Task-Oriented Dialog System : A Survey

Anonymous CVPR submission

Paper ID *****

Abstract

Due to the significance and value in human-computer interaction and natural language processing, task-oriented dialog system is a hot research topic recently. In this paper, we aim to provide a survey on various task-oriented dialog system methods. Firstly, the popular methods of task-oriented dialog system are introduces in details. Secondly, we review the recent progresses in dialog evaluation. Thirdly, we discuss three dominant challenges for task-oriented dialog system. Finally, the future open research problems for task-oriented dialog system are pointed out.

1. Introduction

Building a task-oriented dialogue system has become a hot topic in the research community and industry. The task-oriented dialogue system is designed to help users complete certain tasks in specific fields, such as restaurant reservations, weather inquiries, and flight reservations, which makes it very valuable for real-world businesses. Compared with open-domain dialogue systems whose main goal is to maximize user participation [30], task-oriented dialogue systems are more inclined to complete certain tasks in one or more domains. Usually, task-oriented dialogue systems are built on a structured ontology, which defines the domain knowledge of the task.

Task-oriented dialog systems are divided by architecture into two categories. One type is a pipeline system that has a modular structure, as shown in Figure 1. It consists of four key modules:

- Natural Language Understanding (NLU): It identifies and parses a user's text input to obtain semantic tags that can be understood by computers, such as slot values and intentions.

- Dialog State Tracking (DST): It maintains the current dialog state based on the dialog history. The dialog state is the cumulative meaning of the dialog history, which is generally expressed as slot-value pairs.

- Dialog Policy: It outputs the next system action based on the current dialog state. The DST module and the dialog policy module are collectively referred to as the dialog manager (DM).

- Natural Language Generation (NLG): It converts system actions to natural language output.

This modular system structure is highly interpretable, easy to implement, and suitable for most practical task-oriented dialogue systems in the industry. However, this structure does not flexible enough. The modules are independent of each other and it is difficult to optimize together. This makes it difficult to adapt to changing application scenarios. In addition, due to the accumulation of errors between modules, the upgrade of a single module may require adjustment of the entire system.

Another implementation of a task-oriented dialog system is an end-to-end system, which has been a popular field of academic research in recent years [15, 45, 61, 116]. This type of structure trains an overall mapping relationship from the natural language input on the user side to the natural language output on the machine side. It is highly flexible and scalable, reducing labor costs for design and removing the isolation between modules. However, the end-to-end model places high requirements on the quantity and quality of data and does not provide clear modeling for processes such as slot filling and API calling. This model is still being explored and is as yet rarely applied in the industry

This paper aims to (1) give an overview about task-oriented dialogue systems especially recent advances from deep learning; and (2) discuss possible research directions. The remaining of the article is organized as follows. In section 2, we review task-oriented dialogue systems including pipeline and end- to-end methods. In Section 3, we discuss recent work on task-oriented dialog evaluation, including automatic, simulated, and human evaluation methods. In Section 4, we discuss challenges in this domain and existing works proposed to solve this challenges. In Section 5, we conclude the paper and discuss future research trends.

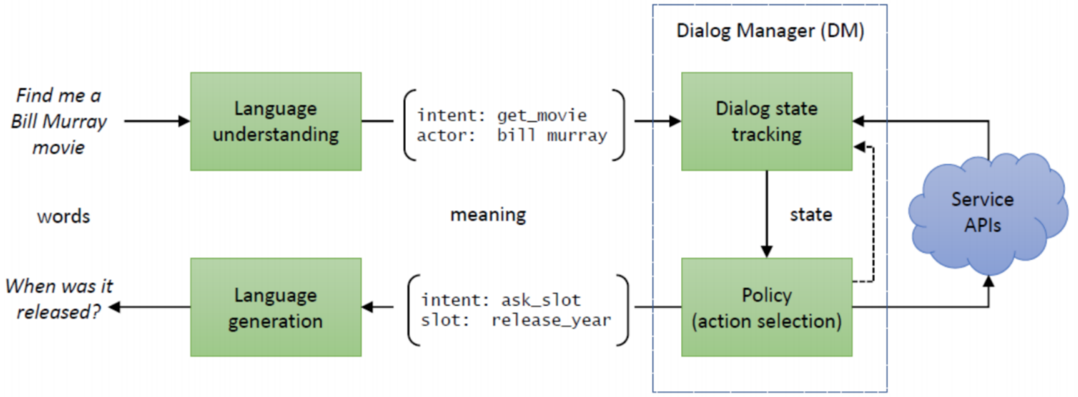

Figure 1. Modular structure of a task-oriented dialog system

2. Models and Approaches

2.1. Pipeline Methods

2.1.1 Language Understanding

Natural language understanding(NLU) typically involves identifying a user's intent and extracting relevant semantic constituents from the natural language sentence. It typically consists of two tasks: intent detection and slot filling [108]. While the intent detection is a standard classification problem where one label is predicted for each sentence, slot filling is often formulated as a sequence labelling task, where a sequence of labels need to be assigned jointly.

Intent Detection Given input utterance $X = (x_1, ..., x_n)$(n denotes the length of X), intent detection (ID) can be considered as a sentence classification task to decide the intent label o^I, which is formulated as:

$$o^I = Intent - Detection(X) \quad (1)$$

Many sentence classification methods have been investigated for intent detection. Xu and Sarikaya [125] used Convolutional Neural Network to extract 5-gram features and apply max-pooling to obtain work representations, which handles intent detection and slot filling simultaneously. Ravuri and Stolcke [86] successfully applied Recurrent Neural Network(RNN) and Long Short-Term Memory Network to intent detection task, which indicates the sequential features are beneficial to this task.

