Prototypical Q Networks for Automatic Conversational Diagnosis and Few-Shot New Disease Adaption

Hongyin Luo[1], Shang-Wen Li[2], James Glass[1]

[1]MIT CSAIL
[2]Amazon AI

`hyluo@mit.edu, shangwel@amazon.com, glass@mit.edu`

Abstract

Spoken dialog systems have seen applications in many domains, including medical for automatic conversational diagnosis. State-of-the-art dialog managers are usually driven by deep reinforcement learning models, such as deep Q networks (DQNs), which learn by interacting with a simulator to explore the entire action space since real conversations are limited. However, the DQN-based automatic diagnosis models do not achieve satisfying performances when adapted to new, unseen diseases with only a few training samples. In this work, we propose the Prototypical Q Networks (ProtoQN) as the dialog manager for the automatic diagnosis systems. The model calculates prototype embeddings with real conversations between doctors and patients, learning from them and simulator-augmented dialogs more efficiently. We create both supervised and few-shot learning tasks with the Muzhi corpus. Experiments showed that the ProtoQN significantly outperformed the baseline DQN model in both supervised and few-shot learning scenarios, and achieves state-of-the-art few-shot learning performances.

Index Terms: dialog system, human-computer interaction, automatic diagnosis

1. Introduction

Recently spoken dialog systems have been a popular research topic in human language technology (HLT) area with various applications. Among these applications, dialog systems for clinical conversation (i.e., medical bot) is a rising direction for its widespread and impactful use [1]. Medical bot assists medical practitioners to converse with patients, collect information about their symptoms, physical and mental conditions, or even make suggestions on diagnosis. The bot has significant potential to make the diagnostic procedure more efficient. An example for automatic diagnosis dialog system is shown in Figure 1. Starting from a self-report, the medical bot collects and distills symptom information before it makes the disease prediction.

One of the core challenges of building such a dialog system is design and train a dialog policy manager that can reason and decide the action to take based on the understanding of user intentions and conversation context. It is more challenging for medical bot because of the need of integrating medical knowledge for reasoning and decision making. [1] proposed a reinforcement learning (RL) framework with a multi-layer perceptron deep Q networks (MLP-DQN) [2]. [3] extended the study by enhancing the DQN with hand-crafted features among diseases and symptoms, generated from the training set. However, both models cannot directly learn from real doctor-patient conversations. For RL agent to fully explore the entire action space, the agent can only learn by interacting with rule-based

---
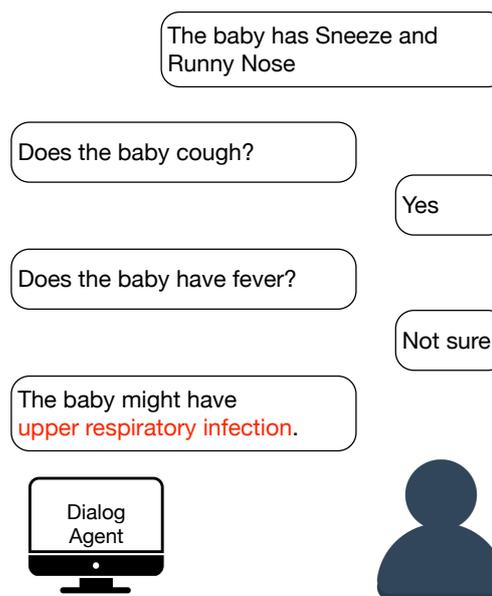
Work done before the second author joined Amazon

Figure 1: *An example of a dialog between a doctor and a user. Firstly, the user provides a self report. Then the agent conduct a dialog by requesting symptoms and conclude by making a decision about the disease.*

user simulators, which can not learn from the dialog histories between real doctors and patients.

Another difficulty faced by RL-based manager is adapting trained policy to new tasks (e.g., adapt trained medical bot to serve new diseases), since adaptation data is usually limited and hard to collect. On the other hand, Meta-learning algorithms are proposed to improve the model performance when training or adaptation data is limited. [4, 5]. Since both many- and few-shot learning heavily depend on the quality of learned representations, these studies encouraged us to combine meta-learning and deep reinforcement learning models to improve the dialog agents for automatic diagnosis in both scenarios by learning better representations of dialog actions and domain knowledge.

