiv:1909.03862*, 2019.

133 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.

134 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.