Slot Filling Slot filling can be seen as a sequence labeling task to produce a sequence slots $o^S = (o^S_1, ..., o^S_n)$, which can be rewritten as:

$$o^S = Slot - Filling(X) \quad (2)$$

Popular neural approaches for slot filling includes Conditional Random Field(CRF), Recurrent neural network (RNN) and RNN-based models. Yao et al. [130] take words as input as in a standard Recurrent Neural Network Language Models(RNN-LM), and then to predict slot labels rather than words on the output side. Mesnil et al. [66] carried out an in-depth investigation on the use of recurrent neural networks including Elman RNN, Jordan RNN and its bi-directional version for NLU. Yao et al. [128] have presented an application of LSTMs to spoken language understanding for the slot filling task. Yao et al. [129] have demonstrated that RNNs can be successfully merged with CRFs to do language understanding. The combination of RNN and CRF, recurrent conditional random field (R-CRF) is proved that can tackle label bias. Liu and Lane [53] proposed to model slot label dependencies using a sampling approach, by feeding sampled output labels (true or predicted) back to the sequence state. Vuetal [111] utilized ranking loss function on bi-RNN model in slot filling, further enhancing performance in ATIS dataset. Kurataetal [39] proposed an encoder-labeler LSTM that can conduct slot filling conditioned on the encoded sentence-level information.

Joint Models Considering the close correlation between intent detection and slot filling, dominant work in the literature adopts joint model to leverage the shared knowledge across tasks. Existing joint work can be classified into two main categories:implicit joint modeling and explicit joint modeling.

Implicit joint modeling denotes that model only adopts a shared encoder to capture shared features, without any explicit interaction, which is illustrated in Figure 2(b). Zhang and Wang [137] introduced a shared RNNs (Joint ID and SF) to learn the correlation between intent and slots. Liu and Lane [54] propose an attention-based neural network model for joint intent detection and slot filling (Attention BiRNN). Liu and Lane [55] used a shared RNN to jointly performs intent detection, slot filling, and language modeling, aiming to improve the ability of online prediction. Hakkani-Tür et al. [23] proposed a shared RNN-LSTM ar-

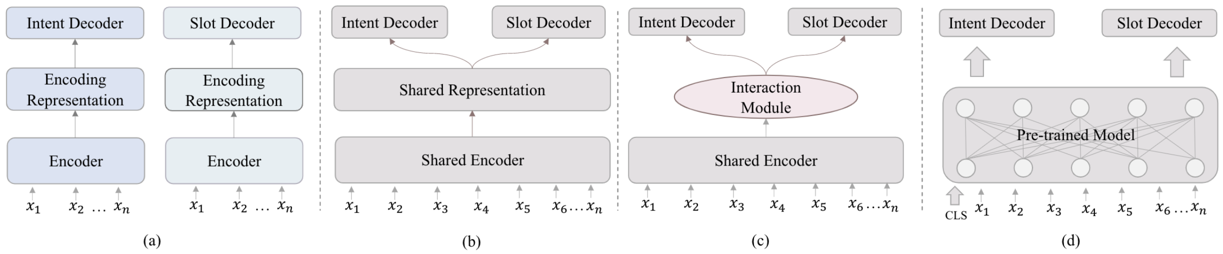

Figure 2. (a) Single models. (b) Implicit Joint Modeling. (c) Explicit Joint Modeling. (d) Pre-trained Model Paradigm

chitecture (Joint Seq) forjoint modeling.

In recent years, explicitly joint modeling has attracted more and more attention. It focus on the interaction between the indent detection and slot filling with an explicit interaction module, which is shown in Figure 2(c). This explicit modeling mode has the advantages of explicitly controlling process of interaction. Goo et al. [18] focuses on learning the explicit slot-intent relations by introducing a slot-gated mechanism into the state-of-the-art attention model. Qin et al. [79] propose a joint model for spoken language understanding with Stack-Propagation to better incorporate the intent information for slot filling. Wang et al. [114] proposed a novel Bi-model based RNN semantic frame parsing model for intent detection and slot filling. Zhang et al. [135] introduced a novel module to harness the hierarchical relationships among words, slots, and intents in the utterance for joint slot filling and intent detection.

Recently, Pre-trained Language Models (PLMs) achieve surprising results across various NLP tasks. Some BERT-based [12] pre-trained work has been explored in NLU direction where a shared BERT is considered as the encoder to extract contextual representations. In BERT-based models, each utterance starts with [CLS] and ends with [SEP], where [CLS] is the special symbol for representing the whole sequence, and [SEP] is the special symbol to separate non-consecutive token sequences. Further, the representation of the special token [CLS] is used for intent detection while other token representations are adopted for slot filling, which is shown in Figure 2(d). More specifically, Chen et al. [10] explored BERT for NLU proposing a joint intent classification and slot filling model based on BERT. In their work BERT is used to extract shared contextual embedding for intent detection and slot filling, which obtains a significant improvement compared with other non pre-trained models. Qin et al. [79]used pre-trained embedding encoder to replace its attention encoder, further boosting model's performance. Qin et al. [81] also explored BERT for NLU, which achieves state-of-the-art performance.

2.1.2 Dialogue State Tracking

In the spoken dialogue system, dialogue state tracking refers to the task of correctly inferring the dialogue state, such as the user's goal, given all the dialogue history records before the round. The existing work of DST can be divided into three categories: hand-crafted rules, generative models and discriminative models.

Hand-crafted Rules Early dialog system used hand-crafted rules for Dialog State Tracking(DST). The advantage of hand-made rules is that they do not require any data to implement, which is a benefit for bootstrapping. Rules also provide developers with an accessible way to integrate domain knowledge.

In the earliest form of rule based models, researchers generally considered a single NLU result, and tracked a single hypothesis for the dialog state. This setting transfer DST to an update rule $F(s, \bar{u}) = s'$ which maps from an existing states and the 1-best NLU result $\bar{u}$ to a new state s'. Larsson and Traum [41] used hand-written update rules to track a rich data structure called an "informationstate". The JUPITER [142] maintained a set of state variables which were updated using hand-written rules in a dialog control table. However, tracking a single dialog state has the shortcoming that it cannot make the use of whole dialog history.