In this work, we propose prototypical Q networks (ProtoQN) borrowing the ideas of prototypical networks [5] and matching networks [6]. We evaluate the model in the medical dialog domain, since it is important and medical conversation usually suffers from data scarcity. The model makes fully use of real doctor-patient conversations by calculating prototype disease and symptom embeddings by encoding the dialog histories in the training set. Experiment results have shown that by learning a shared prototype embedding space, the ProtoQN out-

performs MLP-DQN under both experiment settings. The experiments in this paper will focus on medial dialog to show benefit of proposed method, but we believe the conclusion can be generalized to other domains, since we do not use handcrafted features or external domain specific information.

2. Related Work

2.1. Deep Q Networks

The deep Q networks (DQNs) are proposed in [2] for handling Atari video games. [7] proposed a deep reinforcement learning architecture for mastering the game of GO. In the area of dialog systems, the DQN is a popular model for building dialog managers and learning dialog policies [8, 9, 10, 11, 12].

DQN is usually implemented as the following. The goal of the network is estimating $q(s, a)$, the Q value of taking action a at state s. At each time step, an DQN-based agent selects an action with a ϵ-greedy strategy, i.e., epsilon % of the time selecting action with largest Q value given current state, and picking a random action for the rest. Meanwhile, the transition of the current step, $(S_t, A_t, R_t, \gamma, S_{t+1})$, is added to a memory buffer for future learning [13]. Here, S_t is the state at time t, A_t is the action taken at time t, R_{t+1} stands for the immediate reward at time step $t + 1$, and γ is a discount rate. The objective function for training the neural network is

$$L = (R_t + \gamma max_{a'} q_{\bar{\theta}}(S_{t+1}, a') - q_\theta(S_t, A_t))^2$$

where θ stands for the parameters set of the current network and $\bar{\theta}$ is the parameters of the target network. The parameters are updated by stochastic gradient descent (SGD).

2.2. Spoken Dialog Systems

Spoken dialog systems aim at completing specific tasks [14, 15, 16, 17] by interacting with users through natural language. Conventionally a dialog manager is built to learn the dialog policy, which decides actions by reasoning from dialog state (the combined representation of user intentions and context). Dialog management is often formulated as a partially observable Markov decision process (POMDP), and solved as a reinforcement learning (RL) problem [18]. As of late, many state-of-the-art dialog systems achieve satisfactory results by leveraging DQNs [2, 19], to learn policy and manage dialogs [8].

Typical applications include flight booking [20], movie recommendation [21], restaurant reservation [22], and vision grounding [23]. Medical bot also benefits from this line of research. [1] applies DQN to decide whether to collect more symptom information from patients by continuing the conversation or conclude the diagnosis with predicting a disease. [3] proposes a knowledge-routed DQN to integrate medical knowledge into dialog management. By considering the relations among diseases and symptoms during decision making, the accuracy of disease prediction is improved. In this work, we integrate meta-learning with DQN to improve the learning efficiency, and evaluate the proposed dialog manager on medical domain.

2.3. Meta-Learning

Recently meta learning starts gaining attention among the whole machine learning field for improving model performance when few labeled training data is available. Model-Agnostic Meta-Learning [4] optimizes parameter initialization over multiple out-of-domain subtasks for the initialization to be generalizable in targeted tasks after fine-tuning on few in-domain la-

bels. Neural Turing machines [24] augment neural models with memory modules to improve performance in limited-data regime. Metric-based meta learning, such as prototypical networks (ProtoNets) [5], siamese neural networks [25], matching networks [6], and structure induction models [26, 27] learns embedding or metric space such that the space can be adapted to domains unseen in the training set with few examples from the unseen domains. Meta-learning models have also been applied in dialog generation [28] for quick policy adaption in different dialog domains. In this work, our proposed model is evaluated on the medical domain that requires not only dialog policy, but also multi-step reasoning with domain-specific knowledge.

3. Method

3.1. Dialog State Representations

Following the method proposed in [1] for vectorizing the dialog states, each dialog state consists of 4 parts:

I. UserAction: The user action of the previous dialog turn:
- **Request**: A user sends a self-report containing a set of explicit symptoms and request for diagnosis.
- **Confirm**: A user confirms the existence of an agent-inquired symptom.
- **Deny**: A user denies the existence of a symptom.
- **NotSure**: A user is not sure about the inquired symptom, usually happened when an unrelated symptom is inquired.