Recent work of hand-crafted rules for DST compute scores for all dialog states suggested by the whole ASR/SLUN-best list. Wang and Lemon [115] introduces a simple rule-based belief tracker for dialogue systems, which can maintain beliefs over both marginal and joint representations of user goals using only the information observed within the dialogue itself. Sun et al. [101] proposed Markov Bayesian Polynomial (MBP), to formulate rule-based models which considered rule-based models as a special kind of polynomial functions satisfying some linear constraints.

Generative Models Generative models module the dialog to Bayesian network which relates the dialog state s to the system action a, the (true, unobserved) user action u, and NLU result $\bar{u}$. When the system action and NLU result are observed, a distribution over possible dialog states can be computed by applying Bayesian inference. A number of probabilistic formulations have been explored for how to relate these quantities; one illustrative example is:

$$b'(s') = \eta\Sigma_{u'} P(\bar{u}'|u')P(u'|s', a')\Sigma_{s'} P(s'|s, a)b(s) \quad (3)$$

where $b(s)$is the previous distribution over dialog states, $b(s)$is the (updated) distribution over dialog states being estimated, $P(\bar{u}'|u')$ is the probability of the LU producing the observed output $\bar{u}'$ given the (true, unobserved) user action u, $P(u|s, a)$ is the probability of the user taking action u given the true dialog state s and system action a,$P(s|s, a)$ is probability of the dialog state changing to s given it is currently s and the system takes action a, and η is a normalizing constant.

Early generative approaches used exact inference, enumerating all possible dialog states [29,91]. This is quadratic in the number of possible states, and is intractable as the number of states can be very large. As a result, two approximations are typically used; either maintaining a beam of candidate dialog states [24, 120, 134], or assuming conditional independence between components of the dialog state [8, 33, 34, 107, 121, 132].Typically these techniques use parameters optimized by the system designer, though there are methods to optimize the parameters using corpora of dialog data [42, 106]. Another proposition is to use a secondary step to post-process the output of a generative model [36].

Discriminative Models Generative methods must model all the correlations in the input features, so they cannot easily take advantage of arbitrary features of the dialogue history. Compared with the generative model, the discriminative model directly simulates the distribution of dialogue states given arbitrary and possibly relevant input features. Although the parameters of the generative model are usually made by hand, the model parameters of the discriminative method are adjusted using machine learning, and the dialogue data is labeled. Discriminative methods using machine learning can be divided into two categories-static classifiers that encode the history of the dialogue in the input features, and sequence models that explicitly model the dialogue as a sequence process.

Static classifiers encode dialog history in the input features. Metallinou et al. [67] alter the logistic regression model so that it learns a single weight for each feature. This enables an arbitrary number of hypotheses to be scored since the number of weights to learn no longer increases with the number of state hypotheses to score. Henderson et al. [25] presented a discriminative approach for tracking the state of a dialog applying a deep neural network as a classifier.

Sequence models explicitly model dialog as a sequential process. First,a discriminative Markov Model can be applied, where the distribution from the previous turn's prediction can be used as a feature. Ren et al. [87]proposed a novel approach to enhancing the model capability of stationary discriminative models in dialog state tracking by assuming Markovian dependencies between adjacent turns. Second, a dialog can be cast as a conditional random field(CRF) in which features are associated with each dialog turn, and CRF decoding determines the most likely final dialog state conditioned on the entire sequence [37, 43, 60] Third, recurrent neural networks can be estimated where the inputs are the observed ASR/SLUresults, and the output is a distribution over dialog states [26].

2.1.3 Dialog Policy

Dialog Policy Learning(DPL) systems are responsible for generating the next system action, which is conditioned on the dialogue state. There are three kinds of methods to improve the efficiency of DPL, namely user modeling, hindsight experience replay, and reward shaping.

User Modeling Researchers have developed "two-step" algorithms that first build user models through supervised learning with real conversational data, and then learn dialog policies by interacting with the simulated users [49,92]. In those methods, user modeling must be conducted offline before the start of DPL. As a result, the learned policies are potentially biased toward the historical conversational data. Toward online methods for DPL, researchers have developed algorithms for simultaneously constructing models of real users,and learning from the simulated interaction experience with user models [13,48,52,56,98,122,139]. Those methods enable agents to simultaneously build and leverage user models in dialog policy learning.

Hindsight Experience Replay In comparison to many other RL applications, task-oriented dialog systems have very sparse feedback from the "real world" (human users), where one frequently cannot tell dialogs being successful or not until reaching the very end. Positive feedback is even rarer, when dialog policies are of poor qualities. Hindsight experience replay(HER) [1] methods have been developed to convert unsuccessful trials into successful ones through goal manipulation. The "policy learning with hindsight" idea has been applied to various domains, including dialog [59] .

Reward Shaping Within the dialog policy learning context, reward shaping is another way of providing the dialog agents with extra feedback, where a dense reward function can be manually designed [100],or learned [98]. Researchers also developed efficient exploration strategies to speedup the policy learning process of dialog agents,e.g., [40,77]. Those methods are orthogonal to ours, and can potentially be combined to further improve the dialog learning efficiency.

2.1.4 Natural Language Generation

Natural Language Generation(NLG) is responsible for mapping the system dialogue action produced by the DPL to a natural language utterance and is modelled as a conditional language generation task.

Conventional Methods In order to reduce complexity, conventional approaches to NLG typically divide the task into content selection, sentence planning, and surface realisation. In the dialogue system framework, content selection is typically modelled by the dialogue manager module and therefore can be treated as given in the NLG pipeline [17, 133]. Sentence planning then maps input semantic symbols into an intermediary form representing the utterance, e.g. a syntax tree or a template, then surface realisation converts the intermediate structure into the final text [113]. Although statistical sentence planning has been explored previously, for example, generating the most likely context-free derivations given a corpus [3] or maximising the expected reward using reinforcement learning [90], these methods still rely on a pre-existing, handcrafted generator. To minimise handcrafting,Stent and Molina [97]proposed learning sentence planning rules directly from a corpus of utterances labelled with Rhetorical Structure Theory (RST) discourse relations [64]. However, the required corpus labelling is expensive and additional handcrafting is still needed to map the sentence plan to a valid syntactic form.

Corpus-based Methods Corpus-based NLG aims at learning generation decisions from data with minimal dependence on rules and heuristics. A pioneer in this direction is the class-based n-gram language model approach proposed by Oh and Rudnicky [73]. Ratnaparkhi [85]later addressed some of the limitations of class-based language models in the over-generation phase by using a modified generator based on a syntactic dependency tree. Mairesse and Young [63] proposed a phrase-based NLG system based on factored language models that can learn from a semantically aligned corpus. Although active learning [62]was also proposed to allow learning on-line directly from users, the requirement for human annotated alignment of semantic units with the corresponding words in the utterance limits the practicality of such systems. Another similar approach casts NLG as a template extraction and matching problem, e.g., Angeli et al. [2] train a set of log-linear models to make a series of generation decisions to choose the most suitable template for realisation. However, template matching approaches do not generalise well to unseen combinations of semantic elements.

Network-based Methods The use of neural network-based (NN) methods to NLG is relatively unexplored. The stock reporter system ANA by Kukich [38] is perhaps the earliest example of an NN-based generator, although generation was only done at the phrase level. Recent advances in recurrent neural network-based language models (RNNLM) Mikolov et al. [68, 69] have demonstrated the value of distributed representations and the ability to model arbitrarily long dependencies. Sutskever et al. [102] describes a simple variant of the RNN that can generate meaningful sentences by learning from a character-level corpus. More recently, Karpathy and Fei-Fei [35] have demonstrated that an RNNLM is capable of generating image descriptions by conditioning the network model on a pre-trained convolutional image feature representation. Zhang and Lapata [136] also describes interesting work using RNNs to generate Chinese poetry. Due to RNN's limited ability to model long range dependencies [4], the use of Long Short-term Memory (LSTM) networks [28] has become popular and has been shown to be superior in a variety of tasks, such as speech recognition [20], handwriting recognition [19], machine translation [11, 103], and program induction [21]. Recently, the use of LSTM networks for sequence-to-sequence learning [103] have been widely studied for modeling chatbot [95, 110] and task-oriented dialogue [117, 118] in an end-to-end fashion.

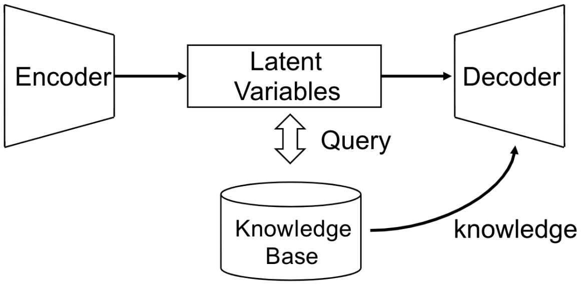

Figure 3. Framework of end-to-end dialog systems. It first encodes natural language context to obtain some latent variables, which can be used for KB query. Then based on the latent variables and query results, the decoder generates a natural language response

2.2. End-to-End Methods

Generally speaking, the components in the pipeline system are individually optimized. This modular structure leads to complex model design, and the performance of each individual component does not necessarily translate into the progress of the entire system [16]. The end-to-end approaches for task-oriented dialog systems are inspired by the researches on open-domain dialog systems, which use neural models to build the system in an end-to-end manner without modular design, as shown in Figure 3. Most of these methods utilized sequence to sequence models as the infrastructural framework, which is end-to-end differentiable and can be optimized by gradient-based methods. 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. [116] 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. [5]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. [61] took a similar approach and further feedthe knowledge base information into the memory networks.In [15] a new memory network structure named key-valuememory networks is introduced to extract relevant information from KB through key-value retrieval. Lei et al. [45] proposed a two-step seq2seq generation model which by passed 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 [13, 45, 48, 56, 122, 139]. In [139], 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. [122] proposed LSTM-based hybrid code networks (HCN), which supports self-defined software.

3. Evaluation

Evaluation of task-oriented dialogue systems is still an active research field because of the multiplicity of possible acceptable responses an agent could give. Most evaluation studies follow the PARADISE [112] 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 users 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 [105]. 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 [109] and ConvLab [44, 141]. 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 [78].

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 [57] and the quality of response generation [119], to assess real user satisfaction. Human evaluation metrics include task success rate, irrelevant turn rate, redundant turn rate, user satisfaction score, etc. The researchers generally hire human users on the crowd-sourcing platform, and human evaluation can be conducted in the following two ways. One is indirect evaluation that asking the annotators to read the simulated dialog between the dialog system and the user simulator, then rate the score [96] or give their preference among different systems [104] 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 [47].

4. Challenges

4.1. Data Efficiency

Different from the research in open-domain dialog systems, data-driven approaches for task-oriented dialog systems often require fine-grained annotations to learn the dialog model in a specific domain, e.g., dialog act and state labels. However, it is often difficult to obtain a large-scale annotated corpus in a specific domain since (1) collecting a domain-specific cor-pus 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.

4.1.1 Transfer Learning

Transfer learning relaxes the assumption of traditional machine learning that the training data and testing data should be sampled independently from an identical probability distribution, thus it can be applied to discover domain-invariant intrinsic features and structures underlying two different but related domains, which establishes successful transfer and reutilization of supervised information across domains.