II. AgentAction: The previous action of the dialog agent:
- **Initiate**: The agent initiates the dialog and asks the user to send the self-report.
- **Request**: The agent asks the user if a symptom exists.
- **Inform**: The system predicts and inform the disease.

III. Slots: The set of symptoms appeared in the dialog history and their status. Each symptom has 4 possible status,
- **Confirmed**: Existence of the symptom is confirmed.
- **Denied**: Existence is denied by the user.
- **Unrelated**: The symptom is not necessary for the doctor to make an accurate diagnosis.
- **NotInquired**: The symptom has not been inquired.

IV. NumTurns: The length of the dialog history, in other words, current number of turns.

In each dialog turn, we represent UserAction, AgentAction, and NumTurns with one-hot vectors a^u, a^r, and n respectively. We use a 66-dimension vector s to represent the Slots, where each dimension indicates the status of a symptom. A confirmed, denied, unrelated, and not inquired symptom possesses value 1, $-1, -2$, and 0 in the corresponding dimension. The final input of the neural networks at the t-th turn is represented as

$$s_t = [a_t^u, a_t^r, n_t, s_t] \qquad (1)$$

3.2. Prototypical Q Networks

In this work, we propose the prototypical Q networks (ProtoQNs) as well as corresponding training and evaluation pipelines, based on conventional DQNs and ProtoNets.

We first define the notation of the our dataset as follows. The dataset S contains N doctor-patient conversations, $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where y_i stands for the disease label of the $i-$th training case and x_i stands for the corresponding dialog history, and x_i can be further represented as

$$x_i = \left\{ u_0^i = E^i, (a_1^i, u_1^i), \ldots, (a_k^i, u_k^i), a_{k+1}^i = D^i \right\} \qquad (2)$$

Here E stands for explicit symptoms reported at the beginning of each dialog, a stands for agent follow-up inquiries, u stands for user responses, and D stands for the predicted disease.

The core of the ProtoNets [5] is calculating and updating the prototype embeddings of the output classes. The network classifies examples by comparing the input embedding with the prototypes, and predict the class with its prototype closest to the input embedding. As compared to other single-stage reasoning task, such as image classification or object detection [5], medical dialog requires multi-stage reasoning to infer dialog action through states. As a result, we propose a method for computing action prototype via dialog state embedding.

3.2.1. Dialog state embedding

For each training conversation, x_i, as defined in Equation 2, the dialog state s_j^i at time step j can be represented as

$$s_j^i = \left\{ u_0, (a_1^i, u_1^i), \ldots, (a_{j-1}^i, u_{j-1}^i) \right\} \tag{3}$$

s_j^i is then converted into a representation vector following Equation 1, denoted as f_{enc}, to obtain the state embedding, e_j^i. That is

$$e_j^i = f_{enc}(s_j^i) \tag{4}$$

With the approach described above, for any conversation we can get e_t, the embedding of the dialog state at time step t (i.e., s_t).

3.2.2. Dialog action prediction

At each step of a dialog, the dialog system is provided a dialog state encoding s_t in Equation 1. With the same encoder f_{enc} for embedding dialog states in the training set, we calculate the embedding of the input dialog state e_t in the new dialogs generated in both training and evaluation phases with Equation 4.

Then we can further compute prototypes and predict dialog action. First, with the state embedding e_t, the protoQNs calculate the Q value of the m-th dialog actions a_m by

$$q(a = a_m) = e_t \cdot P_m \tag{5}$$

where a_m is the embedding of the m-th dialog action, generated by mean-pooling a number of dialog states followed by a_m, and

$$P_m = \frac{\sum_{i,j \in D} v(a_j^i) \cdot \mathbb{1}(a_j^i = a_m)}{\sum_{i,j \in D} \mathbb{1}(a_j^i = a_m)} \tag{6}$$

Here, $\mathbb{1}(\cdot)$ is the indicator function, and D is the set of examples used for computing prototypes. In training, D is a small subset of dialog states sampled from training set followed by action a_m, while in evaluation, D is the entire training set. In Algorithm 1, we show the calculation of prototype embeddings.