The same problem often occurs in task-oriented dialogue systems. For example, in the case of limited data in the hotel field, how does the restaurant reservation dialogue system adapt to hotel reservations? In this case, the ontologies of the two domains are similar, sharing many dialogue behaviors and slots. In this case, transfer learning can greatly reduce the amount of target data required for this adaptation.

Cross-lingual transfer learning methods are recently proposed. In [93], Schuster et al. studied three different cross-lingual transfer methods: (1) translating the training data, (2) using cross-lingual pre-trained embeddings, and (3) a novel method of using a multilingual machine translation encoder as contextual word representations. MLT [58] is a novel zero-shot adaptation method for cross-lingual task-oriented dialogue systems which leverages very few task-related parallel word pairs to generate code-switching sentences for learning the interlingual semantics across languages/

For domain transfer, a novel [71]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 [31], 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. [9]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, 89], the parameters of DST models are shared across domains and is independent of predefined value sets. Therefore, the model is able to transfer to previously unseen domains.

4.1.2 Unsupervised Methods

A key problem in the learning of dialogue strategies is to estimate the reward signal, which is difficult to obtain in practical applications. Therefore, establishing a reward estimation model is necessary for dialogue strategy learning, especially during RL training. By treating the dialogue strategy as a generator and the reward function as a discriminator, a Generative Adversarial Network (GAN) can be used to learn the reward function in an unsupervised manner. Liu et al. [57] first used GAN to learn a binary reward function by discriminating simulated from real user dialogs. Xu et al. [126] extended this idea for detecting dialog failure by using the predicted reward as an indicator of failure. Su et al. [98]used another way for reward estimation using Gaussian Process. By modeling the uncertainty of predicted reward, the model can actively re-quire human intervention on potential failure cases. In their experiment, the requirement for human intervention dramatically decreases with the reduction in the uncertainty of re-ward estimation, which remarkably relieve manual annotation.

Recently, pre-training methods show superior performance on many NLP tasks. In such approaches, extensive linguistic features can be transferred from large-scale unlabeled copora using unsupervised pre-training tasks, such as mask language modeling (MLM) and next sentence prediction (NSP).Wolf et al. [123] followed this way by first pre-training a trans-former model on large-scale dialog data and then fine-tuningthe model on a personalized dialog task with multi-task learn-ing. Budzianowski et al. [6] further explored this idea to task-oriented dialog without explicit standalone dialogue pol-icy 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

4.2. Multi-turn Dynamics

Compared with open-domain dialogue systems, a major feature of task-oriented dialogue systems is the emphasis on multi-turn state-action dynamics, which is mainly related to dialogue management (DST and strategy). In a task-oriented dialogue system, the completion of a specific task is critical. Therefore, the dialog management research responsible for tracking the dialogue status and dialogue process is the backbone of the dialogue system. 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.

4.2.1 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 with-out 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 [27], which maintains a distribution over all possible values for each slot.Therefore, early methods commonly formulated DST as a classification task [**?**, 25, 71, 72].

Zhong et al. [138] 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. [45] 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 [22] takes the user utterance as input and generates a be-lief state span. Then in the second stage, based on the utterance and belief span, another CopyNet generates the response utterance.

Recently, some multi-domain datasets are proposed to promote the research in this direction [7, 84]. Compared to single-domain tasks, the DST in multi-domain scenario has to predict the domain of slot values. Wu et al . proposed TRADE [84], a transferable multi-domain DST using seq2seq model with CopyNet [22] to predict values. The parameters are shared across domains, enabling zero-shot DST for unseen domains. COMER [88] further reduces 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.

4.2.2 Dialog Planning

Model-free RL methods dominated the early studies of neural dialog policy by learning through interaction with real users [13, 70, 122]. It is data-hungry, requiring a large amount of interactions to train a policy model effectively. One common solution is to use user simulators [49, 92]. 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 [96] .

In order to alleviate these problems, model-based RL methods are proposed to model the environment so as to plan the dialogue strategy learning. In the model-based RL method, the environment is modeled to simulate the dynamics of the dialogue. Then in the RL training phase, the dialogue strategy is alternately trained by learning from real users and planning using the environment model. Peng et al. [74] proposed the Deep Dyna-Q (DDQ) framework, which first integrated model-based planning for task-oriented dialogue systems.

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 St 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 [46, 131] introduce dialog roll out planning into negotiation dialogs, in which the agent simulates complete dialogues in a specific state St 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. [32] proposed to look only several limited steps ahead and use those steps as additional features for the policy model to alleviate decision making.

4.2.3 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 [99]. However, in real-world applications where user goals are not available, such rewards cannot be effectively estimated. Another problem is that the reward signal is inconsistent when calculated objectively through predefined rules or subjectively judged by real users. In order to alleviate the above problems, there are some studies to learn independent reward functions to provide reliable supervision signals.

One method for reward estimation is off-line learning with annotated data [127]. 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 [98]. 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.

4.3. Ontology Integration

One of the main problems of task-oriented dialogue systems is to integrate dialogue ontology into dialogue models, including domain schema and knowledge bases. In most previous methods, domain schema are pre-defined and highly dependent on the corpus they use. For example, the slot of the restaurant domain contains address area, dish type, price range, etc. Querying the database and retrieving the results is essential for task-oriented dialogue systems to make decisions and produce appropriate responses. Recently, there have been many efforts to integrate external databases or API calls.

However, due to the large-scale task domain, the ontology integration of task-oriented dialogue models becomes more challenging. Although pre-defined ontology can be considered in model design, these methods are coupled with domain schema and cannot be easily transferred to new tasks. Although it is proposed to add an end-to-end model to alleviate the pattern integration problem, it is not easy to involve contextual information and knowledge base, because unlike the pipeline method, there is no explicit dialog state representation to generate explicit knowledge basic queries.

4.3.1 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 [44] provides additional user dialog act annotation in the MultiWOZ [7]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 [88]extends the representation of dialog states with priority operator that considers the user's preference on slot values. For NLG task, Peng et al. [75] 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.