3.2.3. Disease prediction and training

Each dialog starts from an explicit symptom set provided by a user goal, and the model inquiries a set of symptoms before making the final disease prediction. For each inquiry, the user simulator replies based on the implicit symptom set as described in Section 3.1. The conversation stops when the systems output a disease prediction. Summarizing the previous sections and descriptions, we provide the complete procedure of a medical dialog in Algorithm 2.

For each simulated dialog described above, the model sees a success or failure reward when it informs the user a predicted

---

Algorithm 1 Calculating prototype embeddings

Function $protoEmbed(A, H)$
Inputs: dialog action set A, entire dialog history in the training corpus $H = \{(s_1, a_1), \ldots, (s_m, a_m)\}$.
Outputs: Prototype embeddings of dialog actions P.

1: **for** $a' \in A$ **do**
2:      $A_{a'} = \{(s_k, a_k) \in H, a_k = a'\}$
3:      **if** Training **then**
4:          Sample D from $A_{a'}$ with size = 10
5:      **else if** Evaluating **then**
6:          D = $A_{a'}$
7:      **end if**
8:      Calculate $P_{a'}$ with Equation 6
9: **end for**

---

disease. The ProtoQN updates its weights based on the reward with stochastic gradient descent (SGD) following the standard pipeline of training a DQN applied in [1, 3]. For evaluation, the ProtoQN generates prototype embeddings only once before the evaluation begins with all real dialog histories in the training corpus.

Algorithm 2 Automatic diagnosis dialog process

Inputs: Explicit symptom set E, initial turn Id $t = 1$, empty implicit symptom set $I = \{\}$, dialog history for training H, encoder f_{enc}
Outputs: Final disease prediction D.

1: $P = protoEmbed(A, H)$
2: **while** Dialog not end **do**
3:      $s_t = State(E, I, t)$ with Equation 1
4:      $h_t = f_{enc}(s_t)$
5:      $a_t = \mathrm{argmax}_a \, h_t \cdot P_a^T$ with Equation 5
6:      **if** $a \in$ Symptom **then**
7:          $u_t =$ user response
8:          $I = I + (a_t, u_t)$
9:          $t = t + 1$
10:      **else if** $a \in$ Disease **then**
11:          $D = a_t$
12:          $endDialog()$
13:      **end if**
14: **end while**

---

4. Experiments

4.1. Data and Experiment Settings

In this work, we evaluate the models on Muzhi dataset [1]. The dataset contains 710 medical dialogs between real doctors and patients, and each is annotated as a user goal, covering 4 different diseases and 66 symptoms. In this work, we apply the official train-test split. 568 user goals are used from training and other 142 are used for evaluation.

Meanwhile, we apply a simulator for providing user response in the conversations. To simulate speech noises and mistakes led by user knowledge biases, we apply intent and slot noises to the simulator [3] to evaluate the model performances under different levels of noises. In our experiments, we apply 0%, 10%, 20%, and 30% error rates, i.e., the probability that the status of an inquired symptom is sampled at random rather than based on annotation.

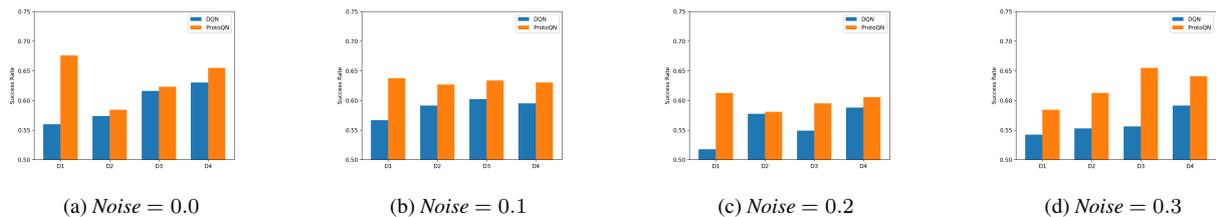We also conduct 2 groups of experiments for evaluating the