4.3.2 Knowledge Base Integration

For task-oriented dialogue systems, it is very important to query external knowledge bases to obtain the information that users query. Early models or pipeline systems retrieve items from the knowledge base by issuing queries based on the current dialog state during dialog interaction, which requires some manual effort. Due to the increasing task complexity in task-oriented scenarios, training end-to-end dialogue systems without intermediate supervision will be more attractive. However, unlike the pipeline method, there is no clear structured dialog state representation in the end-to-end method. Therefore, the knowledge base interaction is carried out by using the intermediate latent representation of the model, and is seamlessly trained 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. [14]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 [61, 124]. In such approaches, the dialog con-text 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 con-text and knowledge base are heterogeneous information from different sources. Lin et al. [51] 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 [80], 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.

5. Conclusion 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. Despite these themes, there are still some interesting and challenging questions. We finally conclude some future trends in task-oriented dialogue systems:

Pre-training Methods for Dialog Systems. Data scarcity is a key challenge in building a task-oriented dialogue system. On the one hand, collecting enough data for a specific area is time-consuming and expensive. On the other hand, the task-oriented dialogue system is a composite NLP task. It not only learns grammar, reasoning, decision-making and language generation from offline data, but also learns from online interaction with users, presenting more fine-grained requirements. Data annotation and model design.

Recently, pre-trained models have shown excellent performance in many NLP tasks [12, 76, 82]. In this case, the basic model is first pre-trained on a large-scale corpus through some unsupervised pre-training tasks, such as mask language model and next sentence prediction. In the pre-training stage, the basic model can capture implicit language knowledge and learn from a large-scale corpus. Using this implicit knowledge, the basic model can quickly adapt to the target task by simply fine-tuning the target task's data. This idea can also be applied to task-oriented dialogue systems to transfer general natural language knowledge corpus to specific dialogue tasks on a large scale. Some early studies have shown the possibility of using pre-trained models to model task-oriented dialogue [6, 65, 75, 123].

Domain Adaptation Unlike open domain dialogue, task-oriented dialogue is based on a well-defined domain ontology, which constrains agent actions, slot values and knowledge bases for specific tasks. Therefore, in order to complete a task, the model of the dialogue system is highly dependent on the domain ontology. However, in most existing studies, this ontology knowledge is hard-coded into the model. For example, dialogue behavior types, slot value vocabulary and even slot-based belief states are all embedded in the model. This hard-coded ontology embedding raises two problems: (1) Human experts are required to analyze tasks and integrate domain ontology into model design, which is a time-consuming process. (2) The existing model cannot be easily transferred to another task. Therefore, decoupling domain ontology and dialogue model to obtain better adaptive performance is a key issue. One ultimate goal is to achieve zero-shot domain adaptation, which can directly construct a dialogue system with a given ontology just like a human, without any training data.

Robustness.The robustness of deep neural models has always been a challenging problem because existing neural models are susceptible to simple input disturbances. For task-oriented dialogue systems, robustness is also a key issue, which mainly comes from two aspects: (1) On the one hand, task-oriented dialogue highly relies on domain ontology. Therefore, in many studies, training data is limited to reasonable examples with little noise. However, models that are strained in this particular way often fail to meet the requirements in practical applications. In these applications, there are many out-of-domain or out-of-distribution inputs [140], such as previously invisible slot values. A powerful dialogue system should be able to handle noise and previously invisible input after deployment. (2) On the other hand, the decision-making of the neural dialogue strategy model is uncontrollable. It is trained through offline imitation learning and online reinforcement learning. Robustness of decision-making is very important to its performance, especially for some special applications with low tolerance for errors, such as in the medical and military fields. Therefore, improving the robustness of neural dialogue models is an important issue. One possible approach is to combine robust rule-based methods with neural models, such as neural symbolic machines [50, 94], which may make the model not only more robust, but also easier to interpret.

End-to-end Modeling Compared with the pipeline method, end-to-end dialog system modeling has attracted more and more attention in recent years. It is easier to train end-to-end models without the need to explicitly model conversation states and strategies. However, existing end-to-end methods still require some intermediate supervision to improve model performance.

For example, in [119], a modular-based end-to-end frame-work 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 [45], although a seq-to-seq framework is proposed to avoid component barriers, intermediate output named belief spansis 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.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017. 4

[2] Gabor Angeli, Percy Liang, and Dan Klein. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512, 2010. 5

[3] Anja Belz. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455, 2008. 5

[4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 5

[5] Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*, 2016. 6

[6] Paweł Budzianowski and Ivan Vulić. 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. 7, 10

[7] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018. 8, 9

[8] Trung H Bui, Mannes Poel, Anton Nijholt, and Job Zwiers. A tractable hybrid ddn–pomdp approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering*, 15(2):273–307, 2009. 4

[9] 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. 7

[10] Qian Chen, Zhu Zhuo, and Wen Wang. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019. 3

[11] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 5

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3, 10

[13] 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. 4, 6, 8

[14] 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. 9

[15] Mihail Eric and Christopher D Manning. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414*, 2017. 1, 6

[16] 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. 5

[17] Milica Gašić and Steve Young. Gaussian processes for pomdp-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40, 2013. 5

[18] 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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, 2018. 3

[19] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008. 5

[20] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013. 5

[21] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 5

[22] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016. 8

[23] Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, pages 715–719, 2016. 2

[24] James Henderson and Oliver Lemon. Mixture model pomdps for efficient handling of uncertainty in dialogue management. In *Proceedings of ACL-08: HLT, Short Papers*, pages 73–76, 2008. 4

[25] 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. 4, 8

[26] Matthew Henderson, Blaise Thomson, and Steve Young. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 360–365. IEEE, 2014. 4

[27] 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. 8

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 5

[29] Eric Horvitz and Tim Paek. A computational architecture for conversation. In *UM99 User Modeling*, pages 201–210. Springer, 1999. 4

[30] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32, 2020. 1

[31] Vladimir Ilievski, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. Goal-oriented chatbot dialog management bootstrapping with transfer learning. *arXiv preprint arXiv:1802.00500*, 2018. 7

[32] Zhuoxuan Jiang, Xian-Ling Mao, Ziming Huang, Jie Ma, and Shaochun Li. Towards end-to-end learning for efficient dialogue agent by modeling looking-ahead ability. *arXiv preprint arXiv:1908.05408*, 2019. 8

[33] Rudolf Kadlec, Jindřich Libovickỳ, Jan Macek, and Jan Kleindienst. Ibm's belief tracker: Results on dialog state tracking challenge datasets. In *Proceedings of the EACL 2014 Workshop on Dialogue in Motion*, pages 10–18, 2014. 4

[34] Rudolf Kadlec, Miroslav Vodolan, Jindř ich Libovickỳ, Jan Macek, and Jan Kleindienst. Knowledge-based dialog state tracking. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 348–353. IEEE, 2014. 4

[35] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 5

[36] Daejoong Kim, Jaedeug Choi Choi, Kee-Eung Kim, Jungsu Lee, and Jinho Sohn. Engineering statistical dialog state trackers: A case study on dstc. In *Proceedings of the SIGDIAL 2013 Conference*, pages 462–466, 2013. 4

[37] Seokhwan Kim and Rafael E Banchs. Sequential labeling for tracking dynamic dialog states. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 332–336, 2014. 4

[38] Karen Kukich. Where do phrases come from: Some preliminary experiments in connectionist phrase generation. In *Natural language generation*, pages 405–421. Springer, 1987. 5

[39] Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. Leveraging sentence-level information with encoder lstm for semantic slot filling. *arXiv preprint arXiv:1601.01530*, 2016. 2

[40] Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149, 2003. 4

[41] Staffan Larsson and David R Traum. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural language engineering*, 6(3-4):323–340, 2000. 3

[42] Byung-Jun Lee, Woosang Lim, Daejoong Kim, and Kee-Eung Kim. Optimizing generative dialog state tracker via cascading gradient descent. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 273–281, 2014. 4

[43] Sungjin Lee and Maxine Eskenazi. Recipe for building robust spoken dialog state trackers: Dialog state tracking challenge system description. In *Proceedings of the SIGDIAL 2013 Conference*, pages 414–422, 2013. 4

[44] Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Xiang Li, Yaoqin Zhang, Zheng Zhang, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, et al. Convlab: Multi-domain end-to-end dialog system platform. *arXiv preprint arXiv:1904.08637*, 2019. 6, 9

[45] 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. 1, 6, 8, 10

[46] Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*, 2017. 8

[47] 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. 7

[48] Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*, 2017. 4, 6

[49] 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. 4, 8

[50] 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. 10

[51] 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. 10

[52] Zachary C Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. Efficient exploration for dialogue policy learning with bbq networks & replay buffer spiking. *arXiv preprint arXiv:1608.05081*, 3, 2016. 4

[53] Bing Liu and Ian Lane. Recurrent neural network structured output prediction for spoken language understanding. In *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions*, 2015. 2

[54] Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*, 2016. 2

[55] Bing Liu and Ian Lane. Joint online spoken language understanding and language modeling with recurrent neural networks. *arXiv preprint arXiv:1609.01462*, 2016. 2

[56] 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. 4, 6

[57] Bing Liu and Ian Lane. Adversarial learning of task-oriented neural dialog models. *arXiv preprint arXiv:1805.11762*, 2018. 6, 7

[58] Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8433–8440, 2020. 7

[59] Keting Lu, Shiqi Zhang, and Xiaoping Chen. Goal-oriented dialogue policy learning from failures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2596–2603, 2019. 4

[60] Yi Ma and Eric Fosler-Lussier. Detecting 'request alternatives' user dialog acts from dialog context. In *Situated Dialog in Speech-Based Human-Computer Interaction*, pages 97–102. Springer, 2016. 4

[61] 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. 1, 6, 10

[62] François Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1552–1561, 2010. 5

[63] François Mairesse and Steve Young. Stochastic language generation in dialogue using factored language models. *Computational Linguistics*, 40(4):763–799, 2014. 5

[64] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988. 5

[65] Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. Pretraining methods for dialog context representation learning. *arXiv preprint arXiv:1906.00414*, 2019. 10

[66] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*, pages 3771–3775, 2013. 2

[67] Angeliki Metallinou, Dan Bohus, and Jason D Williams. Discriminative state tracking for spoken dialog systems. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 466–475, 2013. 4

[68] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010. 5

[69] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE, 2011. 5

[70] 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. 8

[71] 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. 7, 8

[72] Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016. 8

[73] Alice Oh and Alexander Rudnicky. Stochastic language generation for spoken dialogue systems. In *ANLP-NAACL 2000 Workshop: Conversational Systems*, 2000. 5

[74] 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. 8

[75] 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. 9, 10

[76] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018. 10

[77] Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Frezza-Buet. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):1–21, 2011. 4

[78] Olivier Pietquin and Helen Hastie. A survey on metrics for the evaluation of user simulations. *The knowledge engineering review*, 28(1):59–73, 2013. 6

[79] Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. A stack-propagation framework with token-level intent detection for spoken language understanding. *arXiv preprint arXiv:1909.02188*, 2019. 3

[80] 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.