| (a) Noise = 0.0 | (b) Noise = 0.1 | (c) Noise = 0.2 | (d) Noise = 0.3 |

Figure 2: *Performances of ProtoQN and DQN on 4 noise levels. Each figure stands for a noise, while the blue bars stand for the performances of DQNs, and the orange bars stand for the performances of ProtoQNs.*

model performances at various conditions. Firstly we train both ProtoQN and DQN on the entire training set. This is the conventional with fully supervised learning setting. Secondly, we adopt a meta-learning-like setting to evaluate the model performance at few-shot learning. We pre-train the models with 3 diseases, and fine-tune the models with randomly selected training samples from the trained diseases plus a few cases of the new disease. In both learning tasks, the neural models are evaluated on the entire public test set. In this paper we set the success reward of ProtoQN to 20, failure reward to 0, and the maximum number of turns is 44. For DQN, we keep the settings in [1].

4.2. Fully Supervised Learning

We first compare the ProtoQN with DQN at the normal supervised learning setting to evaluate their abilities of learning dialog policy with enough training data. Experimental results are shown in Table 1. We obtain the DQN performance directly from [1].

Table 1: *Experimental results of ProtoQN and DQN on supervised learning.*

Models	Success Rate (%)	Reward
DQN	65	20.51
ProtoQN	70.42	23.58

Experiments showed that the ProtoQN significantly outperformed the DQN baseline, without adding external knowledge and hand-crafted features [3]. The improvement showed that utilizing the dialog history between real doctors and patients to calculate prototype embeddings is effective and provides better ability for the model to learn the dialog policy. While DQN learns Q-values indirectly from simulated conversations, ProtoQN directly relates Q-values with dialog actions and states in real conversation.

4.3. Few-shot New Disease Adaption

In this part, we investigate the model performance at the situation when a new disease appears after the model was trained, and only a few examples are available for training. To evaluate this situation, we adopt an experiment setting popular in few-shot learning, where we first pre-train the models on the training set from a subset of diseases. Then the models are fine-tuned (i.e., adapted) on a small number, N, of examples from a new disease. In order to prevent catastrophic forgetting [29], we also randomly select N samples from each pre-trained diseases to compose the adaptation set. Since the Muzhi corpus includes 4 diseases, we conduct 4 iterations of evaluation, with each cor-

Table 2: *Average success rates (%) of the meta-learning tasks with DQN and ProtoQN under different noise levels.*

Noise	0	0.1	0.2	0.3
DQN	59.51	58.89	55.81	56.07
ProtoQN	63.47	63.21	59.85	62.32

responding to one of the 4 diseases as the new disease, and the remaining as the pre-trained ones. After fine-tuning, the models are also evaluated with the public test set. Also, we consider different noise levels in this task. Our purpose is evaluating how well the models learn new diseases with few examples and not forgetting the knowledge for the pre-trained diseases.

The experiment results of the new disease adaption are shown in Table 2. From the table we see that the ProtoQN significantly outperformed the DQN model and achieves SOTA performance under few-shot learning setting. The fact shows that learning shared embeddings from real conversations is more efficient when adapting the model for learning new diseases with few examples. Meanwhile, many shared knowledge from pre-trained diseases can still be applied for the new ones. The learned knowledge about symptoms allows the neural network to learn new diseases with a better initialization of dialog policy, and thus makes the adaptation faster and better. We also found that although the increasing of noise level degrades the performance of DQNs, ProtoQN yields steady success rate when the noise level varies. we believe the robustness results from the ensemble nature of ProtoNet's inference mechanisms.

5. Conclusion and Future Work

In this work, we propose a novel dialog management model, prototypical Q network, for supervised and few-shot dialog policy learning. We apply this model in the area of automatic conversational diagnosis. Experiments showed that the ProtoQNs outperforms the DQN model in both supervised and few-shot settings. In the supervised setting, ProtoQNs achieve results comparable to SOTA without using domain-specific features. As for the few-shot experiment, ProtoQN learns new diseases using few training samples without forgetting previously learned, and achieves SOTA. The model also shows less degradation as we injecting noise to conversation. Our study suggests that modeling real conversations directly reinforces simulator-based dialog policy learning. Embeddings of dialog actions are shareable among tasks (diseases, in our case) and benefits the fast adaptation to new ones. Here we show promising results in medical domain. In future, we will investigate more adaptive models as well as different domains and corpora toward the goal of modeling new dialog tasks better and with fewer examples.

6. References

[1] Z. Wei, Q. Liu, B. Peng, H. Tou, T. Chen, X. Huang, K.-F. Wong, and X. Dai, "Task-oriented dialogue system for automatic diagnosis," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 201–207.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[3] L. Xu, Q. Zhou, K. Gong, X. Liang, J. Tang, and L. Lin, "End-to-end knowledge-routed relational dialogue system for automatic diagnosis," *arXiv preprint arXiv:1901.10623*, 2019.

[4] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[5] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, 2017, pp. 4077–4087.

[6] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, 2016, pp. 3630–3638.

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[8] T. Zhao and M. Eskenazi, "Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning," *arXiv preprint arXiv:1606.02560*, 2016.

[9] X. Yang, Y.-N. Chen, D. Hakkani-Tür, P. Crook, X. Li, J. Gao, and L. Deng, "End-to-end joint learning of natural language understanding and dialogue manager," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5690–5694.

[10] Z. C. Lipton, J. Gao, L. Li, X. Li, F. Ahmed, and L. Deng, "Efficient exploration for dialogue policy learning with bbq networks & replay buffer spiking," *arXiv preprint arXiv:1608.05081*, vol. 3, 2016.

[11] B. Peng, X. Li, L. Li, J. Gao, A. Celikyilmaz, S. Lee, and K.-F. Wong, "Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning," *arXiv preprint arXiv:1704.03084*, 2017.

[12] M. Fazel-Zarandi, S.-W. Li, J. Cao, J. Casale, P. Henderson, D. Whitney, and A. Geramifard, "Learning robust dialog policies in noisy environments," in *Workshop on Conversational AI: Today's Practice and Tomorrow's Potential, NeurIPS*, 2017.

[13] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, vol. 8, no. 3-4, pp. 293–321, 1992.

[14] K. A. Papineni, S. Roukos, and R. T. Ward, "Natural language task-oriented dialog manager and method," Jun. 12 2001, uS Patent 6,246,981.

[15] K. Scheffler and S. Young, "Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning," in *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 2002, pp. 12–19.

[16] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The hidden information state model: A practical framework for pomdp-based spoken dialogue management," *Computer Speech & Language*, vol. 24, no. 2, pp. 150–174, 2010.

[17] H. Luo, M. Mohtarami, J. R. Glass, K. Krishnamurthy, and B. Richardson, "Integrating video retrieval and moment detection in a unified corpus for video question answering." in *INTERSPEECH*, 2019, pp. 599–603.

[18] S. Young, M. Gašić, B. Thomson, and J. D. Williams, "Pomdp-based statistical spoken dialog systems: A review," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[19] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," 2014.

[20] S. Seneff and J. Polifroni, "Dialogue management in the mercury flight reservation system," in *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3*. Association for Computational Linguistics, 2000, pp. 11–16.

[21] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, and J. Weston, "Evaluating prerequisite qualities for learning end-to-end dialog systems," *arXiv preprint arXiv:1511.06931*, 2015.

[22] A. Bordes, Y.-L. Boureau, and J. Weston, "Learning end-to-end goal-oriented dialog," *arXiv preprint arXiv:1605.07683*, 2016.

[23] P. Chattopadhyay, D. Yadav, V. Prabhu, A. Chandrasekaran, A. Das, S. Lee, D. Batra, and D. Parikh, "Evaluating visual conversational agents via cooperative human-ai games," in *Fifth AAAI Conference on Human Computation and Crowdsourcing*, 2017.

[24] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[25] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[26] Y. Shen, Z. Lin, C.-W. Huang, and A. Courville, "Neural language modeling by jointly learning syntax and lexicon," *arXiv preprint arXiv:1711.02013*, 2017.

[27] H. Luo, L. Jiang, Y. Belinkov, and J. Glass, "Improving neural language models by segmenting, attending, and predicting the future," *arXiv preprint arXiv:1906.01702*, 2019.

[28] K. Qian and Z. Yu, "Domain adaptive dialog generation via meta learning," *arXiv preprint arXiv:1906.03520*, 2019.

[29] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," *arXiv preprint arXiv:1810.11910*, 2018.