[81] Libo Qin, Minheng Ni, Yue Zhang, Wanxiang Che, Yangming Li, and Ting Liu. Multi-domain spoken language understanding using domain-and task-aware parameterization. *arXiv preprint arXiv:2004.14871*, 2020. 3

[82] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 10

[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. 7

[84] 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, pages 8689–8696, 2020. 8

[85] Adwait Ratnaparkhi. Trainable approaches to surface natural language generation and their application to conversational dialog systems. *Computer Speech & Language*, 16(3-4):435–455, 2002. 5

[86] Suman Ravuri and Andreas Stolcke. Recurrent neural network and lstm models for lexical utterance classification. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015. 2

[87] Hang Ren, Weiqun Xu, and Yonghong Yan. Markovian discriminative modeling for cross-domain dialog state tracking. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 342–347. IEEE, 2014. 4

[88] Liliang Ren. *Scalable and accurate dialogue state tracking via hierarchical sequence generation*. University of California, San Diego, 2020. 8, 9

[89] Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. Towards universal dialogue state tracking. *arXiv preprint arXiv:1810.09587*, 2018. 7

[90] Verena Rieser and Oliver Lemon. Natural language generation as planning under uncertainty for spoken dialogue systems. In *Empirical methods in natural language generation*, pages 105–120. Springer, 2009. 5

[91] Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 93–100, 2000. 4

[92] 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, 2007. 4, 8

[93] Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. Cross-lingual transfer learning for multilingual task oriented dialog. *arXiv preprint arXiv:1810.13327*, 2018. 7

[94] 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. 10

[95] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015. 5

[96] 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. 8

[97] Amanda Stent and Martin Molina. Evaluating automatic extraction of rules for sentence plan construction. In *Proceedings of the SIGDIAL 2009 Conference*, pages 290–297, 2009. 5

[98] Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. On-line active reward learning for policy optimisation in spoken dialogue systems. *arXiv preprint arXiv:1605.07669*, 2016. 4, 7, 9

[99] Pei-Hao Su, Milica Gašić, and Steve Young. Reward estimation for dialogue policy optimisation. *Computer Speech & Language*, 51:24–43, 2018. 8

[100] Pei-Hao Su, David Vandyke, Milica Gasic, Nikola Mrksic, Tsung-Hsien Wen, and Steve Young. Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. *arXiv preprint arXiv:1508.03391*, 2015. 4

[101] Kai Sun, Lu Chen, Su Zhu, and Kai Yu. A generalized rule based tracker for dialogue state tracking. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 330–335. IEEE, 2014. 3

[102] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *ICML*, 2011. 5

[103] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 5

[104] Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. *arXiv preprint arXiv:1908.10719*, 2019. 6

[105] 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. 6

[106] Blaise Thomson, F Jurčíček, M Gašić, Simon Keizer, François Mairesse, Kai Yu, and Steve Young. Parameter learning for pomdp spoken dialogue models. In *2010 IEEE Spoken Language Technology Workshop*, pages 271–276. IEEE, 2010. 4

[107] Blaise Thomson and Steve Young. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010. 4

[108] Gokhan Tur and Renato De Mori. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011. 2

[109] 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. 6

[110] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015. 5

[111] Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6060–6064. IEEE, 2016. 2

[112] Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. Paradise: A framework for evaluating spoken dialogue agents. *arXiv preprint cmp-lg/9704004*, 1997. 6

[113] Marilyn A Walker, Owen C Rambow, and Monica Rogati. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, 16(3-4):409–433, 2002. 5

[114] Yu Wang, Yilin Shen, and Hongxia Jin. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. *arXiv preprint arXiv:1812.10235*, 2018. 3

[115] Zhuoran Wang and Oliver Lemon. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432, 2013. 3

[116] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve J. Young. A network-based end-to-end trainable task-oriented dialogue system. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 438–449. Association for Computational Linguistics, 2017. 1, 5

[117] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. Conditional generation and snapshot learning in neural dialogue systems. *arXiv preprint arXiv:1606.03352*, 2016. 5

[118] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina M Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. Multi-domain neural network language generation for spoken dialogue systems. *arXiv preprint arXiv:1603.01232*, 2016. 5

[119] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016. 6, 10

[120] Jason D Williams. Using particle filters to track dialogue state. In *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 502–507. IEEE, 2007. 4

[121] Jason D Williams. Incremental partition recombination for efficient tracking of multiple dialog states. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5382–5385. IEEE, 2010. 4

[122] 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. 4, 6, 8

[123] 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. 7, 10

[124] Chien-Sheng Wu, Richard Socher, and Caiming Xiong. Global-to-local memory pointer networks for task-oriented dialogue. *arXiv preprint arXiv:1901.04713*, 2019. 10

[125] Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 ieee workshop on automatic speech recognition and understanding*, pages 78–83. IEEE, 2013. 2

[126] 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. 7

[127] 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. 9

[128] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. Spoken language understanding using long short-term memory neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 189–194. IEEE, 2014. 2

[129] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. Recurrent conditional random field for language understanding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4077–4081. IEEE, 2014. 2

[130] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528, 2013. 2

[131] Denis Yarats and Mike Lewis. Hierarchical text generation and planning for strategic dialogue. In *International Conference on Machine Learning*, pages 5591–5599. PMLR, 2018. 8

[132] Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174, 2010. 4

[133] 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. 5

[134] Steve Young, Jost Schatzmann, Karl Weilhammer, and Hui Ye. The hidden information state approach to dialog management. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–149. IEEE, 2007. 4

[135] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*, 2018. 3

[136] Xingxing Zhang and Mirella Lapata. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, 2014. 5

[137] Xiaodong Zhang and Houfeng Wang. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, volume 16, pages 2993–2999, 2016. 2

[138] 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. 8

[139] 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. 4, 6

[140] Yinhe Zheng, Guanyi Chen, and Minlie Huang. Out-of-domain detection for natural language understanding in dialog systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1198–1209, 2020. 10

[141] 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. 6

[142] Victor Zue, Stephanie Seneff, James R Glass, Joseph Polifroni, Christine Pao, Timothy J Hazen, and Lee Hetherington. Juplter: a telephone-based conversational interface for weather information. *IEEE Transactions on speech and audio processing*, 8(1):85–96, 2000. 3